Giza Systems

Final Project

YOUSSEF OSSAMA SAYED

y.ossama45@gmail.com

Github: https://github.com/YoussefOssamaa

Phone: 01000061252

Project Goal

Build a secure, automated, and remote-accessible MySQL setup across two VMs. This reflects real-world infrastructure tasks in modern system administration, DevOps, and backend engineering roles.

What will be built:

- **A Red Hat 9 VM as the database server.**
- **An Ubuntu 22.04 VM as the remote client.**
- **A fully functioning MySQL 8+ server, configured with:**
- **A dedicated project database (ProjectDB)**
- **Two user roles: readonly_user and readwrite_user**
- **Opened port 3306 with safe firewall and bind-address settings.**

Automation with Scripts

- **setup_mysql_server.sh: Sets up the server, creates users with specific privileges, configures the firewall, and allows remote connections.**
- **setup_mysql_client.sh: Automates connection tests and executes SQL queries to confirm user permissions.**

setup_mysql_server.sh:

Declaring Variables:

```
#!/bin/bash

set -e ##to exit on error


MYSQL_ROOT_PASSWORD="Root@1234"
DB_NAME="ProjectDB"
READ_ONLY_USER="readonly_user"
READ_ONLY_PASSWORD="Readonly@1234"
READ_WRITE_USER="readwrite_user"
READ_WRITE_PASSWORD="Readwrite@1234"
```

Resetting the Network Adapters and Create Fresh Configurations

```
echo " Resetting and Configuring Network Adapters"

OLD_CONS=$(nmcli -t -f NAME connection show)
for con in $OLD_CONS; do
    echo "Deleting old connection: $con"
    sudo nmcli connection delete "$con" || true
done


sleep 2

IF_NAT=$(nmcli device | awk '/ethernet/ {print $1}' | head -n1)
IF_HOST=$(nmcli device | awk '/ethernet/ {print $1}' | tail -n1)
```

Setting the Gateway and the routing tables

```
echo "Using $IF_NAT for NAT and $IF_HOST for Host-Only"

sudo nmcli connection add type ethernet ifname "$IF_NAT" con-name nat0
sudo nmcli connection add type ethernet ifname "$IF_HOST" con-name hostonly0

sudo nmcli connection up nat0
sudo nmcli connection up hostonly0

GATEWAY=$(nmcli -g IP4.GATEWAY device show "$IF_NAT" | head -n1)
if [ -n "$GATEWAY" ]; then
    sudo ip route del default || true
    sudo ip route add default via "$GATEWAY" dev "$IF_NAT"
    echo " Default route set via $GATEWAY on $IF_NAT"
else
    echo " Could not detect gateway on $IF_NAT"
fi


echo " Testing Internet..."
ping -c 3 8.8.8.8 || echo "Ping to 8.8.8.8 failed"
```

Installing MYSQL Server and starting the service

```
echo "INSTALLING MYSQL SERVER:"

sudo dnf install -y https://dev.mysql.com/get/mysql80-community-release-el9-1.noarch.rpm

echo "Importing the latest MySQL GPG key:"
sudo rpm --import https://repo.mysql.com/RPM-GPG-KEY-mysql-2023


sudo dnf clean all

if dnf module list mysql | grep -q mysql; then
sudo dnf module reset mysql -y
sudo dnf module disable mysql -y
fi
sudo dnf install -y mysql-community-server

echo "Starting MySQL service:"
sudo systemctl enable mysqld
sudo systemctl start mysqld
```

Check if the service is running and the installation is completed

```
if ! sudo systemctl is-active --quiet mysqld; then
    echo "Error: MySQL service is not running."
    exit 1
fi

echo "MYSQL SERVER INSTALLATION COMPLETED"
```

Get the temporary default Root Password created from MYSQL LOGS

```
TEMP_PASS=$(sudo grep 'temporary password' /var/log/mysqld.log | tail -n 1 | awk '{print $NF}')
if [ -z "$TEMP_PASS" ]; then
    echo "Error: Could not retrieve temporary password from /var/log/mysqld.log"
    exit 1
fi


echo "Checking if MySQL root password is already set..."
if mysqladmin -u root -p"$TEMP_PASS" status 2>/dev/null; then
    echo "Temporary password is valid, proceeding with secure installation."
else
    echo "Temporary password is invalid. Checking if root password is already set to $MYSQL_ROOT_PASSWORD..."
    if mysqladmin -u root -p"$MYSQL_ROOT_PASSWORD" status 2>/dev/null; then
        echo "Root password is already set to $MYSQL_ROOT_PASSWORD. Skipping secure installation."
        TEMP_PASS="$MYSQL_ROOT_PASSWORD"
    else
        echo "Error: Unable to log in with temporary password or root password."
        exit 1
    fi
fi
```

