

ORACLE



# Modernisez la haute disponibilité de votre MySQL

Webinar du 7 Octobre 2025

---

**Emmanuel COLUSSI**

MySQL Solution Architect EMEA

Oracle MySQL GBU



# Emmanuel COLUSSI

## Architecte Solutions MySQL chez Oracle

25 ans d'expérience dans la conception et la mise en œuvre de solutions complexes et hautement disponibles, dans des secteurs tels que la santé, les télécommunications, l'assurance, la finance et la défense.

Spécialisé dans les architectures cloud-native, les déploiements sur Kubernetes, et passionné par la programmation en Go.



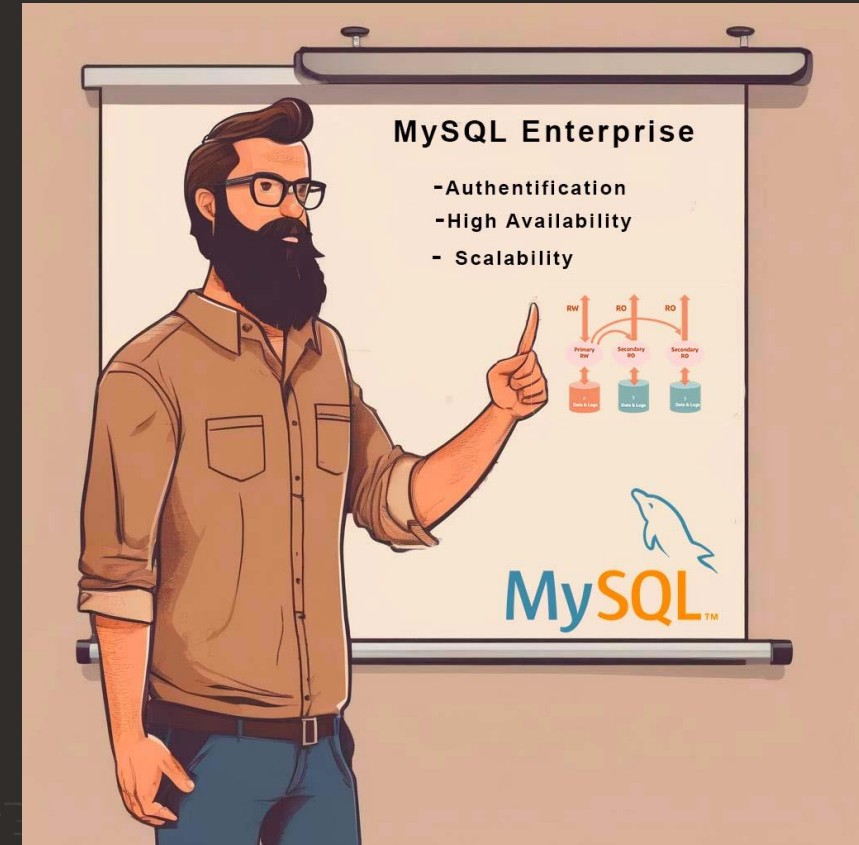
📁 Mes webinars sur GitHub :  
<https://github.com/colussim/webinar>

👉 Restons en contact :

- <https://www.linkedin.com/in/emmanuel-colussi-b8b29113/>

# Agenda

- 👉 MySQL Enterprise Edition
- 👉 Building Blocks InnoDB Cluster
- 👉 Architecture InnoDB Cluster
- 👉 Topology
- 👉 Demo



# Questions ?





# HAPPY BIRTHDAY



# MySQL

# MySQL Enterprise Edition

---

The different versions

# MySQL Community VS MySQL Enterprise

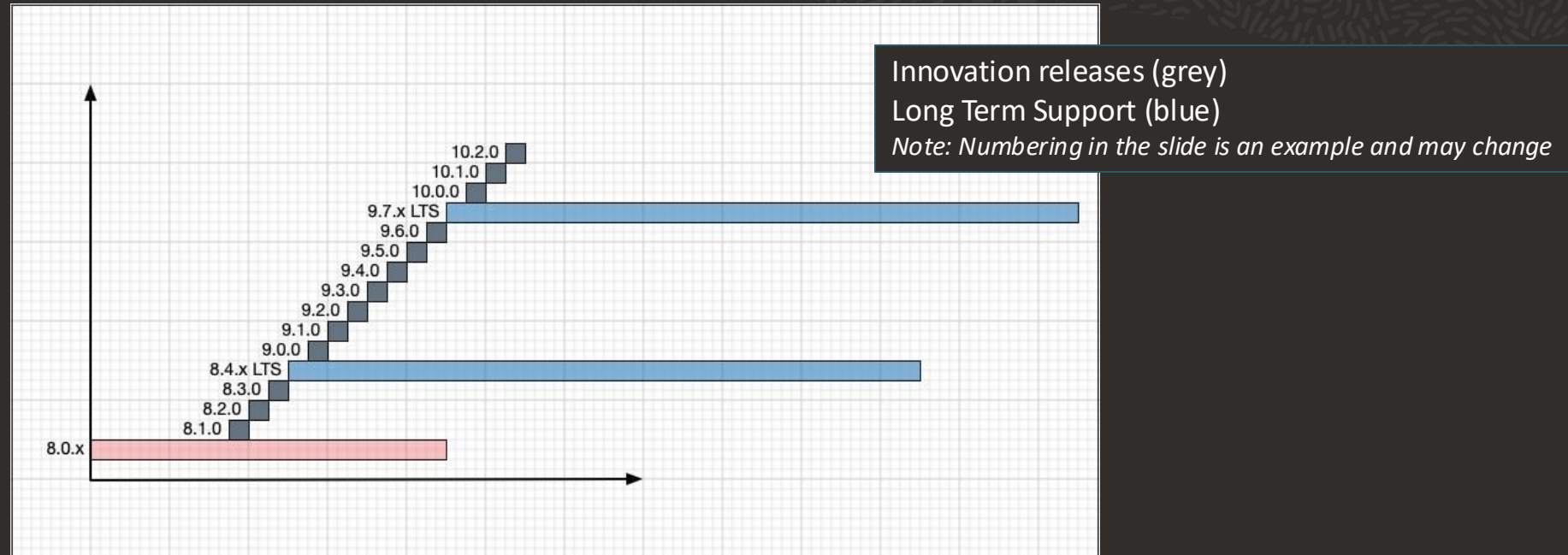


MySQL Community



# NEW: MySQL LTS & Innovation Releases

MySQL Community and Enterprise



## MySQL Long-Term Support (LTS) Releases

- bugfix & security patches only
- backwards compatibility
- support lifecycle: 5y premier + 3y extended
- Include tools (like MySQL Shell)

## MySQL Innovation Releases

- leading-edge innovations
- will likely released every quarter
- Include tools (like MySQL Shell)
- Connectors released as Innovation only



# High Availability, Reduced Downtime!





# High Availability: Always Up, Never Down!



# Building Blocks of MySQL InnoDB Cluster

## MySQL Instances

### Concepts – RTO & RPO

#### ✓ Recovery Time Objective (RTO)

- Total time required to complete a repair and reconnect a system.

#### ✓ Recovery Point Objective (RPO).

The point in time to which you need to recover data. This represents the window of data loss.

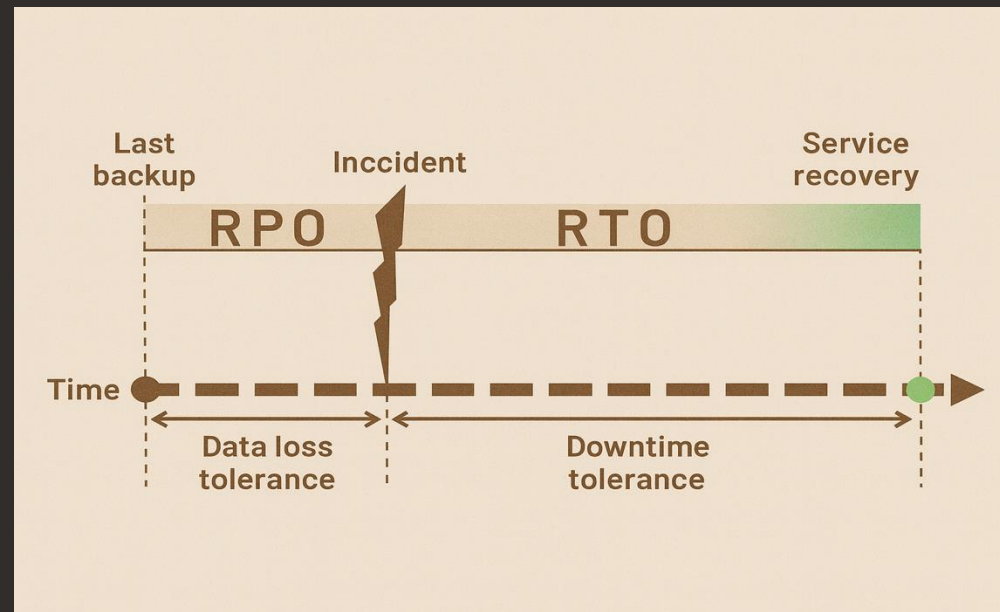


### Types d'échec

High Availability: Single server failure, network partition.

Disaster Recovery: Total region/network outage

Human Error: Errors DBA, SQL Injection , ...



# MySQL InnoDB Cluster

---

Building Blocks



# Building Blocks of MySQL InnoDB Cluster

1



## MySQL Instances

Instances that make up the cluster.

2



## InnoDB

The main storage engine that provides transactional capabilities.

3



## Group Replication

A synchronous replication mechanism ensuring data consistency between instances.

4



## MySQL Shell

Tool for managing and configuring the cluster.

5



## MySQL Router

Middleware for routing queries to the appropriate instances.



