Batch - T7
Assignment No. - 8
Title - Configuration of MySQL for distributed databases
Student Name - Sharaneshwar Bharat Punjal
Student PRN - 23520011

1. Do the installation and configuration of Oracle/MySQL/IBM DB2 distributed databases. [Take minimum 3 nodes]
2. Use the above installed distributed database as backend for Assignment No. 3 & 4.
3. Demonstrate the working by connecting portals to different nodes, adding data from one node and available / display on portal connected to another node etc.

MySQL Distributed Database Setup on Windows

Prerequisites

Ensure we have the following on all three machines:

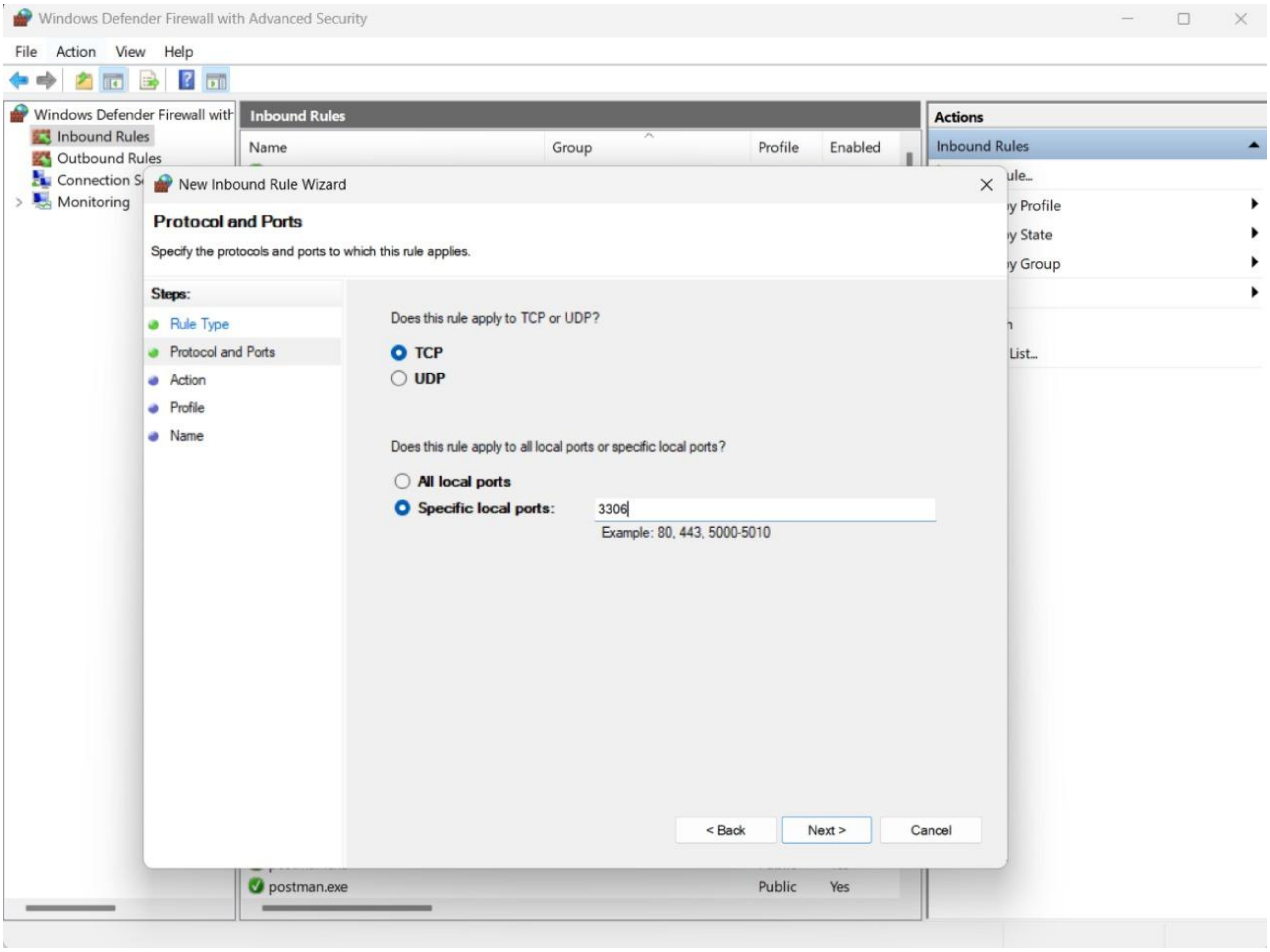
- MySQL Server installed.
- Network connectivity between the machines.
- Administrator privileges to modify MySQL configurations.