Securing MySQL and setting the Password

```
mysql --connect-expired-password -u root -p"$TEMP_PASS" <<EOF

ALTER USER 'root'@'localhost' IDENTIFIED BY '$MYSQL_ROOT_PASSWORD';
DELETE FROM mysql.user WHERE User='';
DROP DATABASE IF EXISTS test;
FLUSH PRIVILEGES;
EOF
```

Creating the new Database with the 2 users (Read_only user and Read_write user)

```
echo "Creating a new database: $DB_NAME"
sudo mysql -uroot -p"$MYSQL_ROOT_PASSWORD" <<EOF
CREATE DATABASE IF NOT EXISTS $DB_NAME;
CREATE USER IF NOT EXISTS '$READ_ONLY_USER'@'%' IDENTIFIED BY '$READ_ONLY_PASSWORD';
GRANT SELECT ON $DB_NAME.* TO '$READ_ONLY_USER'@'%';

CREATE USER IF NOT EXISTS '$READ_WRITE_USER'@'%' IDENTIFIED BY '$READ_WRITE_PASSWORD';
GRANT CREATE, SELECT, INSERT, UPDATE, DELETE ON $DB_NAME.* TO '$READ_WRITE_USER'@'%' ;

FLUSH PRIVILEGES;
EOF
```

At the end, enabling clients to connect to the server through the port 3306

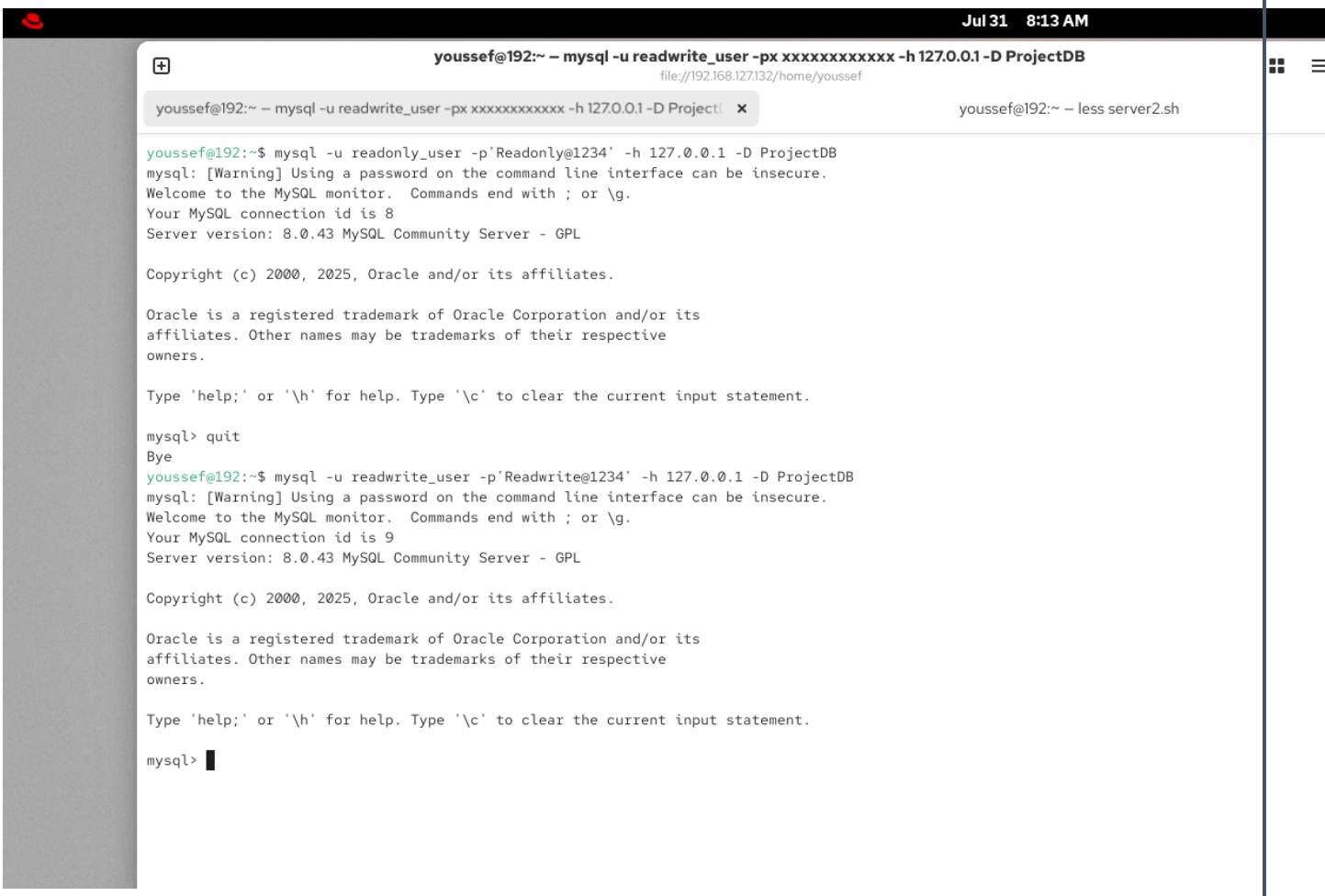
```
sudo sed -i "s/^bind-address.*/bind-address = 0.0.0.0/" /etc/my.cnf.d/mysqld.cnf || echo "bind-address=0.0.0.0" | sudo tee -a /etc/my.cnf.d/mysqld.cnf

echo "Restarting MySQL service to apply changes:"
sudo systemctl restart mysqld

echo "adding port 3306 to firewall"
sudo firewall-cmd --permanent --add-port=3306/tcp
sudo firewall-cmd --reload


echo "Installation and configuration of MySQL server completed successfully."
echo "Server script completed."
```