# Building Blocks of MySQL InnoDB Cluster

## MySQL Instances

# 1



### MySQL Instances

Instances that make  
up the cluster.

- ✓ **Minimum number of instances:** You need at least 3 instances to ensure effective resilience and fault tolerance. This allows for achieving a majority during decision processes (like leader election).
- ✓ **Maximum number of instances:** While there is no strict limit defined by MySQL, common practice recommends not exceeding 9 instances in a cluster to avoid complexities and performance issues. Generally, an odd number of instances is preferable to facilitate decision-making in case of failure.

# Building Blocks of MySQL InnoDB

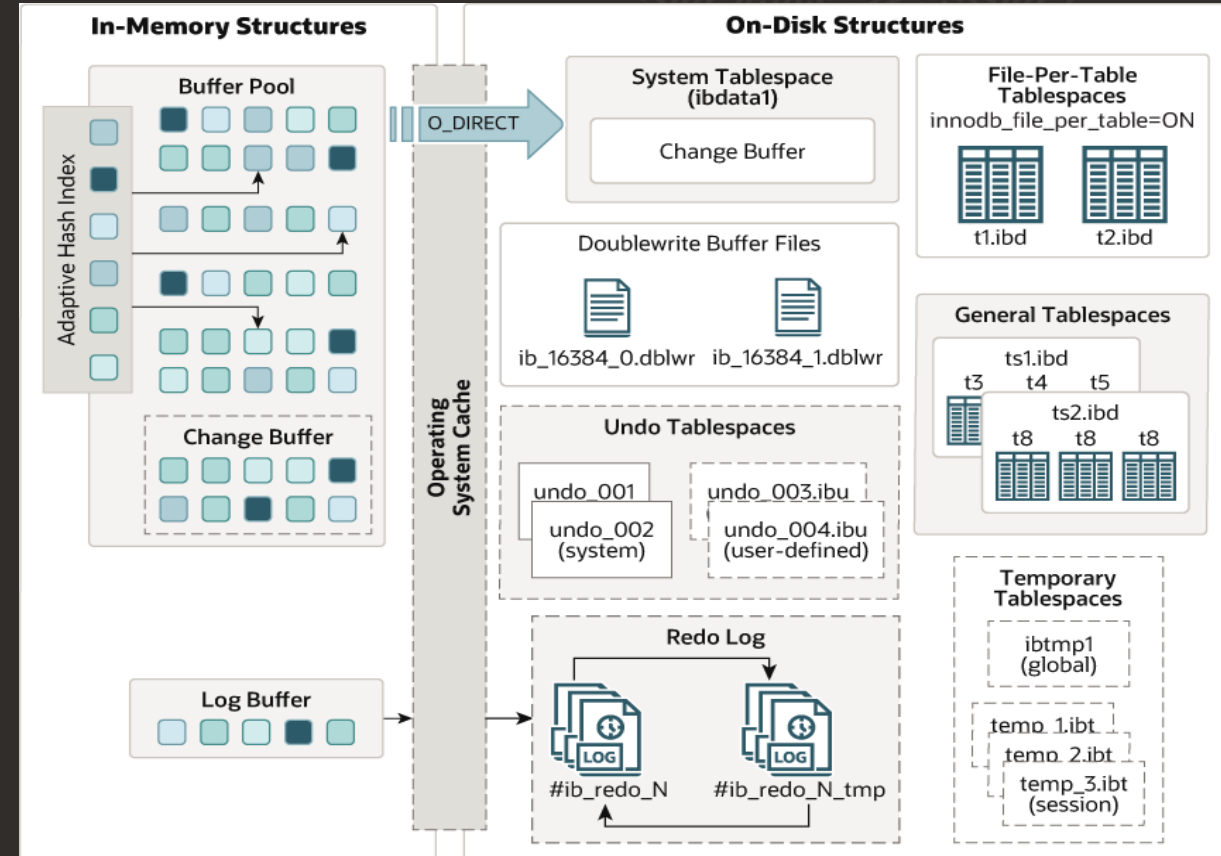
## InnoDB

### 2

#### InnoDB

The main storage engine that provides transactional capabilities.

**MySQL InnoDB** is a storage engine integrated into MySQL that offers several key features, making MySQL suitable for a variety of applications, particularly those requiring advanced transaction management.



# Building Blocks of MySQL InnoDB

## InnoDB : main characteristics



# 2

### InnoDB

The main storage engine that provides transactional capabilities.

- ✅ **ACID Transactions:** InnoDB supports ACID properties (Atomicity, Consistency, Isolation, Durability), ensuring that transactions are processed reliably.
- ✅ **Row-Level Locking:** InnoDB uses a row-level locking mechanism, allowing for more efficient concurrent access to the database. This reduces conflicts and improves performance in high-traffic environments.
- ✅ **Crash Recovery:** It has a recovery log that allows the database to restore its state in the event of a failure or unexpected shutdown.
- ✅ **Foreign Key Support:** InnoDB includes support for referential integrity constraints, allowing relationships between tables to be defined through foreign keys.
- ✅ **Performance:** It uses a buffer pool-based architecture for caching, which improves the performance of input/output (I/O) operations and optimizes read and write performance.
- ✅ **Compressed Tables:** InnoDB allows for table compression, which can reduce the storage space required while maintaining good performance.
- ✅ **Partitioning:** It also supports table partitioning, allowing data to be divided into segments that can be managed independently, improving performance and management.

# Building Blocks of MySQL InnoDB Cluster

## Group Replication

# 3

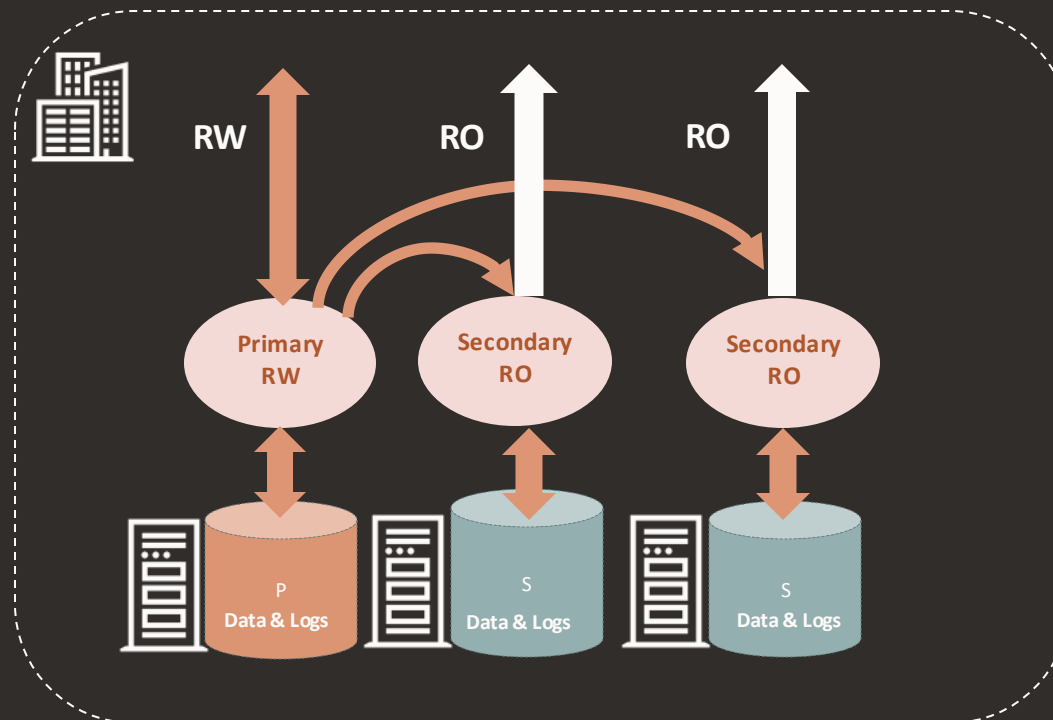


### Group Replication

A synchronous replication mechanism ensuring data consistency between instances.

MySQL Group Replication is a built-in feature in MySQL that enables the creation of highly available and fault-tolerant MySQL database clusters.

It allows multiple MySQL servers to work together as a single group, where changes made to one server are automatically propagated to all other servers in the group.



# Building Blocks of MySQL InnoDB Cluster

## Group Replication : key aspects

3



### Group Replication

A synchronous replication mechanism ensuring data consistency between instances.

- ✓ **Data Consistency:** Group Replication ensures that all members of the group have the same data and maintain consistency through its use of distributed transactions.
- ✓ **High Availability:** If one server goes down, the other servers in the group can continue to operate normally, improving overall availability.
- ✓ **Automatic Failover:** The system can automatically detect failure situations and promote a new primary server without manual intervention.
- ✓ **Read and Write Scalability:** It supports both read and write operations, enabling horizontal scalability as you can add more servers to the group to handle increased load.
- ✓ **Synchronous Replication:** Group Replication uses a synchronous replication mechanism, meaning changes are confirmed by a majority of the group members before being considered successful.
- ✓ **Conflict Handling:** It includes built-in mechanisms for conflict resolution, allowing for safe concurrent write operations.



# Building Blocks of MySQL InnoDB Cluster

## Group Replication : requirements

3



### Group Replication

A synchronous replication mechanism ensuring data consistency between instances.

