MySQL Replication between two servers

Disclaimer:- *This materials designed for educational purposes only,Ideas,contents are my own not to ORACLE Corp*.

In this tutorial our focus to demonstrate the functionality of Replica set only to use in testing purpose

Sandbox instances are only suitable for deploying and running on your local machine for testing purposes. In a production environment, the MySQL Server instances are deployed to various host machines on the network.

Master	Slave
10.0.0.158 :3310	10.0.0.158:3320

Prerequisite

##Make sure below component already install or else first install

1. MySQL Server
2. MySQL Shell

```
#rpm -Uhv /home/opc/bootcamp/MySQLShell/mysql-shell-commercial-8.0.30-1.1.el7.x86_64.rpm
```

3. MySQL Router

```
#rpm -Uhv /home/opc/bootcamp/MySQLRouter/mysql-router-commercial-8.0.30-1.1.el7.x86_64.rpm
```

In what kind of scenarios MySQL Replica Set is Recommended ?

Below are top Features which makes life of DBA simple:-

- ✓ To scale Read workloads.
- ✓ Manual failover in event of primary node goes down.
- ✓ Useful where we can compromise RPO/RTO time.
- ✓ MySQL Shell Automatically configures users and Replication.
- ✓ Easy to deploy without editing into my.cnf/my.ini file.
- ✓ Not to spend time on Backup →Restore to provision new node , MySQL CLONE feature in-built which will save a lot time to bring another server for replication. More on Cloning:- https://mysqlserverteam.com/clone-create-mysql-instance-replica/
- ✓ Integrated MySQL Router Load balancing .
- ✓ Easy to getting started into MySQL high availability for all tier type applications.

How to configure and deploy MySQL Replica Set

```
#sudo su root
#mysqlsh -uroot -pMySQL8.0
MySQL JS>

######Deploy Sandbox Instance , below two command ONLY for LEARNING/Development
NOT to use in Production

MySQL JS>dba.deploySandboxInstance(3310)
MySQL JS>dba.deploySandboxInstance(3320)
```

Configure Machine to participate into MySQL InnoDB Replica Set

```
MySQL JS>dba.configureReplicaSetInstance("root@127.0.1:3310")
MySQL JS>dba. configureReplicaSetInstance("root@127.0.1:3320")
```

Above command will validate and fix the configuration parameters

```
NOTE: Some configuration options need to be fixed:
+----------------------------------------+----------------+----------------+--------------------------------------------------+
| Variable                               | Current Value  | Required Value | Note                                             |
+----------------------------------------+----------------+----------------+--------------------------------------------------+
| binlog_transaction_dependency_tracking | COMMIT_ORDER   | WRITESET       | Update the server variable                       |
| enforce_gtid_consistency               | OFF            | ON             | Update read-only variable and restart the server |
| gtid_mode                              | OFF            | ON             | Update read-only variable and restart the server |
| server_id                              | 1              | <unique ID>    | Update read-only variable and restart the server |
+----------------------------------------+----------------+----------------+--------------------------------------------------+

Some variables need to be changed, but cannot be done dynamically on the server.
ERROR: workshop-8-0-30:3306: Instance must be configured and validated with dba.configureReplicaSetInstance() before it
can be used in a replicaset.
Dba.createReplicaSet: Instance check failed (MYSQLSH 51150)
```

Create Replica Set and Add database node to form Replica Set.

```
shell.connect("root@127.0.1:3310");
#mysqlsh -h127.0.1  -uroot -pMySQL8.0

var rs = dba.createReplicaSet("MyReplicatSet")

rs.addInstance("root@127.0.1:3320");


rs.status();
```

Configure Router to talk from App to Replica Set.

```
#mysqlrouter --bootstrap root@127.0.0.1:3310 --user=root --directory
myrouter

Please enter MySQL password for root:
# Bootstrapping MySQL Router instance at '/home/opc/myrouter'...

- Creating account(s) (only those that are needed, if any)
- Verifying account (using it to run SQL queries that would be run by Router)
- Storing account in keyring
- Adjusting permissions of generated files
- Creating configuration /home/opc/myrouter/mysqlrouter.conf

# MySQL Router configured for the InnoDB ReplicaSet 'MyReplicatSet'

After this MySQL Router has been started with the generated configuration

    $ mysqlrouter -c /home/opc/myrouter/mysqlrouter.conf

InnoDB ReplicaSet 'MyReplicatSet' can be reached by connecting to:

## MySQL Classic protocol

- Read/Write Connections: localhost:6446
- Read/Only Connections:  localhost:6447

## MySQL X protocol

- Read/Write Connections: localhost:6448
- Read/Only Connections:  localhost:6449

[root@instance-master opc]#
[root@instance-master opc]#
```

```
########Start MySQL Router

# myrouter/start.sh
```

Using Replica Set

```
mysqlsh

MySQL JS>
#shell.connect("root@127.0.0.1:6446");
\sql
MySQL SQL> select @@port;
```

```
CREATE DATABASE sales;USE sales;
CREATE TABLE if not exists sales.employee(empid int primary key auto_increment,empname
varchar(100),salary int,deptid int);
```

```
INSERT sales.employee(empname,salary,deptid) values('Ram',1000,10);
```

```
INSERT sales.employee(empname,salary,deptid) values('Raja',2000,10);
```

```
INSERT sales.employee(empname,salary,deptid) values('Sita',3000,20);
SELECT * FROM  sales.employee;
```

Connect Router to another machine to verify changes.

```
mysqlsh

JS>
shell.connect("root@127.0.0.1:6447");
\sql
```

```
SQL>SELECT * FROM sales.employee;
INSERT sales.employee values(100,'Ram',1000,10);
<Error> because this machine is not allowed to execute DML,DDL statements.>
```

Ingest data into the Replica Set

```
while [ 1 ]; do sleep 1; mysql -h127.0.0.1 -uroot -pMySQL8.0 -P6446 -e "INSERT
sales.employee(empname,salary,deptid) values('Ram',1000,10); select count(*) from
sales.employee"; done
```

Failure Scenarios :- Site Failure (primary node down)

```
ps -ef|grep "mysqld"
kill -9 $(ps aux|grep 'mysqld'|grep 3310|awk '{print $2}')
```

```
shell.connect("root@localhost:3320")
rs=dba.getReplicaSet()
rs.forcePrimaryInstance("root@127.0.1:3320")
rs.status()
```

##Record mismatch in both table

```
Instance-1
shell.connect("root@127.0.0.1:3310")
\sql select * from sales.employee;
Instance-2

shell.connect("root@127.0.0.1:3320")
\sql select * from sales.employee;
```

Rejoin the Instance (here process is manual)

```
MySQL JS>rs.rejoinInstance("root@127.0.0.1:3310")
rs.status()

\sql select * from sales.employee;
```

#2 Switch Back from Instance-2 to Instance-1 (DR to DC)

```
dba.startSandboxInstance(3310)
rs=dba.getReplicaSet()
rs.setPrimaryInstance("root@127.0.0.1:3310")
rs.status()
```