Executing the Server Script

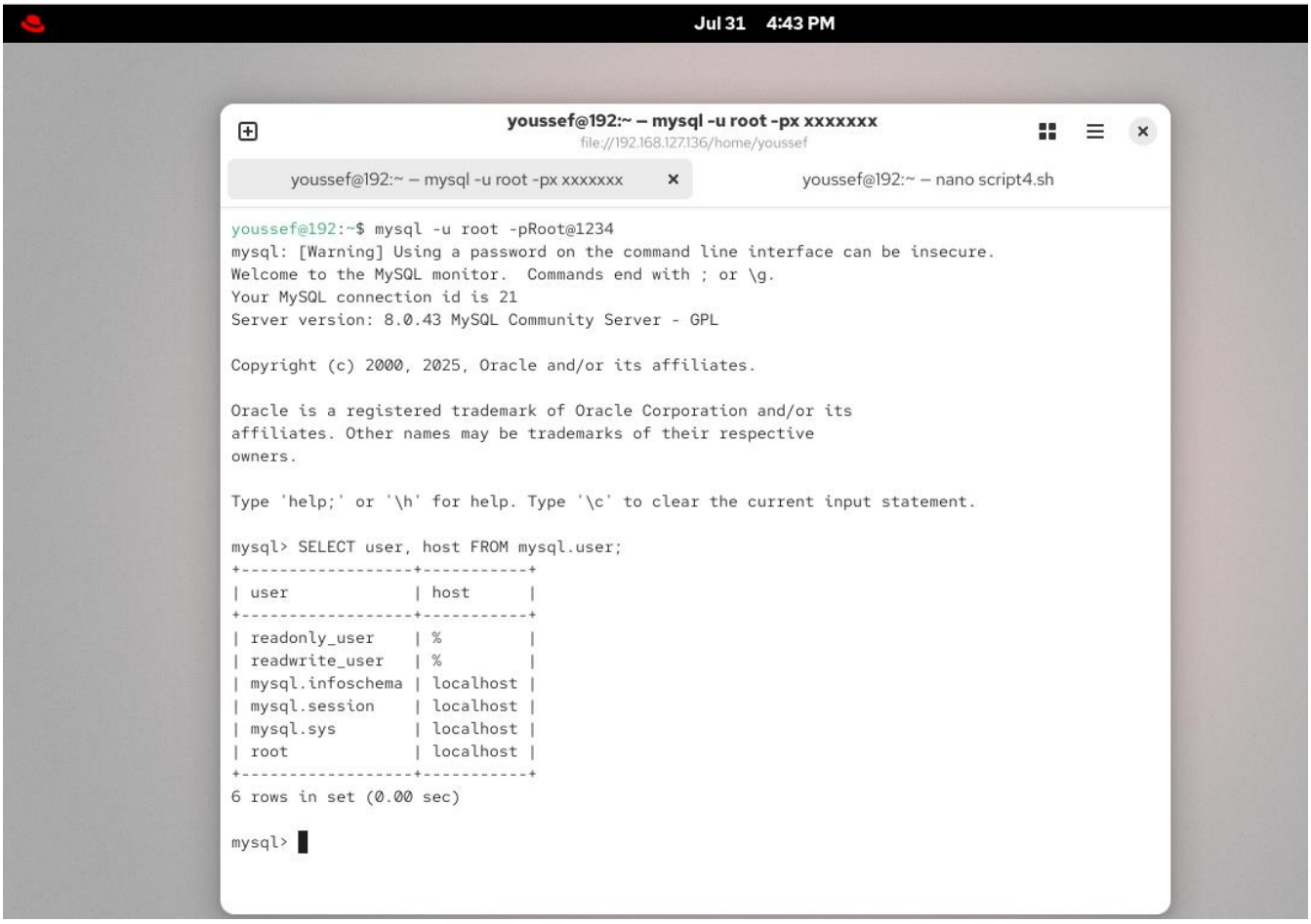
Testing The mysqld status

```
youssef@192:~$ sudo systemctl status mysqld
● mysqld.service - MySQL Server
     Loaded: loaded (/usr/lib/systemd/system/mysqld.service; enabled; preset: disabled)
     Active: active (running) since Thu 2025-07-31 08:08:34 EEST; 1min 57s ago
 Invocation: 8a02f962725d4967b26a0c2c85edb4f7
       Docs: man:mysqld(8)
             http://dev.mysql.com/doc/refman/en/using-systemd.html
    Process: 7465 ExecStartPre=/usr/bin/mysqld_pre_systemd (code=exited, status=0/SUCCESS)
   Main PID: 7494 (mysqld)
     Status: "Server is operational"
      Tasks: 37 (limit: 13425)
     Memory: 379.6M (peak: 397.7M)
        CPU: 2.518s
     CGroup: /system.slice/mysqld.service
             └─7494 /usr/sbin/mysqld

Jul 31 08:08:32 192.168.127.132 systemd[1]: Starting mysqld.service - MySQL Server...
Jul 31 08:08:34 192.168.127.132 (mysqld)[7494]: mysqld.service: Referenced but unset environment variable evaluates to an empty string: MYSQLD_>
Jul 31 08:08:34 192.168.127.132 systemd[1]: Started mysqld.service - MySQL Server.
lines 1-18/18 (END)
```