- ✓ InnoDB only tables
- ✓ Primary key is required on every table
  - Invisible columns may have Primary Key/Unique Index
  - Generated Invisible Primary Key mode (GIPK) automatically adds invisible columns with PK on tables created without an explicit primary key
- ✓ Binary Logs active (ROW)
- ✓ Global Transaction Identifiers (GTID) turned on
- ✓ 3, 5, 7 or 9 nodes

# Building Blocks of MySQL InnoDB Cluster

## MySQL Shell

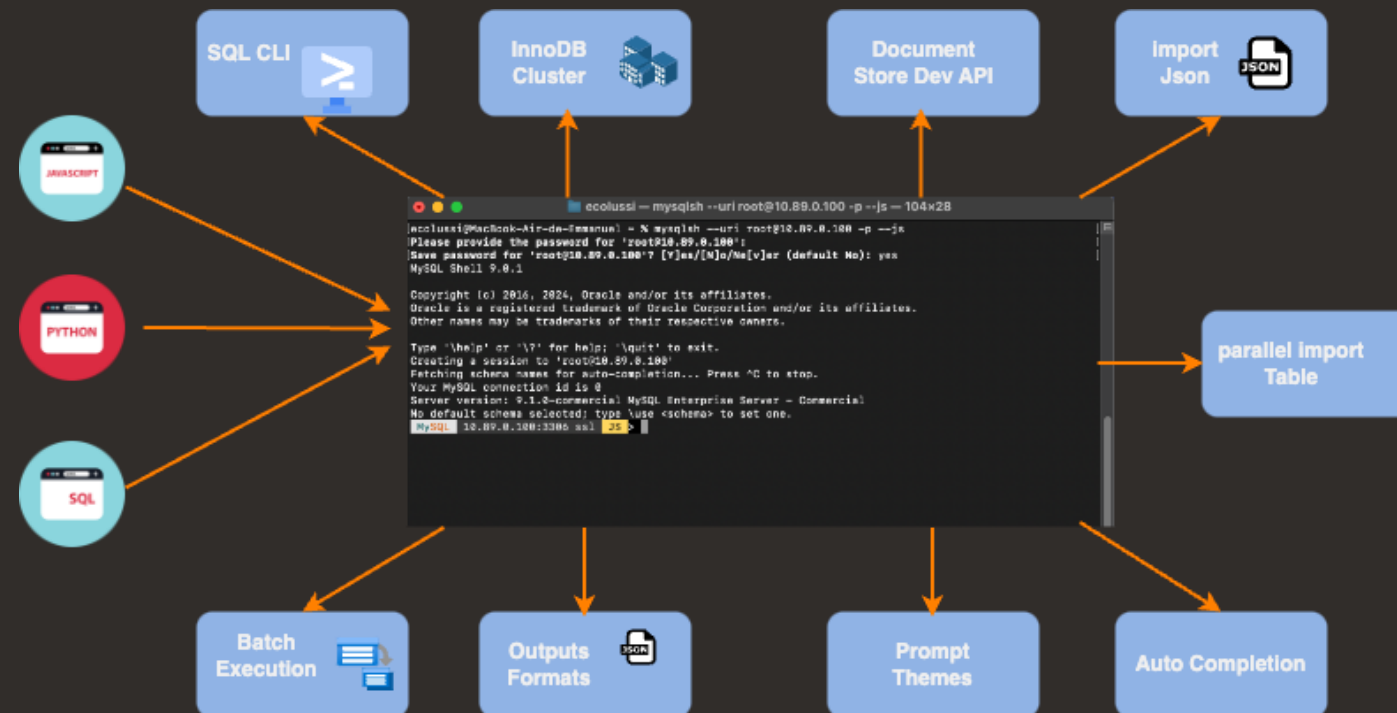
4

### MySQL Shell

Tool for managing and configuring the cluster.

MySQL Shell is a powerful and interactive environment for working with MySQL.

It offers several usage modes, including SQL mode, JavaScript mode, and Python mode, allowing developers and database administrators to execute scripts and queries efficiently.



# Building Blocks of MySQL InnoDB Cluster

## MySQL Shell : features



### 4

#### MySQL Shell

Tool for managing and configuring the cluster.

- ✓ **Multi-language Support:** Ability to execute scripts in SQL, JavaScript, or Python. This offers flexibility for developers and administrators based on their preferences.
- ✓ **Automation:** Allows for the automation of common database management tasks, such as backups, recoveries, and InnoDB Cluster deployment.
- ✓ **Integration with MySQL Server:** MySQL Shell easily connects to MySQL instances, facilitating command execution and server configuration management.
- ✓ **Increased Productivity:** Provides an interactive environment with auto-completion features.
- ✓ **Ease of Cluster Management:** Simplifies MySQL cluster management through streamlined commands.

```
MySQL JS > \c root@localhost
Creating a session to 'root@localhost'
Enter password:
Fetching schema names for autocompletion... Press ^C to stop.
Your MySQL connection id is 13 (X protocol)
Server version: 8.0.11 MySQL Community Server - GPL
No default schema selected; type \use <schema> to set one.
MySQL > localhost:33060+ JS > session.createSchema('docstore')
MySQL > localhost:33060+ JS > \use docstore
Default schema 'docstore' accessible through db.
MySQL > localhost:33060+ JS > docstore JS >
```

# Building Blocks of MySQL InnoDB Cluster

## MySQL Router

5



### MySQL Router

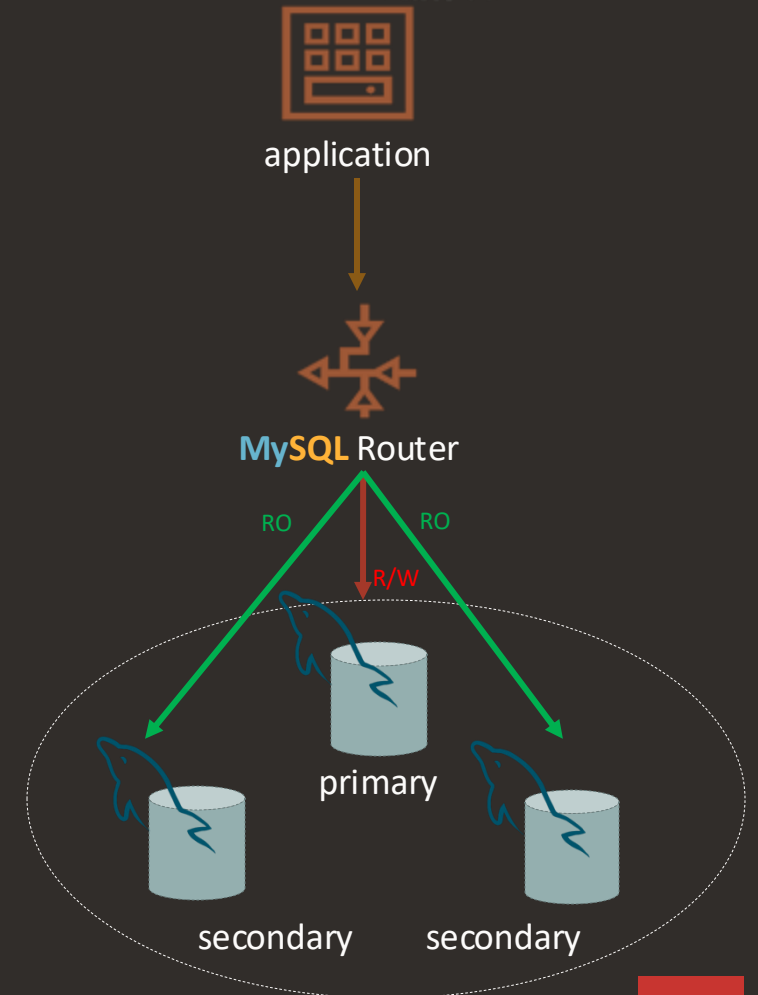
Middleware for routing queries to the appropriate instances.

MySQL Router is a lightweight software that acts as a routing layer for applications connecting to MySQL instances.

It manages the routing of connections to the appropriate backend servers in a high-availability environment.  
Copy

MySQL Router offers 3 ports :

- ✓ connect to read/write node (6446)
- ✓ connect to read only node (6447) : Round robin
- ✓ Transparent split read/writes (6450):
  - Direct all read traffic to read-only instances, and all write traffic to read-write instances
  - Each client session can communicate with one read\_write and one read\_only destination

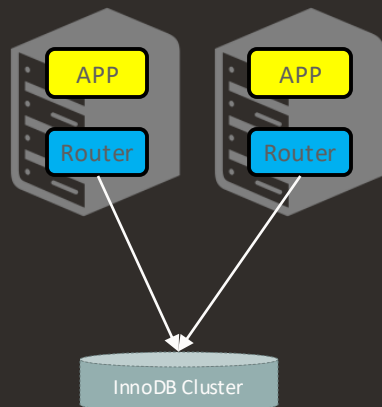


# Building Blocks of MySQL InnoDB Cluster

## MySQL Router : deployment scenarios

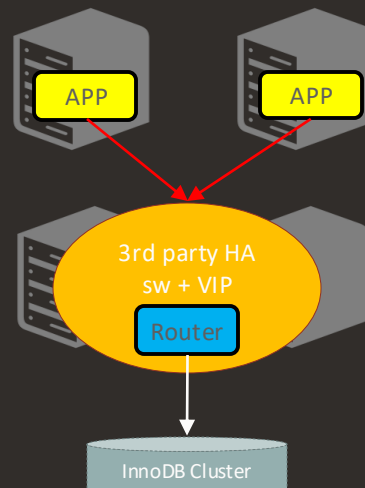
### One per application server

- **Pro**
  - minimal overhead
  - No network hop between application and database
  - easy deploy of new Routers
- **CONS**
  - firewall between layers requires a rule for each application server



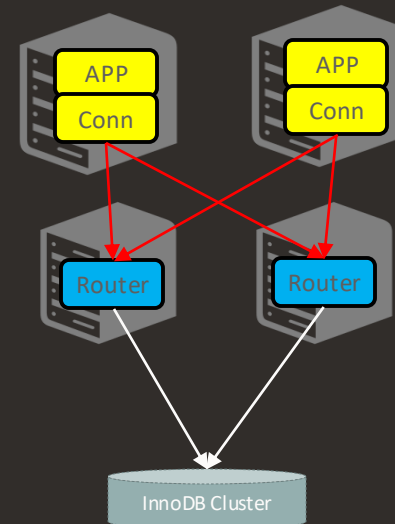
### 3rd party cluster on dedicated servers

