High Availability MariaDB Clusters on AWS

Objective

- To demonstrate the implementation of a high-availability clustering using two approaches:
  1. Local setup Aws and Dbeaver/cli commands with mysqldump.
  2. AWS setup with Galera Manager.

Key Goals

- Ensure data consistency, replication, and automated backups.
- Compare and evaluate the two approaches and also explore others

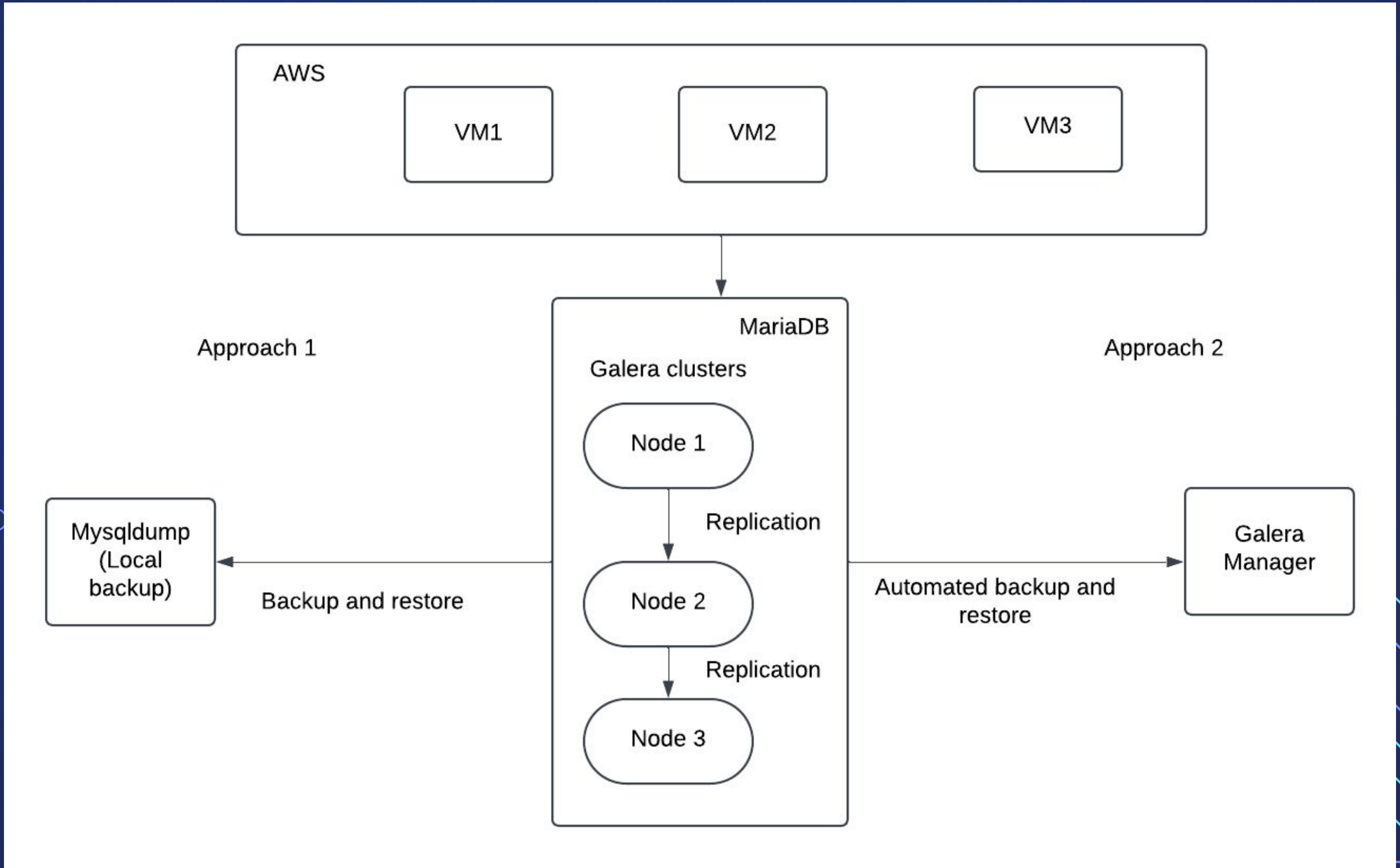
Dataset

- FIrst load- 10 MB file (SQL file to test basic replication)
- https://github.com/kite1988/nus-sms-corpus/blob/master/smsCorpus_en_sql_2015.03.09_all.zip