Install MySQL on All Three Machines

1. Download and Install MySQL:
   - Go to: MySQL Downloads.
   - Download the MySQL Installer.
   - Select MySQL Server and Workbench during installation.
2. Verify Installation:
   - Open Command Prompt.
   - Run: mysql --version

Network Configuration

- Assign Static IPs to the three machines:
  - Master: 192.168.1.10
  - Slave 1: 192.168.1.11
  - Slave 2: 192.168.1.12
- Allow MySQL Through Firewall:
  - Go to Windows Defender Firewall → Advanced Settings → Inbound Rules.
  - Add a new rule to allow port 3306 for MySQL.



New Inbound Rule Wizard

Protocol and Ports

Specify the protocols and ports to which this rule applies.

Steps:
- Rule Type
- Protocol and Ports
- Action
- Profile
- Name

Does this rule apply to TCP or UDP?

● TCP
○ UDP

Does this rule apply to all local ports or specific local ports?

○ All local ports
● Specific local ports: 3306

Example: 80, 443, 5000-5010

Configure the Master Node (Machine 1)

Step 1: Modify MySQL Configuration

1. Open MySQL configuration file:
   Location: C:\ProgramData\MySQL\MySQL Server x.x\my.ini
2. Add the following configuration under [mysqld]:
   ```
   [mysqld]
   server-id = 1
   log-bin = mysql-bin
   binlog-do-db = distributed_db
   ```
3. Restart MySQL Service:
   ```
   net stop mysql
   net start mysql
   ```

Step 2: Create a Replication User

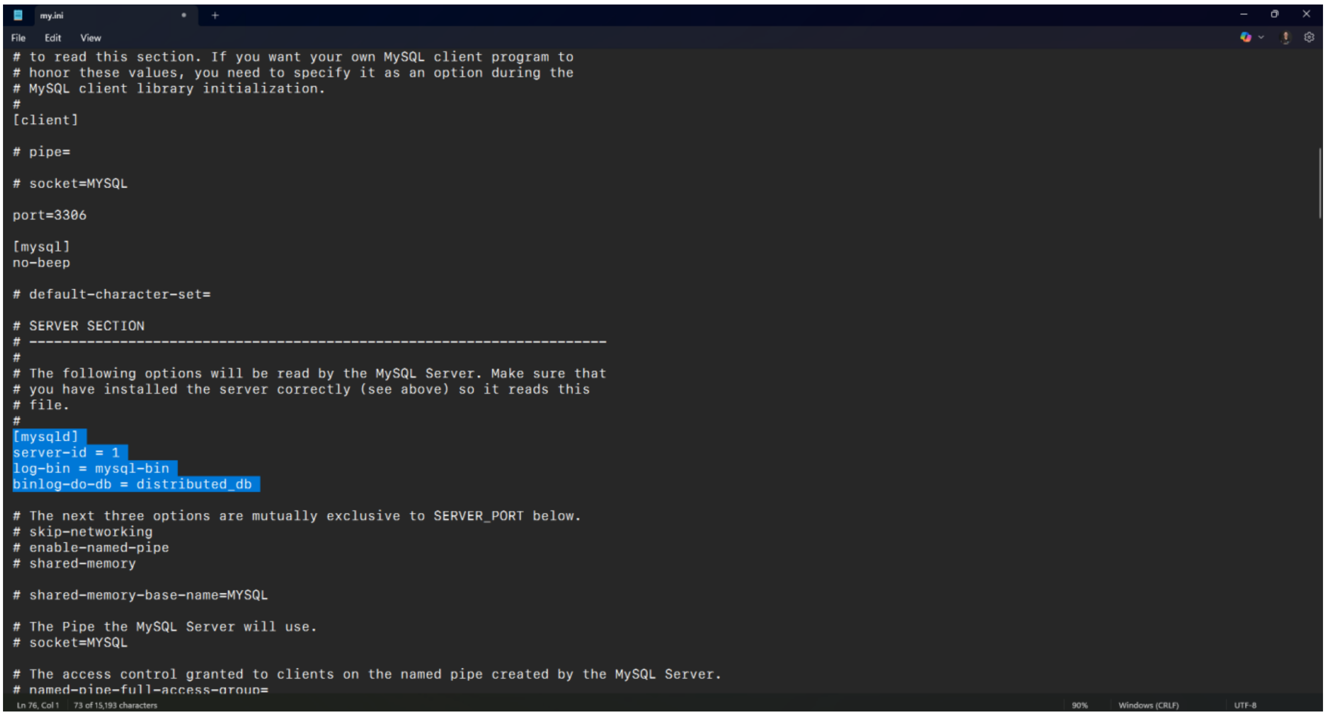
1. Open MySQL CLI: mysql -u root -p
2. Create a replication user:
   ```
   CREATE USER 'replicator'@'%' IDENTIFIED BY 'password';
   GRANT REPLICATION SLAVE ON *.* TO 'replicator'@'%';
   ```