- **Pro**
  - no single point of failure
  - easy configuration with firewalls
- **CONS**
  - required 3<sup>rd</sup> party software
  - required an additional hop on the network



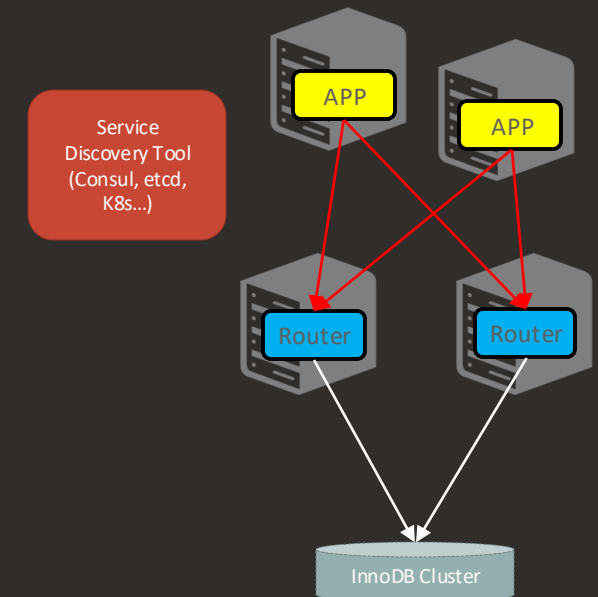
### Dedicated servers + connectors built-in HA

- **Pro**
  - No need of 3rd party sw or VIP
- **CONS**
  - Not all connectors have built-in HA capabilities
  - required an additional hop on the network



### DNS/SRV

- **PRO**
  - a single DNS domain can map to multiple targets (servers)
- **CONS**
  - Requires 3<sup>rd</sup> party SW
  - Requires connectors that support it





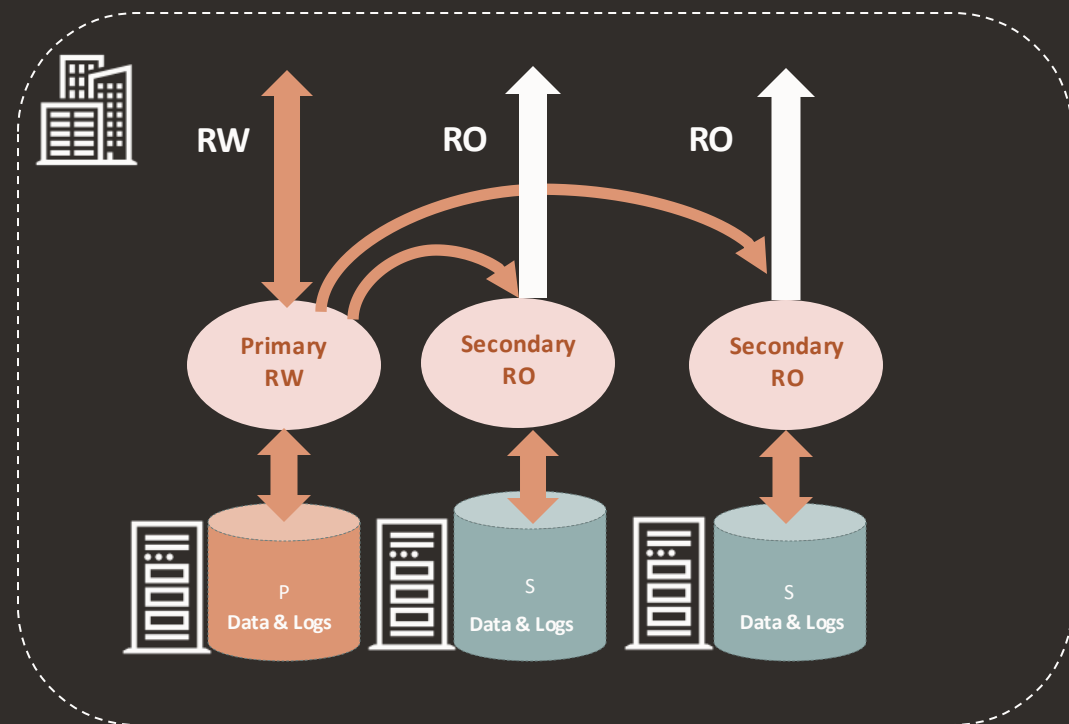
# MySQL InnoDB Cluster

---

## Topologies

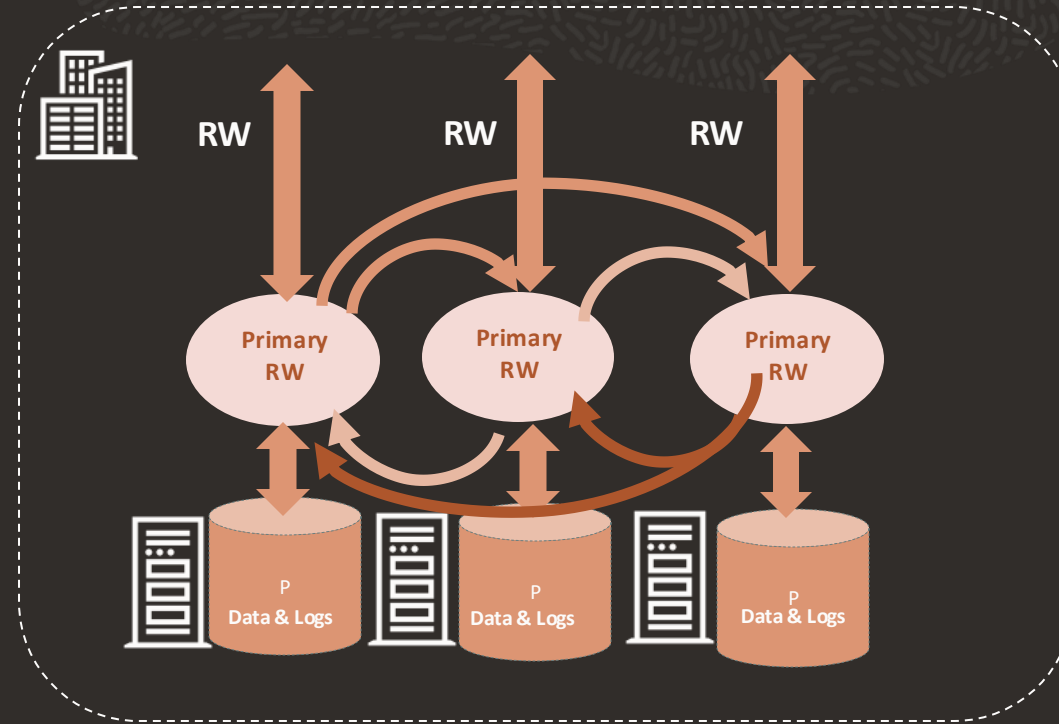
# InnoDB Cluster

## Topologies



### Single primary node topology by default

- 3, 5, 7, or 9 nodes
- The cluster remains resilient after 1, 2, 3, or 4 failures, respectively.

