**MySQL InnoDB Cluster Setup: A High Availability Solution for MySQL 8.0**

**Environment:** Oracle Enterprise Linux 8

Introduction

MySQL InnoDB Cluster offers a high availability solution with single and multi-master capabilities and automatic failover. This guide details setting up a new MySQL InnoDB Cluster using MySQL Community Server 8.0.

Prerequisites

* 3 Linux (OEL) systems running.
* SSH access with sudo privileges on each system.
* Firewall configured to allow traffic on port 5566.

For a webb Application of 50k users. Below will be my hardware requirements for the Database server.

CPU: 16cores  
RAM: 32GB  
Storage: SSD 1TB plis  
Network: Gigabit network card

MySQL InnoDB Cluster Environment Details

| **IP** | **Hostname** | **Description** |
| --- | --- | --- |
| 192.168.56.2 | PGPRI | Master |
| 192.168.56.3 | PGSTB | Replica |
| 192.168.56.4 | PGBAK | Replica |

IP and Hostname Mapping

Add the following entries to **/etc/hosts** on each server for proper name resolution:

Vi /etc/hosts

192.168.56.2 PGPRI   
192.168.56.3 PGSTB   
192.168.56.4 PGBAK

# Fix potential DNS resolution issues on all 3 servers.

echo "nameserver 8.8.8.8" > /etc/resolv.conf

echo "nameserver 8.8.4.4" >> /etc/resolv.conf

#Add Port 5566 to mysqld\_t: Use the semanage command to add port 5566 to the SELinux type for MySQL (mysqld\_t):

sudo semanage port -a -t mysqld\_port\_t -p tcp 5566

**Setup Procedure**

Step 1: Add MySQL APT Repository

1. **On all servers**, download and install the MySQL APT repository:

wget <https://dev.mysql.com/get/mysql80-community-release-el8-1.noarch.rpm>

rpm -Uvh mysql80-community-release-el8-1.noarch.rpm

Step 2: Install MySQL Server and Shell

1. **Install MySQL Server and MySQL Shell on all servers**:

yum install -y mysql-community-server mysql-shell

Step 3: Configure MySQL

1. **Create a MySQL admin user on all servers**:

Log into MySQL:

mysql -u root -p

Create the admin user:

CREATE USER lab@'%' IDENTIFIED BY '\*\*\*\*\*\*\*;  
GRANT ALL PRIVILEGES ON \*.\* TO 'lab'@'%' WITH GRANT OPTION;  
 FLUSH PRIVILEGES;

1. **Change the bind address to allow remote connections**:

vi /etc/my.cnf.d/ mysql-server.cnf :

bind-address = 0.0.0.0

Restart MySQL to apply changes:

sudo systemctl restart mysqld

Step 4: Setup MySQL InnoDB Cluster

1. **Configure instances for InnoDB Cluster**:

On each server (PGPRI, PGSTB, PGBAK), configure MySQL instance:

mysqlsh

JS > dba.configureInstance('lab@PGPRI:5566')

JS > dba.configureInstance('lab@PGSTB:5566')

JS > dba.configureInstance('lab@PGBAK:5566'

1. **Create the InnoDB Cluster on PGPRI**:

Connect to the MySQL Shell on PGPRI:

mysqlsh --uri lab@PGPRI:5566

Create the cluster:

var cluster =dba.createCluster('LabCluster');

Add PGSTB and PGBAK to the cluster:

cluster.addInstance('lab@PGSTB:5566');

cluster.addInstance('lab@PGBAK:5566');

Verify the cluster status:  
  
var cluster = dba.getCluster('LabCluster');

cluster.status();

MySQL PGPRI:5566 ssl JS > cluster.status();

{

"clusterName": "LabCluster",

"defaultReplicaSet": {

"name": "default",

"primary": "PGPRI:5566",

"ssl": "REQUIRED",

"status": "OK",

"statusText": "Cluster is ONLINE and can tolerate up to ONE failure.",

"topology": {

"PGBAK:5566": {

"address": "PGBAK:5566",

"memberRole": "SECONDARY",

"mode": "R/O",

"readReplicas": {},

"replicationLag": "applier\_queue\_applied",

"role": "HA",

"status": "ONLINE",

"version": "8.0.36"