FLUSH PRIVILEGES;

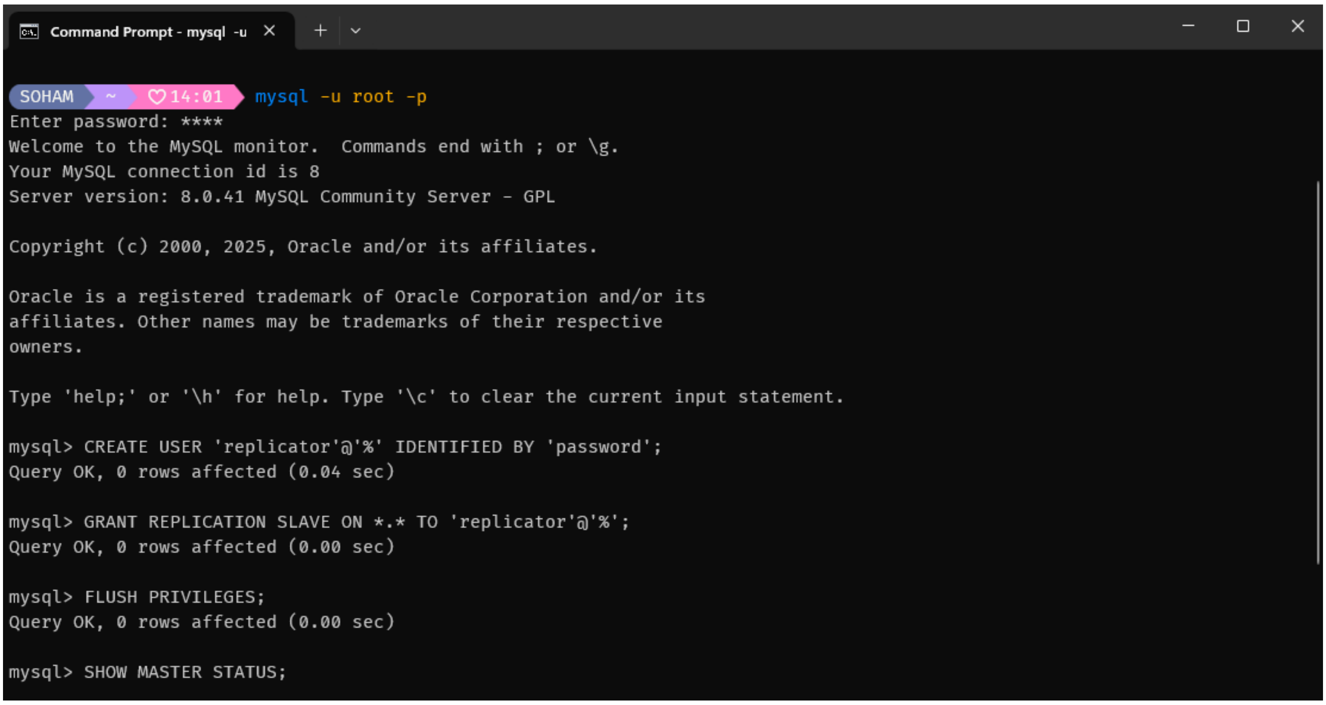
Step 3. Get the Master Status

Run the following command to get the master log file and position:

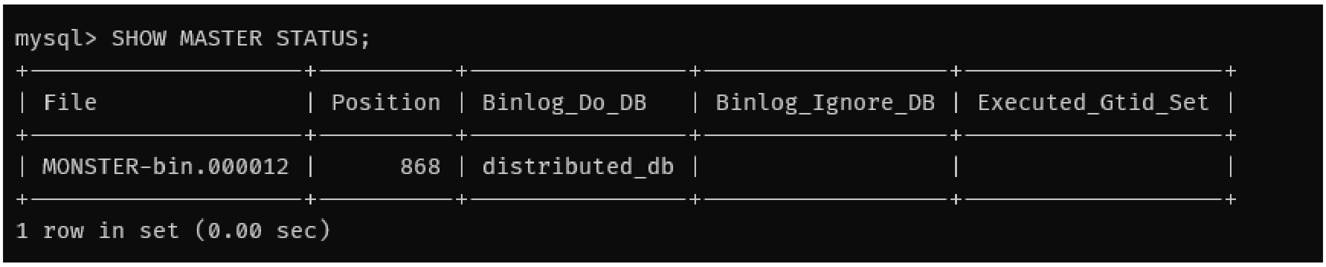
SHOW MASTER STATUS;



```
# to read this section. If you want your own MySQL client program to
# honor these values, you need to specify it as an option during the
# MySQL client library initialization.
#
[client]

# pipe=

# socket=MYSQL

port=3306

[mysql]
no-beep

# default-character-set=

# SERVER SECTION
# ----------------------------------------------------------------------
#
# The following options will be read by the MySQL Server. Make sure that
# you have installed the server correctly (see above) so it reads this
# file.
#
[mysqld]
server-id = 1
log-bin = mysql-bin
binlog-do-db = distributed_db

# The next three options are mutually exclusive to SERVER_PORT below.
# skip-networking
# enable-named-pipe
# shared-memory

# shared-memory-base-name=MYSQL

# The Pipe the MySQL Server will use.
# socket=MYSQL

# The access control granted to clients on the named pipe created by the MySQL Server.
# named-pipe-full-access-group=
```



```
SOHAM  ~  ♡14:01  mysql -u root -p
Enter password: ****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.41 MySQL Community Server - GPL

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> CREATE USER 'replicator'@'%' IDENTIFIED BY 'password';
Query OK, 0 rows affected (0.04 sec)

mysql> GRANT REPLICATION SLAVE ON *.* TO 'replicator'@'%';
Query OK, 0 rows affected (0.00 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.00 sec)

mysql> SHOW MASTER STATUS;
```



```
mysql> SHOW MASTER STATUS;
+---------------------+----------+----------------+------------------+-------------------+
| File                | Position | Binlog_Do_DB   | Binlog_Ignore_DB | Executed_Gtid_Set |
+---------------------+----------+----------------+------------------+-------------------+
| MONSTER-bin.000012 |      868 | distributed_db |                  |                   |
+---------------------+----------+----------------+------------------+-------------------+
1 row in set (0.00 sec)
```

Configure Slave 1 (Machine 2)