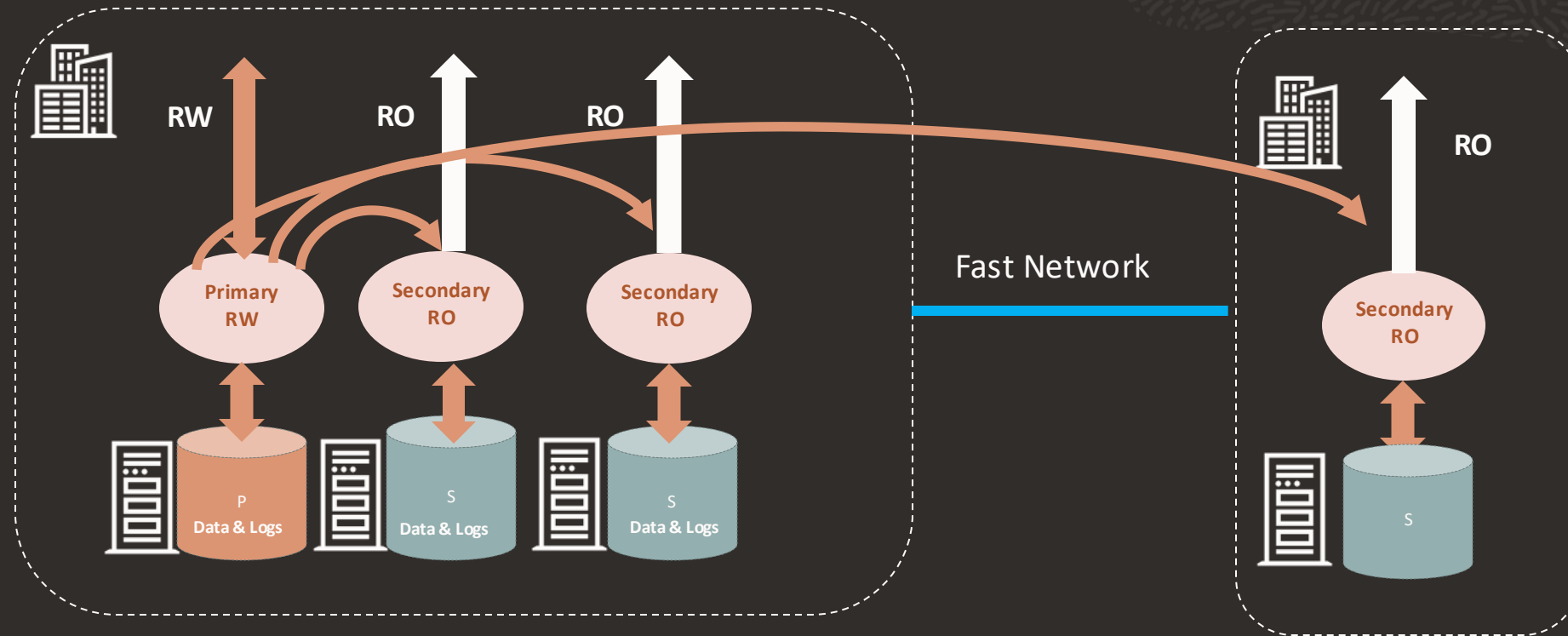


### Multi-Primary Topology

All nodes are read/write

# InnoDB Cluster

## Topologies

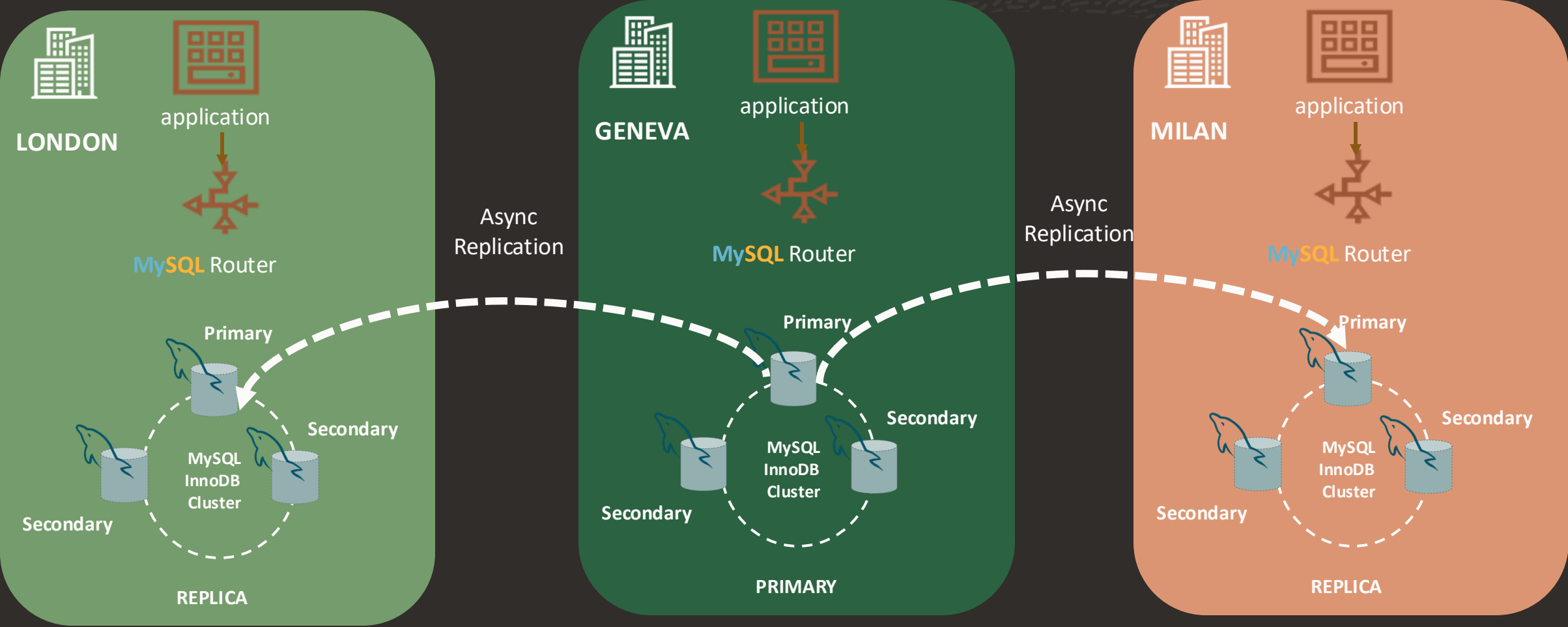


### Single primary node topology by default

- 3, 5, 7, or 9 nodes
- The cluster remains resilient after 1, 2, 3, or 4 failures, respectively.

# InnoDB Cluster

Topologies : ClusterSet



# InnoDB Cluster

Topologies : ClusterSet main features



## Centralized Management :

- ✓ Provides a unified interface for managing multiple MySQL clusters, simplifying administrative operations and monitoring.

## Horizontal Scalability :

- ✓ Allows easy addition of new clusters to handle data growth and performance demands.
- ✓ Ideal for applications requiring constant availability and real-time data access.

## Load Balancing :

- ✓ Facilitates workload distribution across multiple clusters, optimizing performance and resilience.

## High Availability and Recovery :

- ✓ Offers advanced options for fault tolerance, replication, and disaster recovery.
- ✓ Integrates failover mechanisms to ensure service continuity even during hardware failures.