},

"PGPRI:5566": {

"address": "PGPRI:5566",

"memberRole": "PRIMARY",

"mode": "R/W",

"readReplicas": {},

"replicationLag": "applier\_queue\_applied",

"role": "HA",

"status": "ONLINE",

"version": "8.0.36"

},

"PGSTB:5566": {

"address": "PGSTB:5566",

"memberRole": "SECONDARY",

"mode": "R/O",

"readReplicas": {},

"replicationLag": "applier\_queue\_applied",

"role": "HA",

"status": "ONLINE",

"version": "8.0.36"

}

},

"topologyMode": "Single-Primary"

},

"groupInformationSourceMember": "PGPRI:5566"

}

Creating accounts for Developers, DBA, and a dedicated account for web application

CREATE ROLE 'dba';

CREATE ROLE 'devs';

CREATE ROLE 'applications';  
  
GRANT SELECT ON music.\* TO 'dev';

GRANT ALL PRIVILEGES ON \*.\* TO 'dba';

GRANT SELECT ON music.\* TO 'applications';

FLUSH PRIVILEGES;

CREATE USER 'web\_app'@'%' IDENTIFIED BY 'Lab@d1';

CREATE USER 'dba\_1'@'%' IDENTIFIED BY 'Lab@d1';

CREATE USER 'dba\_2'@'%' IDENTIFIED BY 'Lab@d1';

CREATE USER 'dba\_3'@'%' IDENTIFIED BY 'Lab@d1';

CREATE USER 'dev\_user1'@'%' IDENTIFIED BY 'Lab@d1';

CREATE USER 'dev\_user2'@'%' IDENTIFIED BY 'Lab@d2';

CREATE USER 'dev\_user3'@'%' IDENTIFIED BY 'Lab@d3';

CREATE USER 'dev\_user4'@'%' IDENTIFIED BY 'Lab@d4';

CREATE USER 'dev\_user5'@'%' IDENTIFIED BY 'Lab@d5';

CREATE USER 'dev\_user6'@'%' IDENTIFIED BY 'Lab@d6';

CREATE USER 'dev\_user7'@'%' IDENTIFIED BY 'Lab@d7';

CREATE USER 'dev\_user8'@'%' IDENTIFIED BY 'Lab@d8';

CREATE USER 'dev\_user9'@'%' IDENTIFIED BY 'Lab@d9';

CREATE USER 'dev\_user10'@'%' IDENTIFIED BY 'Lab@10';

SET DEFAULT ROLE 'application' TO 'webb\_app'@'%';

SET DEFAULT ROLE 'dba' TO 'dba\_1'@'%';

SET DEFAULT ROLE 'dba' TO 'dba\_2'@'%';

SET DEFAULT ROLE 'dba' TO 'dba\_3'@'%';

SET DEFAULT ROLE 'devs' TO 'dev\_user1'@'%';

SET DEFAULT ROLE 'devs' TO 'dev\_user2'@'%';

SET DEFAULT ROLE 'devs' TO 'dev\_user3'@'%';

SET DEFAULT ROLE 'devs' TO 'dev\_user4'@'%';

SET DEFAULT ROLE 'devs' TO 'dev\_user5'@'%';

SET DEFAULT ROLE 'devs' TO 'dev\_user6'@'%';

SET DEFAULT ROLE 'devs' TO 'dev\_user7'@'%';

SET DEFAULT ROLE 'devs' TO 'dev\_user8'@'%';

SET DEFAULT ROLE 'devs' TO 'dev\_user9'@'%';

SET DEFAULT ROLE 'devs' TO 'dev\_user10'@'%';

GRANT 'dba' TO 'dba\_1'@'%';

GRANT 'dba' TO 'dba\_2'@'%';

GRANT 'dba' TO 'dba\_3'@'%';

GRANT 'dev' TO 'dev\_user1'@'%';

GRANT 'dev' TO 'dev\_user2'@'%';

GRANT 'dev' TO 'dev\_user3'@'%';

GRANT 'dev' TO 'dev\_user4'@'%';

GRANT 'dev' TO 'dev\_user5'@'%';

GRANT 'dev' TO 'dev\_user6'@'%';

GRANT 'dev' TO 'dev\_user7'@'%';

GRANT 'dev' TO 'dev\_user8'@'%';

GRANT 'dev' TO 'dev\_user9'@'%';

GRANT 'dev' TO 'dev\_user10'@'%'