MySQL Replication

Introduction

Replication enables data from one MySQL database server (known as a source) to be copied to one or more MySQL database servers (known as replicas).

In this lab you will prepare and create a replica.

Estimated Lab Time: 25 minutes

Objectives

In this lab, you will:

- Install binaries on a new server
- Have that server act as a Replica

Task 1: Prepare Replica (mysql2)

> **Note:**

- Server: mysql2
- mysql2 doesn't have binaries installed, so first part of the lab installs them. This is the same to first installation of mysql-advanced on mysql1

1. Open an SSH client to app-srv

```
<span style="color:green">shell></span> <copy>ssh -i <private_key_file>
opc@<your_compute_instance_ip></copy>
```

2. Connect to mysql2 server through app-srv

```
<span style="color:green">shell-app-srv$</span> <copy>ssh -i
$HOME/sshkeys/id_rsa_mysql2 opc@mysql2</copy>
```

3. Execute below script that replicate what we did in manual installation lab (create mysqluser/mysqlgrp, folders and install binaries)

```
<span style="color:green">shell-mysql2></span>
<copy>/workshop/support/MySQL_Replication___Prepare_replica_server.sh</copy>
```

```

```

```
<span style="color:green">shell-mysql2></span> <copy>sudo ls -l
/mysql</copy>
```

4. Close and reopen the SSH connection to mysql2 server to let opc user have the new group.

```
<span style="color:green">shell-mysql2></span> <copy>exit</copy>
```

```
<span style="color:green">shell-app-srv$</span> <copy>ssh -i
$HOME/sshkeys/id_rsa_mysql2 opc@mysql2</copy>
```

Task 2: Create Replica

> **Note:**

- Servers:
  - mysql1 (source)
  - mysql2 (replica)
- Some commands must run inside the source, other on replica: please read carefully the instructions

1. If not already connected, open two connections, one to mysql1 and one to mysql2

```
<span style="color:green">shell-app-srv$</span> <copy>ssh -i
$HOME/sshkeys/id_rsa_mysql1 opc@mysql1</copy>
```

And in a second shell

```
<span style="color:green">shell-app-srv$</span> <copy>ssh -i
$HOME/sshkeys/id_rsa_mysql2 opc@mysql2</copy>
```

2. mysql1 (source): Create a backup of source in the shared folder to easily restore on the replica:
   - take a full backup of the source using MySQL Enterprise Backup in your NFS folder /workshop/backups:

```
<span style="color:green">shell-mysql1></span> <copy>mysqlbackup --port=3307
--host=127.0.0.1 --user=mysqlbackup --password --backup-
dir=/workshop/backups/mysql1_source backup</copy>
```

3. mysql2 (replica): We now create the my.cnf for second instance.
   Please note that the content is like the one on mysql1, except for server_id: it's mandatory that each

server in a replication topology have a unique server id

```
<span style="color:green">shell-mysql2></span> <copy>sudo cp
/workshop/support/my.cnf.mysql2 /mysql/etc/my.cnf</copy>
```

```
 ```
```

shell-mysql2> sudo nano /mysql/etc/my.cnf

```

```

shell-mysql2> sudo chown mysqluser:mysqlgrp /mysql/etc/my.cnf

```

```

4. mysql2 (replica): Then restore the backup
    - restore the backup from share folder

```
shell-mysql2> <copy>sudo /mysql/mysql-
latest/bin/mysqlbackup --defaults-file=/mysql/etc/my.cnf --backup-
dir=/workshop/backups/mysql1_source copy-back-and-apply-log</copy>
```

```
shell-mysql2> <copy>sudo chown -R
mysqluser:mysqlgrp /mysql</copy>
```

    - start the new replica instance it and verify that it works (hostname have to be mysql2).

      Don’t make any change to the instance content!

```
shell-mysql2> <copy>sudo systemctl
start mysqld-advanced</copy>
```

```
shell-mysql2> <copy>mysqlsh
admin@mysql2:3307</copy>
```
```

```
<span style="color:blue">mysql-replica></span> <copy>SELECT @@hostname;
</copy>
```

```
<span style="color:blue">mysql-replica></span> <copy>SHOW DATABASES;
</copy>
```

5. mysql1 (source): Create the replication user

```
<span style="color:green">shell-mysql1></span> <copy>mysqlsh
admin@mysql1:3307</copy>
```

```
<span style="color:blue">mysql-source></span> <copy>CREATE USER 'repl'@'%'
IDENTIFIED BY 'Welcome1!' REQUIRE SSL;</copy>
```

```
<span style="color:blue">mysql-source></span> <copy>GRANT REPLICATION SLAVE
ON *.* TO 'repl'@'%';</copy>
```

6. mysql2 (replica): Time to connect and start the replica

   - Configure the replica connection.

```
<span style="color:blue">mysql-replica></span> <copy>CHANGE REPLICATION
SOURCE TO SOURCE_HOST='mysql1', SOURCE_PORT=3307,SOURCE_USER='repl',
SOURCE_PASSWORD='Welcome1!', SOURCE_AUTO_POSITION=1, SOURCE_SSL=1;</copy>
```

   - Start the replica threads

```
<span style="color:blue">mysql-replica></span> <copy>START REPLICA;</copy>
```

   - Verify replica status, e.g. that IO_Thread and SQL_Thread are started searching the value with the following command (in case of problems check error log)

```
<span style="color:blue">mysql-replica></span> <copy>SHOW REPLICA
STATUS\G</copy>
```

7. mysql1 (source): Let's test that data are replicated. Connect to source and make some changes

```
<span style="color:blue">mysql-source></span> <copy>CREATE DATABASE newdb;
</copy>
```

```
<span style="color:blue">mysql-source></span> <copy>USE newdb;</copy>
```

```
<span style="color:blue">mysql-source></span> <copy>CREATE TABLE t1 (c1 int
primary key);</copy>
```

```
<span style="color:blue">mysql-source></span> <copy>INSERT INTO t1
VALUES(1);</copy>
```

```
<span style="color:blue">mysql-source></span> <copy>INSERT INTO t1
VALUES(2);</copy>
```

```
<span style="color:blue">mysql-source></span> <copy>DROP DATABASE employees;
</copy>
```

8. mysql2 (replica): Verify that the new database and table is on the replica, to do so connect to replica and submit

```
<span style="color:blue">mysql-replica></span> <copy>SHOW DATABASES;</copy>
```

```
<span style="color:blue">mysql-replica></span> <copy>SELECT * FROM newdb.t1;
</copy>
```

Acknowledgements

- **Author** - Marco Carlessi, Principal Sales Consultant
- **Contributors** - Perside Foster, Principal Sales Consultant, Selena Sánchez, MySQL Solutions Engineer
- **Last Updated By/Date** - Perside Foster, Partner Solutions Engineer, March 2025