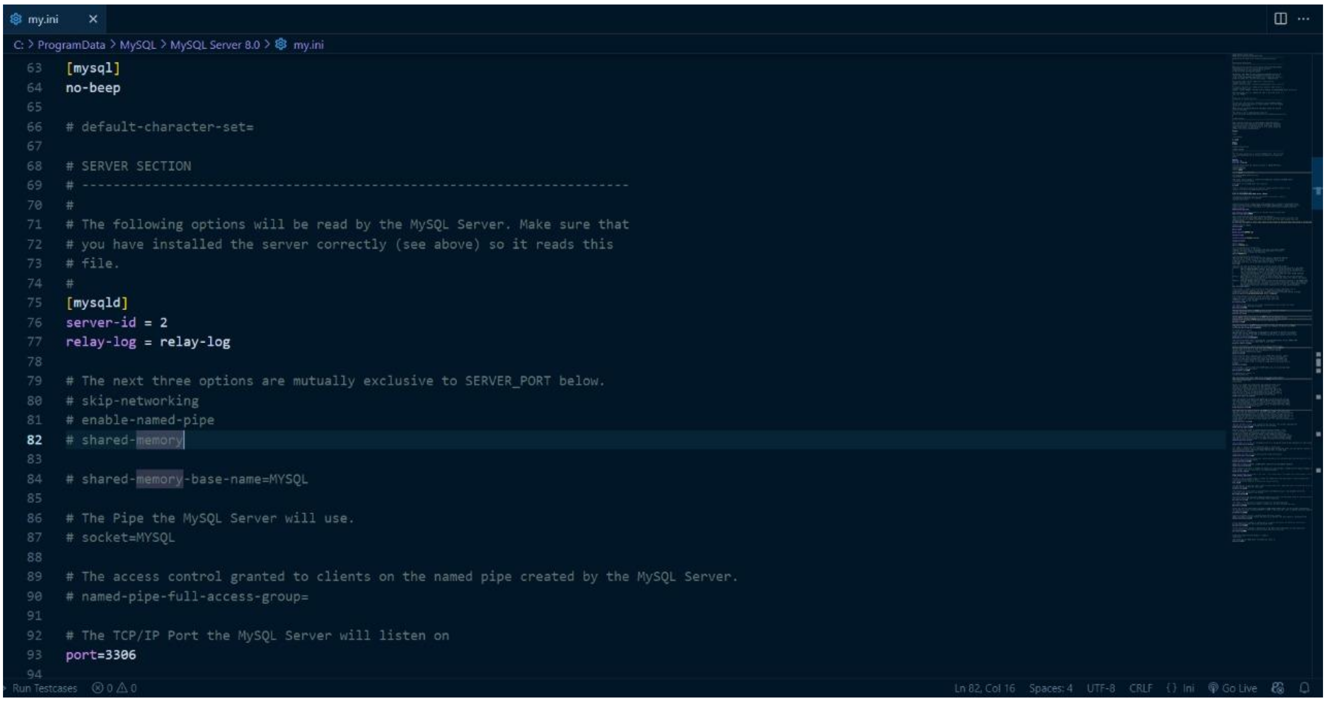
Step 1: Modify MySQL Configuration

1. Open my.ini file: C:\ProgramData\MySQL\MySQL Server x.x\my.ini
2. Add the following configuration:
   ```
   [mysqld]
   server-id = 2
   relay-log = relay-log
   ```
3. Restart MySQL Service:
   ```
   net stop mysql
   net start mysql
   ```