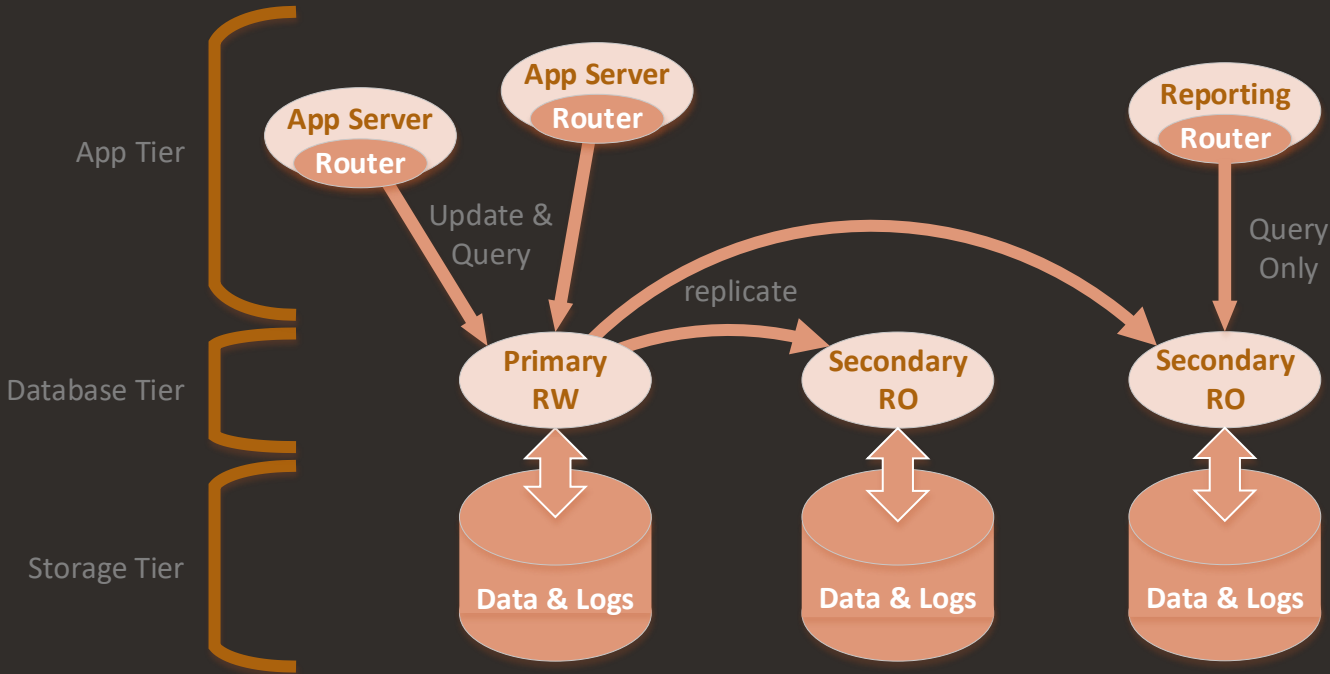
# MySQL InnoDB Cluster

---

## Failover and Recovery

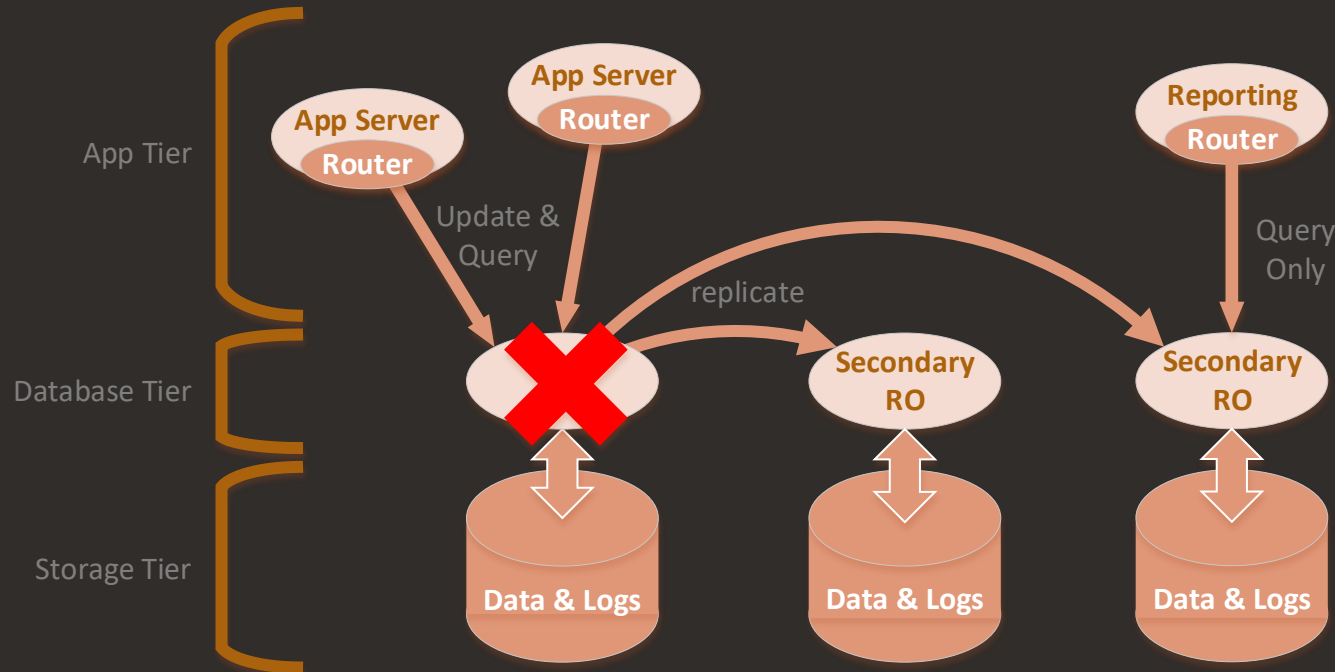
# InnoDB Cluster

## Standard Single Primary Node Topology



# InnoDB Cluster

## Standard Single Primary Node Topology

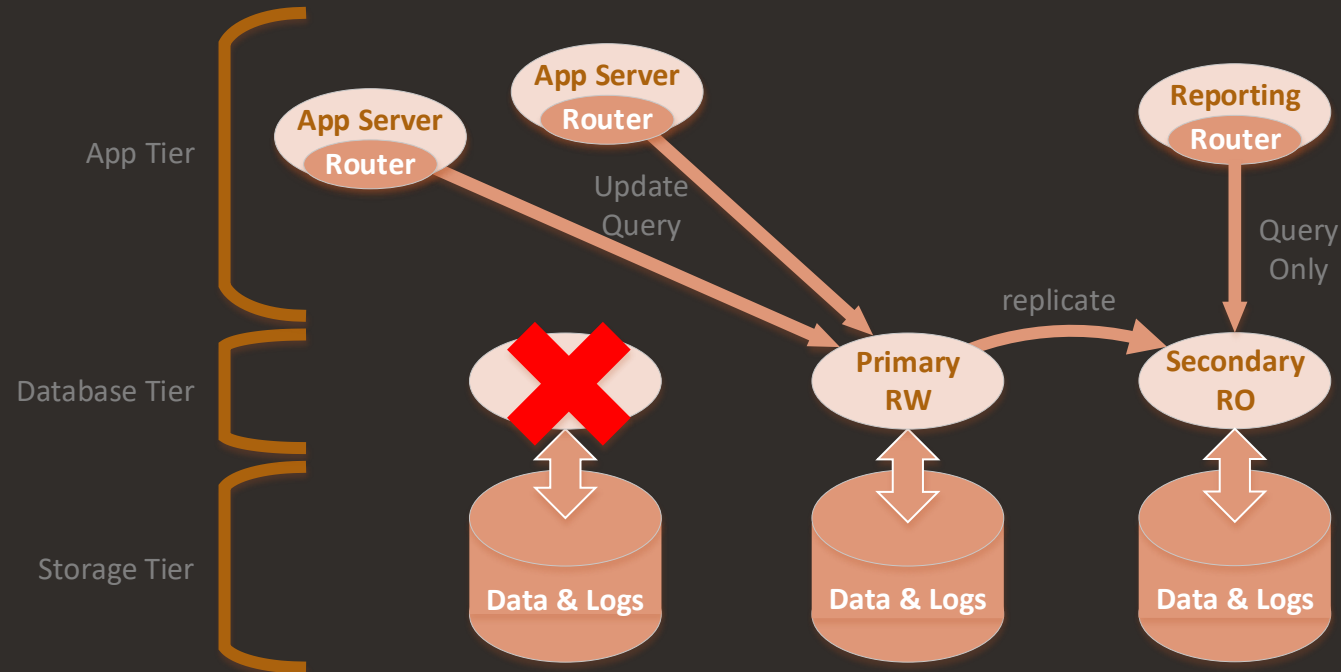


### Primary goes down

- ✓ The worst case!
- ✓ Remaining 2 nodes can form quorum
- ✓ Decision made to failover
- ✓ Nodes elect a new master
- ✓ Decision can be influenced with weighting

# InnoDB Cluster

## Standard Single Primary Node Topology

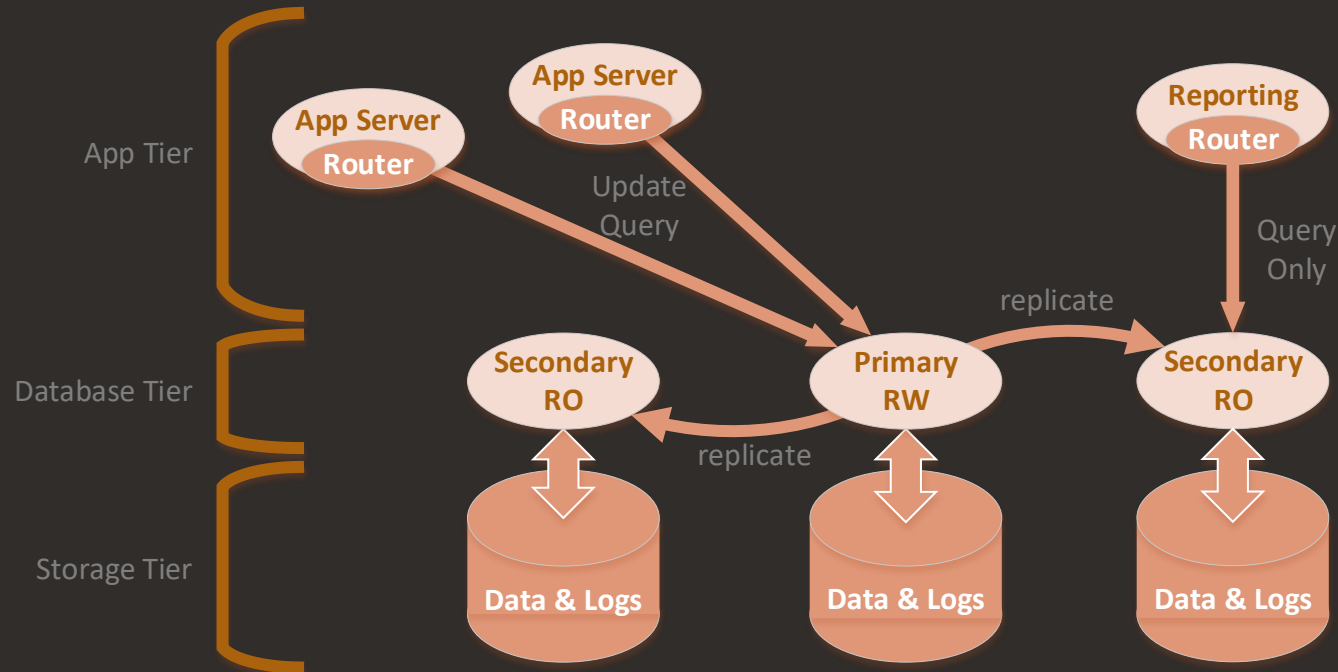


The failover occurs:

- ✓ Routers automatically switch to the new primary
- ✓ No script is required.

# InnoDB Cluster

## Standard Single Primary Node Topology



- ✓ Reinserts the node into the cluster as a secondary
- ✓ Applies the lagging data
- ✓ Becomes consistent

 `cluster.setPrimaryInstance()`

# MySQL InnoDB Cluster

---

Maintenance



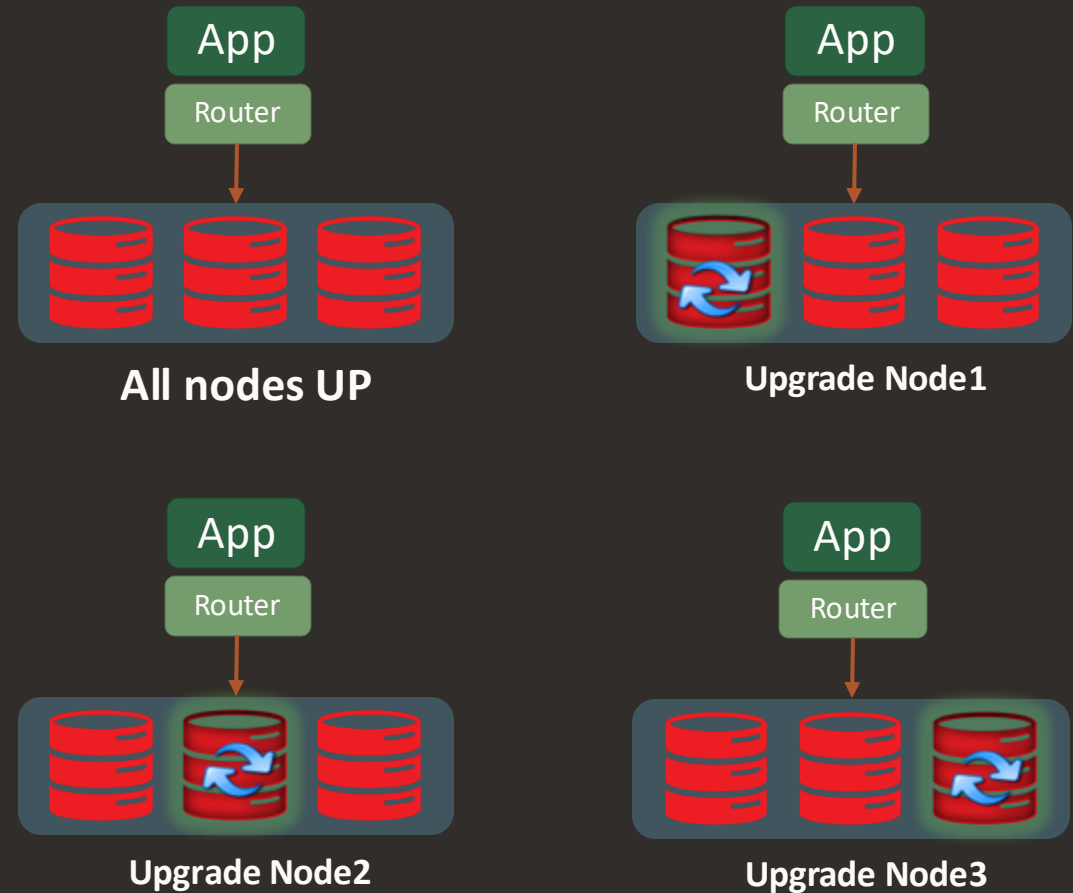
# InnoDB Cluster

## Online Upgrade of InnoDB Cluster (OS/MySQL/HW)

✓ Upgrade the operating system, hardware, or software without downtime for the application.