- Second load - 90MB
- https://www.kaggle.com/datasets/thedevastator/unlock-profits-with-e-commerce-sales-data

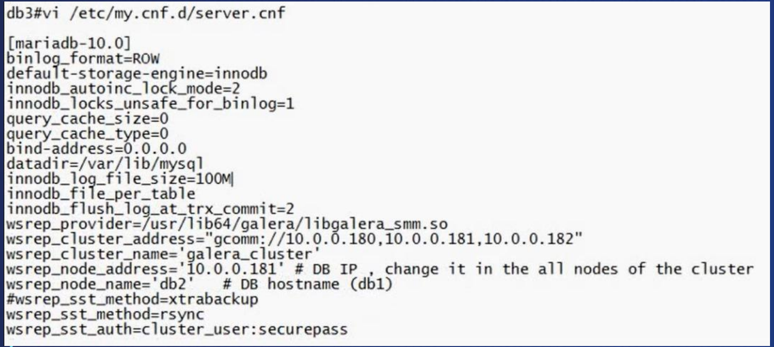
- Third Load- 180MB
- https://catalog.data.gov/dataset/crime-data-from-2020-to-present

System Architecture

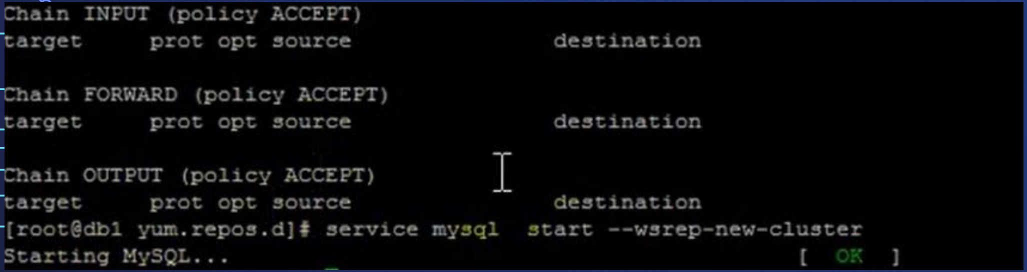


Local Setup

Configuration



```
db3#vi /etc/my.cnf.d/server.cnf

[mariadb-10.0]
binlog_format=ROW
default-storage-engine=innodb
innodb_autoinc_lock_mode=2
innodb_locks_unsafe_for_binlog=1
query_cache_size=0
query_cache_type=0
bind-address=0.0.0.0
datadir=/var/lib/mysql
innodb_log_file_size=100M
innodb_file_per_table
innodb_flush_log_at_trx_commit=2
wsrep_provider=/usr/lib64/galera/libgalera_smm.so
wsrep_cluster_address="gcomm://10.0.0.180,10.0.0.181,10.0.0.182"
wsrep_cluster_name='galera_cluster'
wsrep_node_address='10.0.0.181' # DB IP , change it in the all nodes of the cluster
wsrep_node_name='db2'   # DB hostname (db1)
#wsrep_sst_method=xtrabackup
wsrep_sst_method=rsync
wsrep_sst_auth=cluster_user:securepass
```

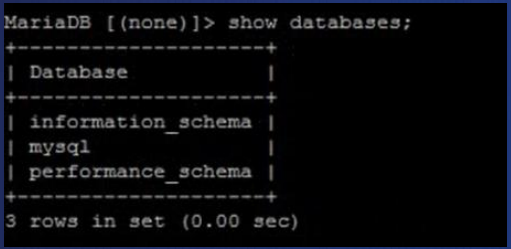


```
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
[root@db1 yum.repos.d]# service mysql  start --wsrep-new-cluster
Starting MySQL...                                          [  OK  ]
```

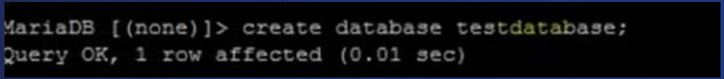
- 3 nodes configuration and replication using gcom//:
- Replaced hostname and Ip address
  - 3 active nodes configured with active status