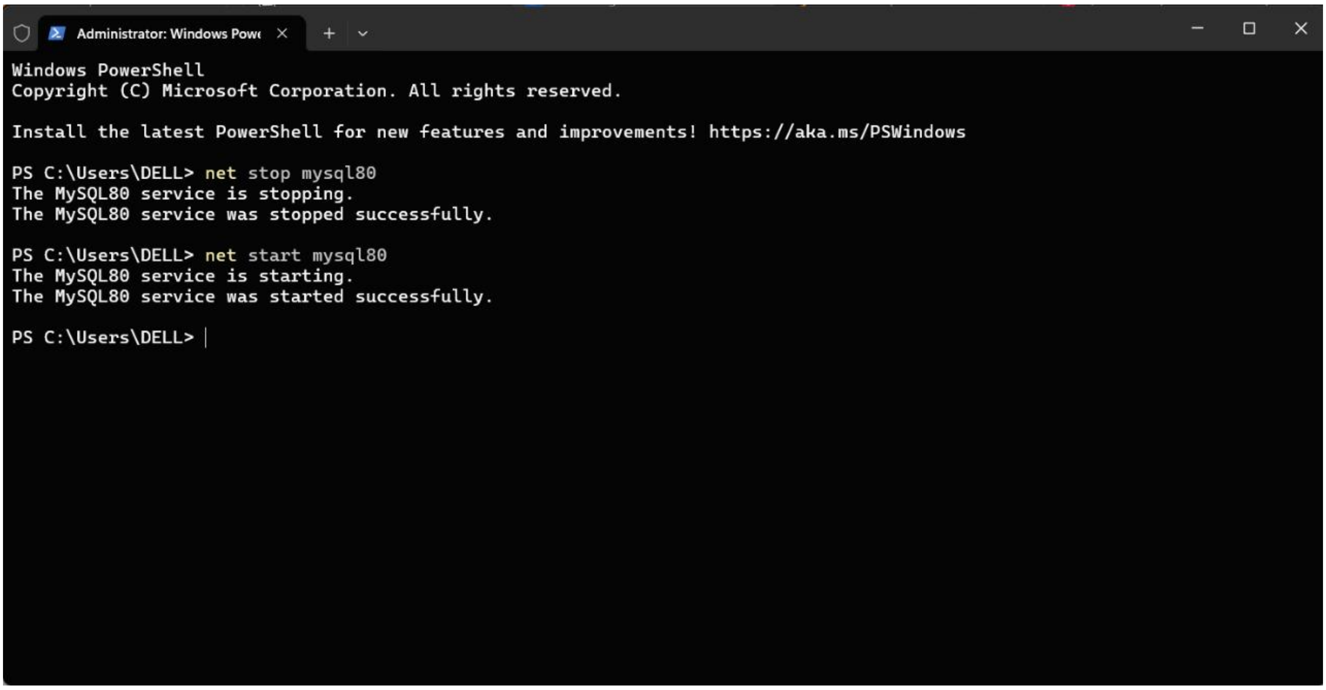
Step 2: Connect Slave 1 to Master

1. Open MySQL CLI:
   ```
   mysql -u root -p
   ```
2. Configure the slave with the master details:
   ```
   CHANGE MASTER TO
   MASTER_HOST='192.168.1.10',
   MASTER_USER='replicator',
   MASTER_PASSWORD='password',
   MASTER_LOG_FILE='mysql-bin.000001',
   MASTER_LOG_POS=154;
   ```
3. Start the Slave:
   ```
   START SLAVE;
   ```

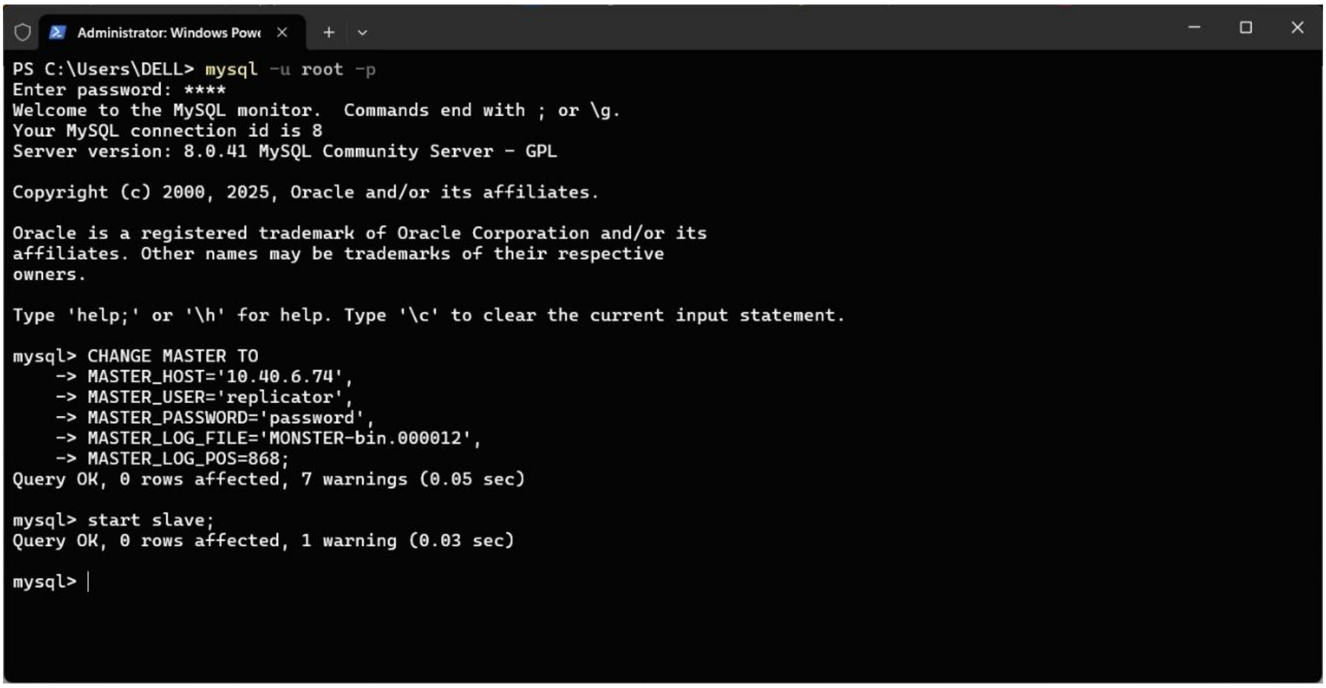
Repeat the similar process for Slave 2 with its IP address.