✓ An online upgrade is possible by performing a rolling upgrade on each server, starting with the secondaries and finishing with the primary:

- Stop MySQL: `systemctl stop mysqld`
- Upgrade binaries: `yum update mysql`
- Start MySQL: `systemctl start mysqld`



# InnoDB Cluster

## Upgrade Steps

✓ Upgrade MySQL Router

✓ Upgrade MySQL Shell

Upgrade the InnoDB Cluster Metadata schema : `dba.upgradeMetadata()`

✓ Upgrade MySQL Server

✓ Post Upgrade Status Check

Learn More Explore the following links. 

[InnoDB Cluster Upgrade](#)

[Upgrade Metadata Schema](#)

# InnoDB Cluster

## Main useful commands

See these links:

Restoring and Rebooting an InnoDB Cluster :

<https://dev.mysql.com/doc/mysql-shell/8.4/en/troubleshooting-innodb-cluster.html>

Monitoring InnoDB Cluster

<https://dev.mysql.com/doc/mysql-shell/8.4/en/monitoring-innodb-cluster.html>

Upgrade InnoDB Cluster

<https://dev.mysql.com/doc/mysql-shell/8.4/en/mysql-innodb-cluster-upgrade.html>

MySQL Shell API: Cluster Class Reference

[https://dev.mysql.com/doc/dev/mysqlsh-api-javascript/8.0/classmysqlsh\\_1\\_1dba\\_1\\_1\\_cluster.html#af517c5f41a3d1104124b06bad029f423](https://dev.mysql.com/doc/dev/mysqlsh-api-javascript/8.0/classmysqlsh_1_1dba_1_1_cluster.html#af517c5f41a3d1104124b06bad029f423)



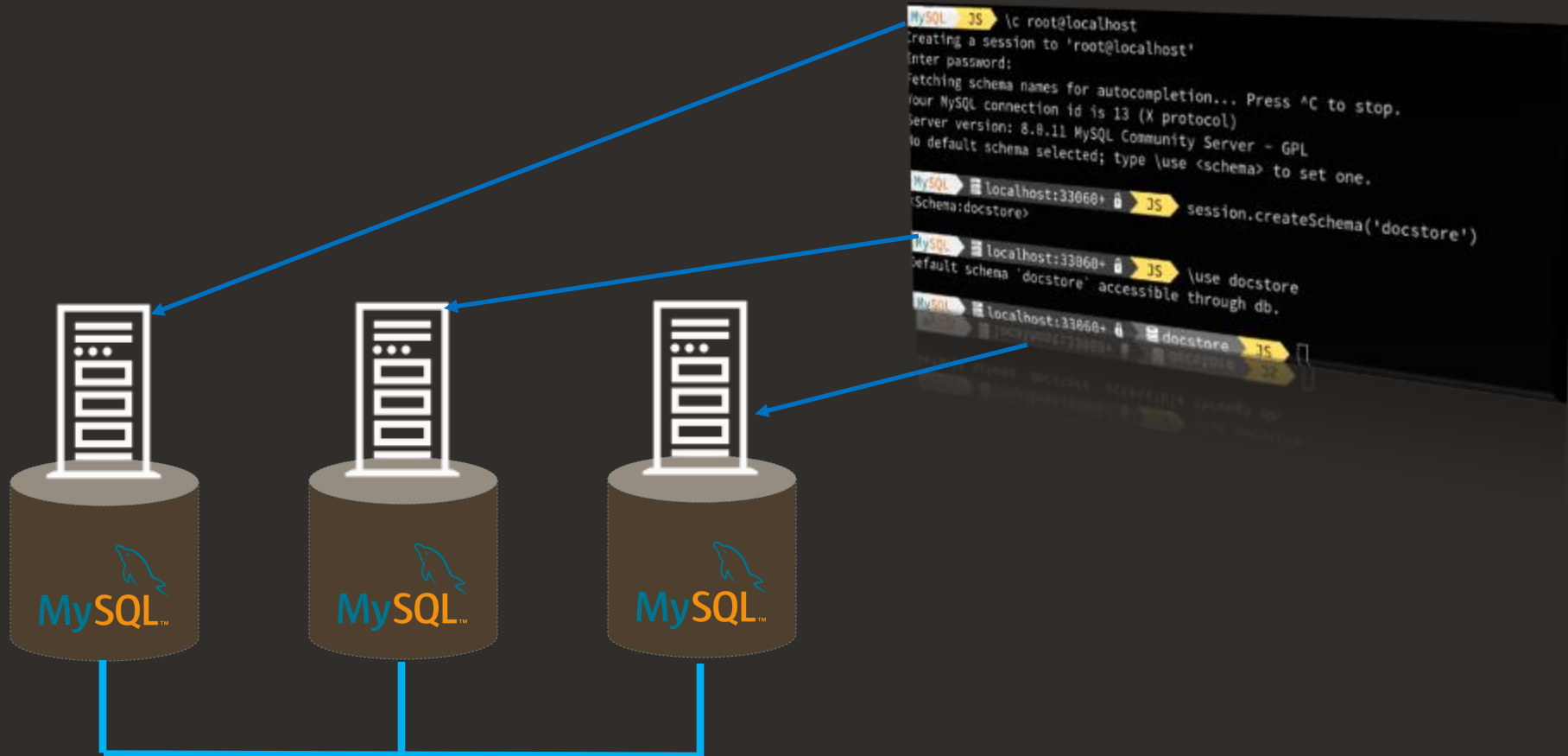
# MySQL InnoDB Cluster

## Installation

# InnoDB Cluster

## Installation

- ✓ Configure and create a cluster using MySQL Shell and 3 MySQL instances



# InnoDB Cluster

## Installation: minimal setup



1. Check the configuration of each instance:

- `MySQL JS > dba.checkInstanceConfiguration('root@host1')`

2. If it's not correct, prepare the instances for group replication :

- `MySQL JS > dba.configureInstance('root@host1')`

3. Connect to the first instance (the primary):

- `MySQL JS > \c root@host1`

4. Create a cluster:

- `MySQL host1 ... JS > cluster=dba.createCluster('prod_cluster');`

5/6. Add all other nodes using the cluster method. :

- `MySQL ... JS > cluster.addInstance('host2')`

- `MySQL ... JS > cluster.addInstance('host3')`

• *That's it!* 😊

- Now, you can check the cluster status and add the routers :

```
MySQL ... JS > cluster.status()
```

```
$> mysqlrouter --bootstrap root@host1
```

```
$> systemctl start mysqlrouter.service
```



# InnoDB Cluster

## Installation ClusterSet: minimal setup



### 1. Creating the ClusterSet:

- `MySQL JS` > `cs = cluster.createClusterSet('mydrsolution')`

### 2. Creating replicas :

- `MySQL JS` > `repl = cs.createReplicaCluster('repl_host1', 'repl_cluster')`

### 3. Add instances :

- `MySQL JS` > `repl.addInstance('repl_host2')`
- `MySQL JS` > `repl.addInstance('repl_host3')`

• *That's it!* 😊

### • You can now check the status of your ClusterSet :

```
MySQL ... JS > cs.status()
```

# Monitoring

## InnoDB Cluster Monitoring



# InnoDB Cluster

## Monitoring

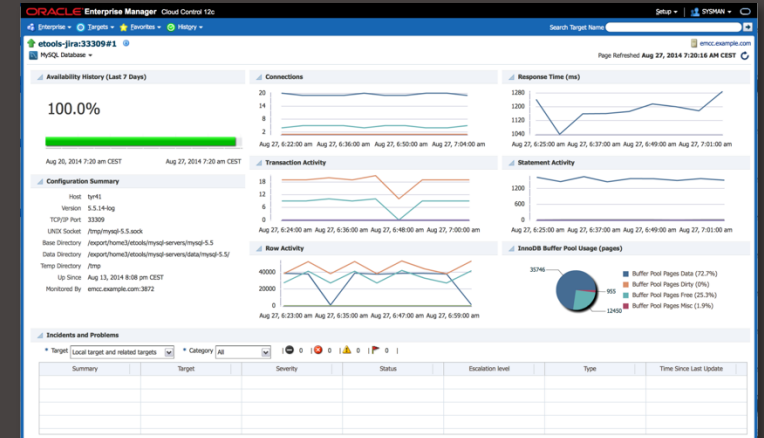


### MySQL Monitoring

Observabilité avancée :  
corrélation des traces,  
métriques et logs

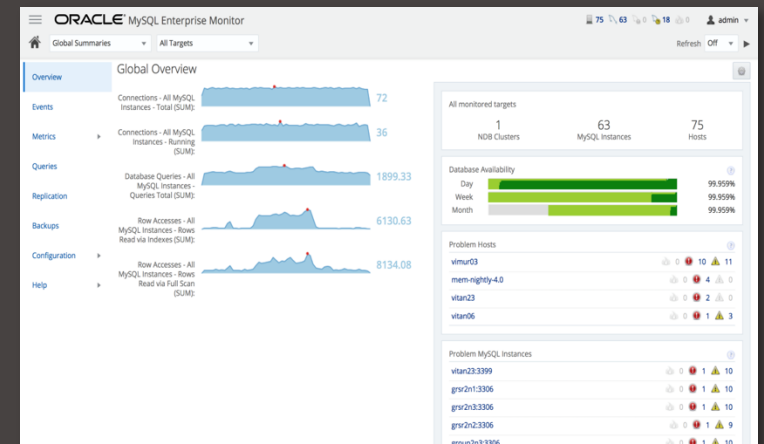
- **Oracle Enterprise Manager Plugin for MySQL**