Replications and Backup

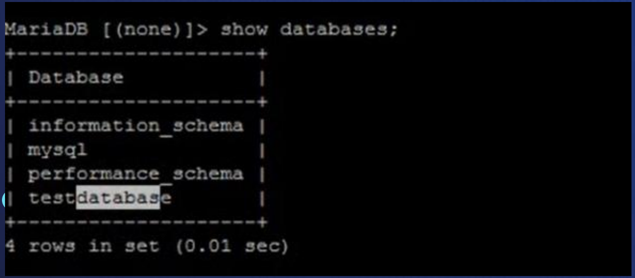
Replication in all 3 nodes (db1, db2, db3)



```
MariaDB [(none)]> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
+--------------------+
3 rows in set (0.00 sec)
```

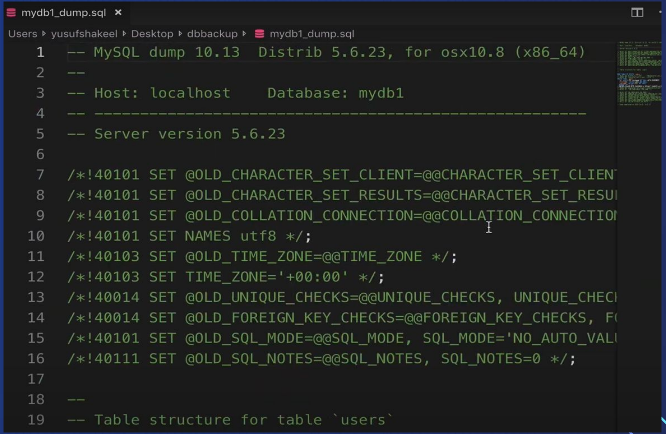


```
MariaDB [(none)]> create database testdatabase;
Query OK, 1 row affected (0.01 sec)
```



```
MariaDB [(none)]> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| testdatabase       |
+--------------------+
4 rows in set (0.01 sec)
```

Backup of all 3 nodes (db1, db2, db3) generated on local machine



```
-- MySQL dump 10.13  Distrib 5.6.23, for osx10.8 (x86_64)
--
-- Host: localhost    Database: mydb1
-- ------------------------------------------------------
-- Server version 5.6.23

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIEN
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESU
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTIO
/*!40101 SET NAMES utf8 */;
/*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
/*!40103 SET TIME_ZONE='+00:00' */;
/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHEC
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, F
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_VAL
/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;

--
-- Table structure for table `users`
```

Challenges with Manual Setup with MariaDB

Replication:

- High latency during bulk inserts.
- Nodes desynchronize under heavy load.

Backup:

- Manual backups are time-consuming and prone to inconsistencies.

Failover:

- Requires manual intervention, leading to downtime.

Summary:

- While feasible for small-scale use, manual setups cannot handle large datasets efficiently.

What happens when there is huge streams of data loading ?

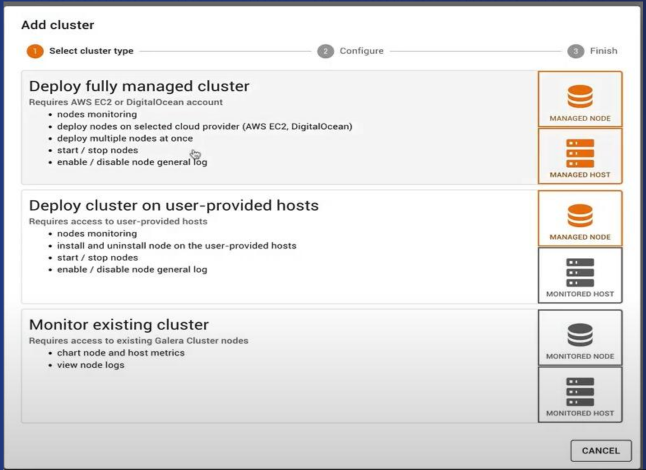
Galera Cluster Automation

Why Galera Manager?

- Automates cluster management and backups.
- Handles large datasets and high-concurrency workloads.
- Provides built-in monitoring and failover mechanisms.

Key Features:

- Automated backups to AWS S3.
- Incremental State Transfer (IST) for fast synchronization.
- Real-time monitoring and alerting.