my.ini
C: > ProgramData > MySQL > MySQL Server 8.0 > my.ini
Run Testcases ⊗ 0 ⚠ 0
Ln 82, Col 16 Spaces: 4 UTF-8 CRLF {} Ini Go Live

```
[mysql]
no-beep

# default-character-set=

# SERVER SECTION
# ----------------------------------------------------------------------
#
# The following options will be read by the MySQL Server. Make sure that
# you have installed the server correctly (see above) so it reads this
# file.
#
[mysqld]
server-id = 2
relay-log = relay-log

# The next three options are mutually exclusive to SERVER_PORT below.
# skip-networking
# enable-named-pipe
# shared-memory

# shared-memory-base-name=MYSQL

# The Pipe the MySQL Server will use.
# socket=MYSQL

# The access control granted to clients on the named pipe created by the MySQL Server.
# named-pipe-full-access-group=

# The TCP/IP Port the MySQL Server will listen on
port=3306

```



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\DELL> net stop mysql80
The MySQL80 service is stopping.
The MySQL80 service was stopped successfully.

PS C:\Users\DELL> net start mysql80
The MySQL80 service is starting.
The MySQL80 service was started successfully.

PS C:\Users\DELL> |
```



```
PS C:\Users\DELL> mysql -u root -p
Enter password: ****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.41 MySQL Community Server - GPL

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> CHANGE MASTER TO
    -> MASTER_HOST='10.40.6.74',
    -> MASTER_USER='replicator',
    -> MASTER_PASSWORD='password',
    -> MASTER_LOG_FILE='MONSTER-bin.000012',
    -> MASTER_LOG_POS=868;
Query OK, 0 rows affected, 7 warnings (0.05 sec)

mysql> start slave;
Query OK, 0 rows affected, 1 warning (0.03 sec)