- Availability and Performance monitoring
- Configuration monitoring
- Queries information
- Performance and configuration metrics
- Compliance Framework
- Metric Extension included



- **MySQL Enterprise Monitor**

- Availability and Performance monitoring
- Queries information
- Performance and configuration metrics



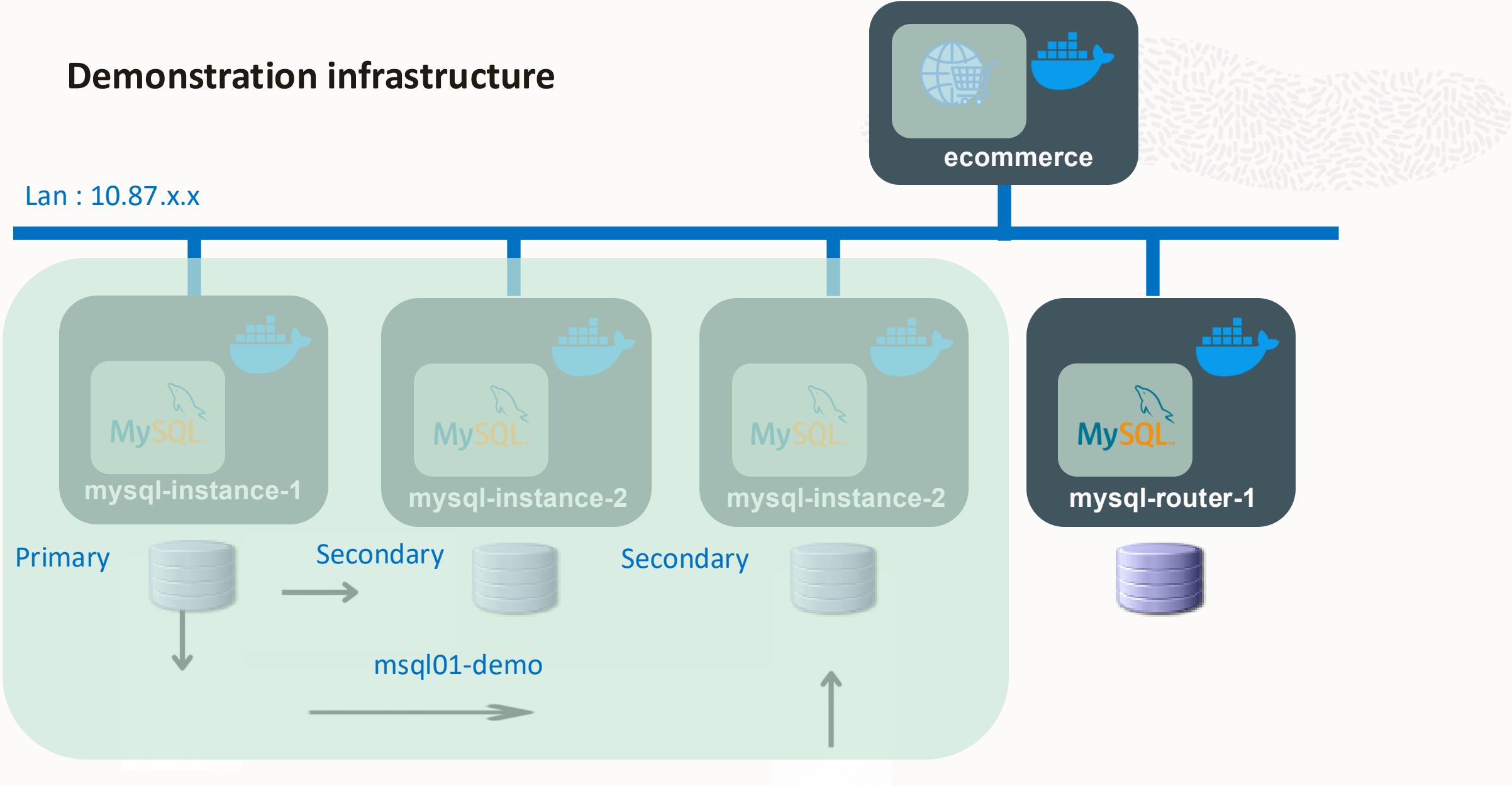


# Demonstration

---

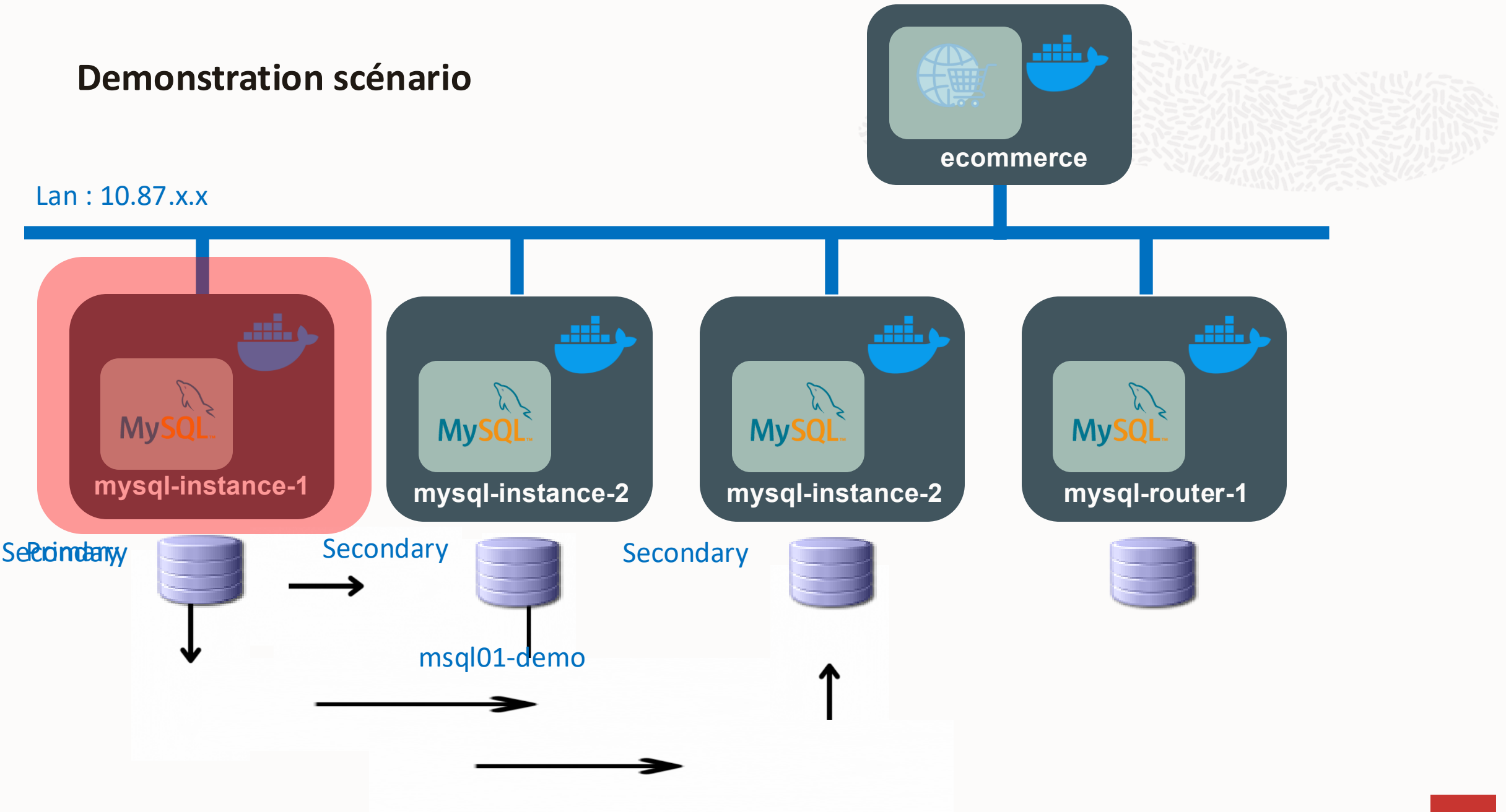


# Demonstration infrastructure





# Demonstration scénario



Lan : 10.87.x.x

Primary

Secondary

Secondary

msql01-demo



# Let's Get Hands-On.





## Key takeaways



# Resources

## Editions

<https://www.mysql.com/products/>

## Reference Manual

<https://dev.mysql.com/doc/refman/9.4/en/>

## Pre-Checking Instance Configuration for InnoDB Cluster Usage

<https://dev.mysql.com/doc/mysql-shell/9.4/en/check-instance-configuration.html>

## InnoDB Cluster

<https://dev.mysql.com/doc/mysql-shell/9.4/en/mysql-innodb-cluster.html>

## Replication Sets

<https://dev.mysql.com/doc/mysql-shell/9.4/en/mysql-innodb-replicaset.html>

## MySQL Diagnostic Information

<https://dev.mysql.com/doc/mysql-shell/9.4/en/mysql-shell-utilities-debug-diagnostics.html>

# Thank you

---

**Emmanuel COLUSSI**

[emmanuel.colussi@oracle.com](mailto:emmanuel.colussi@oracle.com)





ORACLE