Add cluster

1 Select cluster type — 2 Configure — 3 Finish

Deploy fully managed cluster

Requires AWS EC2 or DigitalOcean account

- nodes monitoring
- deploy nodes on selected cloud provider (AWS EC2, DigitalOcean)
- deploy multiple nodes at once
- start / stop nodes
- enable / disable node general log

Deploy cluster on user-provided hosts

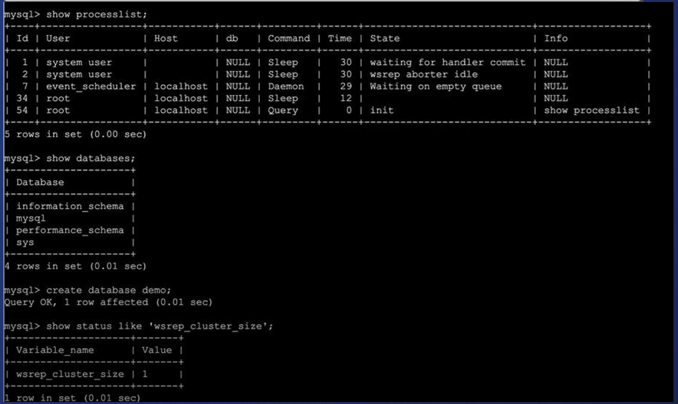
Requires access to user-provided hosts

- nodes monitoring
- install and uninstall node on the user-provided hosts
- start / stop nodes
- enable / disable node general log

Monitor existing cluster

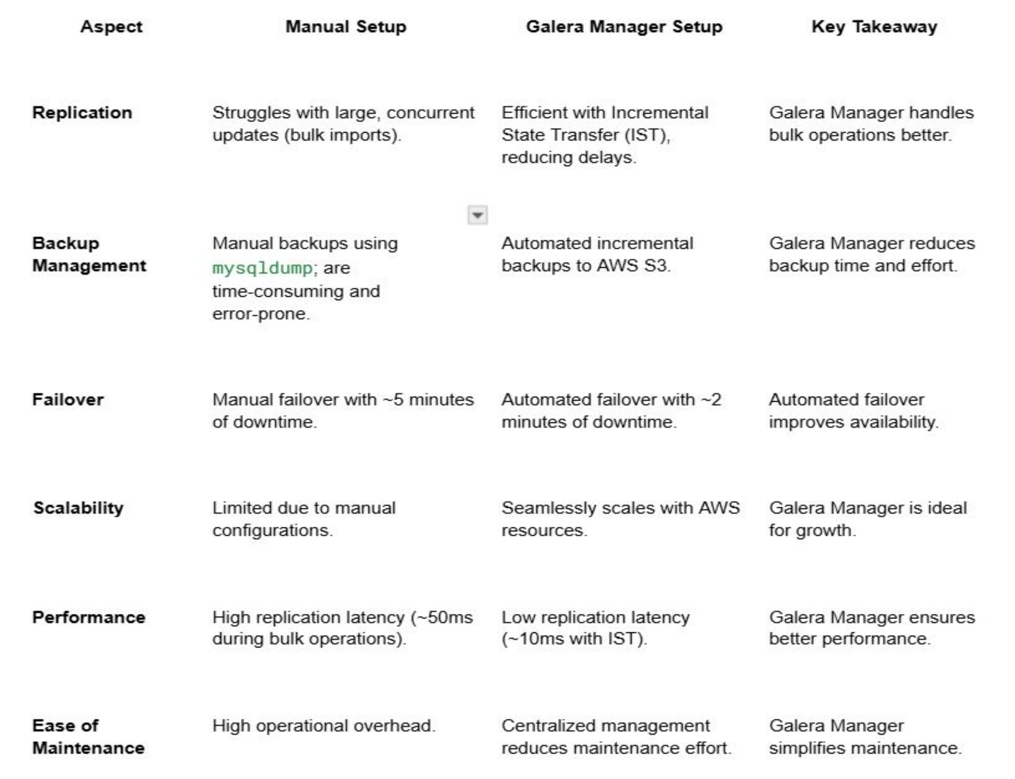
Requires access to existing Galera Cluster nodes