mysql> |
```

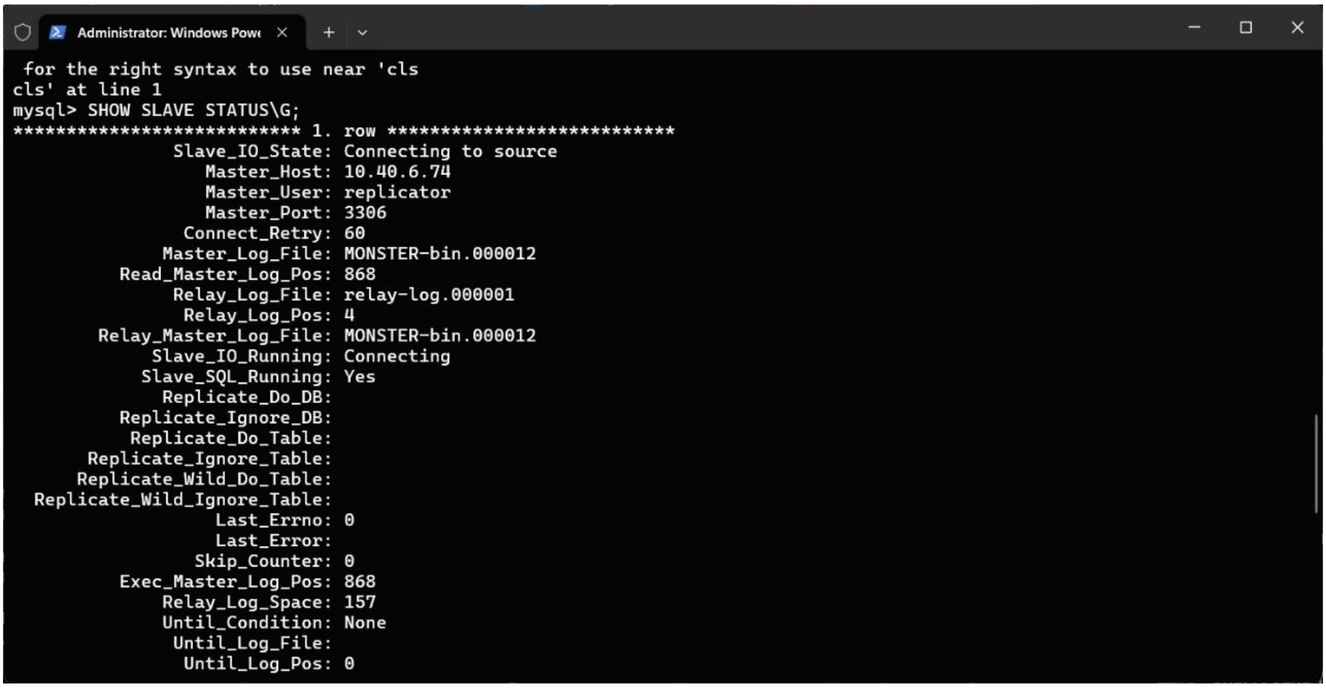
Verify Replication Status

On both Slave 1 and Slave 2, check the replication status:

SHOW SLAVE STATUS\G;

Check the following:

- Slave_IO_Running: Yes
- Slave_SQL_Running: Yes
- No replication errors.



```
 for the right syntax to use near 'cls
cls' at line 1
mysql> SHOW SLAVE STATUS\G;
*************************** 1. row ***************************
               Slave_IO_State: Connecting to source
                  Master_Host: 10.40.6.74
                  Master_User: replicator
                  Master_Port: 3306
                Connect_Retry: 60
              Master_Log_File: MONSTER-bin.000012
          Read_Master_Log_Pos: 868
               Relay_Log_File: relay-log.000001
                Relay_Log_Pos: 4
        Relay_Master_Log_File: MONSTER-bin.000012
             Slave_IO_Running: Connecting
            Slave_SQL_Running: Yes
              Replicate_Do_DB:
          Replicate_Ignore_DB:
           Replicate_Do_Table:
       Replicate_Ignore_Table:
      Replicate_Wild_Do_Table:
  Replicate_Wild_Ignore_Table:
                   Last_Errno: 0
                   Last_Error:
                 Skip_Counter: 0
          Exec_Master_Log_Pos: 868
              Relay_Log_Space: 157
              Until_Condition: None
               Until_Log_File:
                Until_Log_Pos: 0
```

Testing Distributed Database

1. **On Master:**

   Create a database and table:

   ```
   CREATE DATABASE distributed_db;
   USE distributed_db;

   CREATE TABLE employees (
     id INT PRIMARY KEY,
     name VARCHAR(50),
     salary DECIMAL(10, 2)
   );

   INSERT INTO employees VALUES (1, 'Alice', 60000);
   ```

2. **On Slave 1 and Slave 2:**

   Verify that the data is replicated:

   ```
   USE distributed_db;
   SELECT * FROM employees;
   ```

Conclusion

We have successfully set up a MySQL distributed database on three Windows machines using Master-Slave replication. This setup ensures data redundancy, failover protection, and better scalability.