- chart node and host metrics
- view node logs

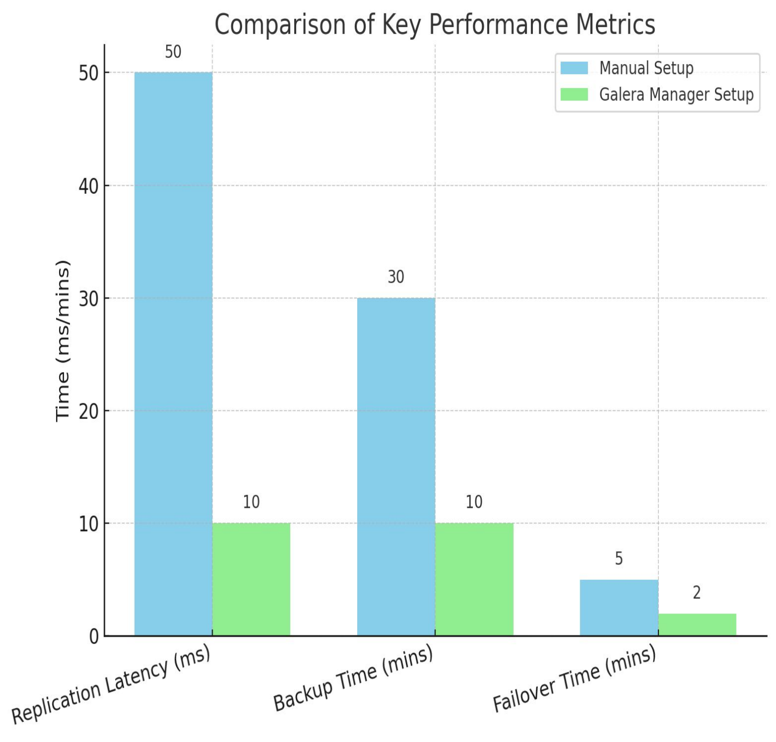


```
mysql> show processlist;
+----+-----------------+-----------+------+---------+------+---------------------------+------------------+
| Id | User            | Host      | db   | Command | Time | State                     | Info             |
+----+-----------------+-----------+------+---------+------+---------------------------+------------------+
|  1 | system user     |           | NULL | Sleep   |   30 | waiting for handler commit | NULL            |
|  2 | system user     |           | NULL | Sleep   |   30 | wsrep aborter idle        | NULL             |
|  7 | event_scheduler | localhost | NULL | Daemon  |   29 | Waiting on empty queue    | NULL             |
| 34 | root            | localhost | NULL | Sleep   |   12 |                           | NULL             |
| 54 | root            | localhost | NULL | Query   |    0 | init                      | show processlist |
+----+-----------------+-----------+------+---------+------+---------------------------+------------------+
5 rows in set (0.00 sec)

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
4 rows in set (0.01 sec)

mysql> create database demo;
Query OK, 1 row affected (0.01 sec)

mysql> show status like 'wsrep_cluster_size';
+--------------------+-------+
| Variable_name      | Value |
+--------------------+-------+
| wsrep_cluster_size | 1     |
+--------------------+-------+
1 row in set (0.01 sec)
```

Comparative analysis



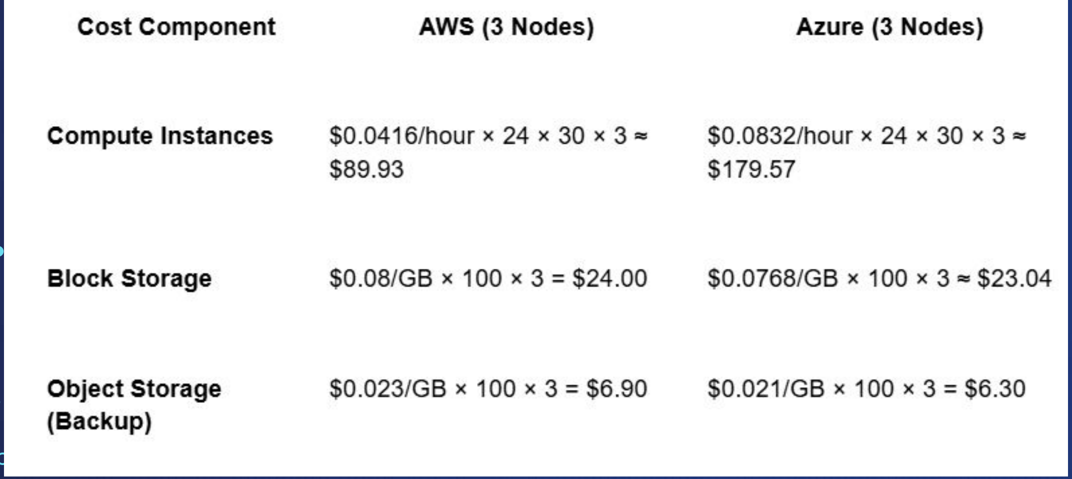
Aspect	Manual Setup	Galera Manager Setup	Key Takeaway
Replication	Struggles with large, concurrent updates (bulk imports).	Efficient with Incremental State Transfer (IST), reducing delays.	Galera Manager handles bulk operations better.
Backup Management	Manual backups using `mysqldump`; are time-consuming and error-prone.	Automated incremental backups to AWS S3.	Galera Manager reduces backup time and effort.
Failover	Manual failover with ~5 minutes of downtime.	Automated failover with ~2 minutes of downtime.	Automated failover improves availability.
Scalability	Limited due to manual configurations.	Seamlessly scales with AWS resources.	Galera Manager is ideal for growth.
Performance	High replication latency (~50ms during bulk operations).	Low replication latency (~10ms with IST).	Galera Manager ensures better performance.
Ease of Maintenance	High operational overhead.	Centralized management reduces maintenance effort.	Galera Manager simplifies maintenance.



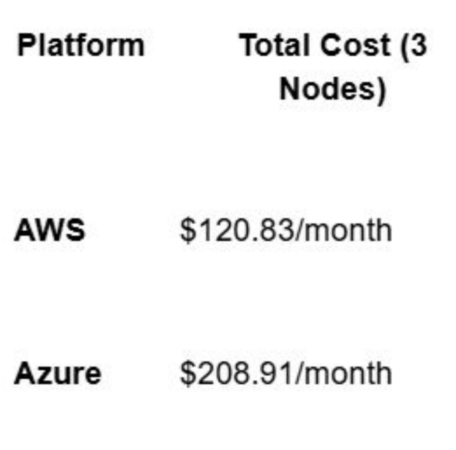
Comparison of Key Performance Metrics

Cost Analysis

AWS vs Azure



Cost Component	AWS (3 Nodes)	Azure (3 Nodes)
Compute Instances	$0.0416/hour × 24 × 30 × 3 ≈ $89.93	$0.0832/hour × 24 × 30 × 3 ≈ $179.57
Block Storage	$0.08/GB × 100 × 3 = $24.00	$0.0768/GB × 100 × 3 ≈ $23.04
Object Storage (Backup)	$0.023/GB × 100 × 3 = $6.90	$0.021/GB × 100 × 3 = $6.30



Platform	Total Cost (3 Nodes)
AWS	$120.83/month
Azure	$208.91/month

Conclusion and inferences

Primary Node Election: MariaDB Galera Cluster doesn't rely on a dedicated primary node. Failover requires a minimum of two nodes, with an arbitrator ensuring cluster continuity in critical scenarios.

Manual Setup Limitations: Suitable for small-scale setups but inefficient for large datasets due to replication delays, manual backups, and lack of automated failover.

Galera Manager Advantages: Automates cluster management, handles bulk imports via Incremental State Transfer (IST), and minimizes downtime with automated failover.

Final Takeaway: Galera Manager is essential for scalable and reliable enterprise databases, outperforming manual setups in efficiency and automation

References

- **Dataset-** https://github.com/kite1988/nus-sms-corpus/blob/master/smsCorpus_en_sql_2015.03.09_all.zip

  https://www.kaggle.com/datasets/thedevastator/unlock-profits-with-e-commerce-sales-data

  https://www.kaggle.com/datasets/manjeetsingh/retaildataset/data

- https://galeracluster.com/library/documentation/galera-manager.html

- https://dev.mysql.com/doc/refman/8.4/en/mysqldump.html

- https://galeracluster.com/library/documentation/install-mariadb.html

THANKS!