Accessing Both users from the server



Show all the users on the server

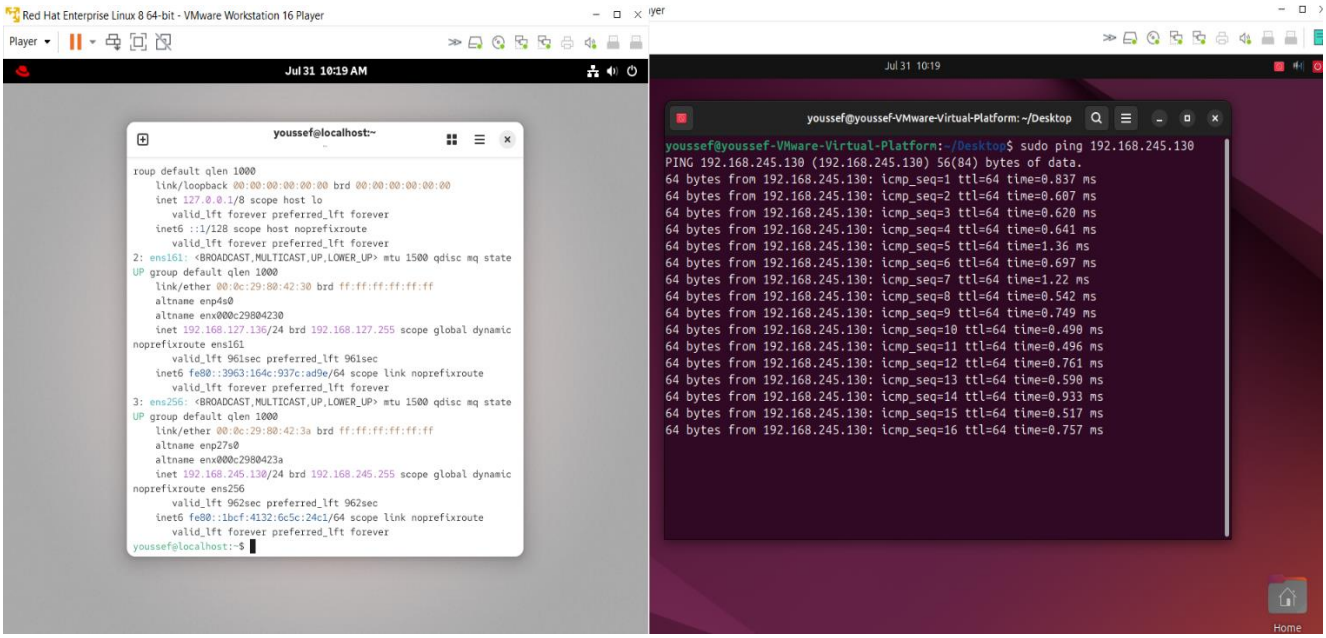


setup_mysql_client.sh

Declaring Variables and Installing MYSQL-Client

```
#!/bin/bash

set -e

MYSQL_SERVER_IP="192.168.245.130"
DB_NAME="ProjectDB"
READ_ONLY_USER="readonly_user"
READ_ONLY_PASSWORD="Readonly@1234"
READ_WRITE_USER="readwrite_user"
READ_WRITE_PASSWORD="Readwrite@1234"


echo "Installing MySQL client"
sudo apt update -y
sudo apt install mysql-client -y


echo "Client script completed."
```

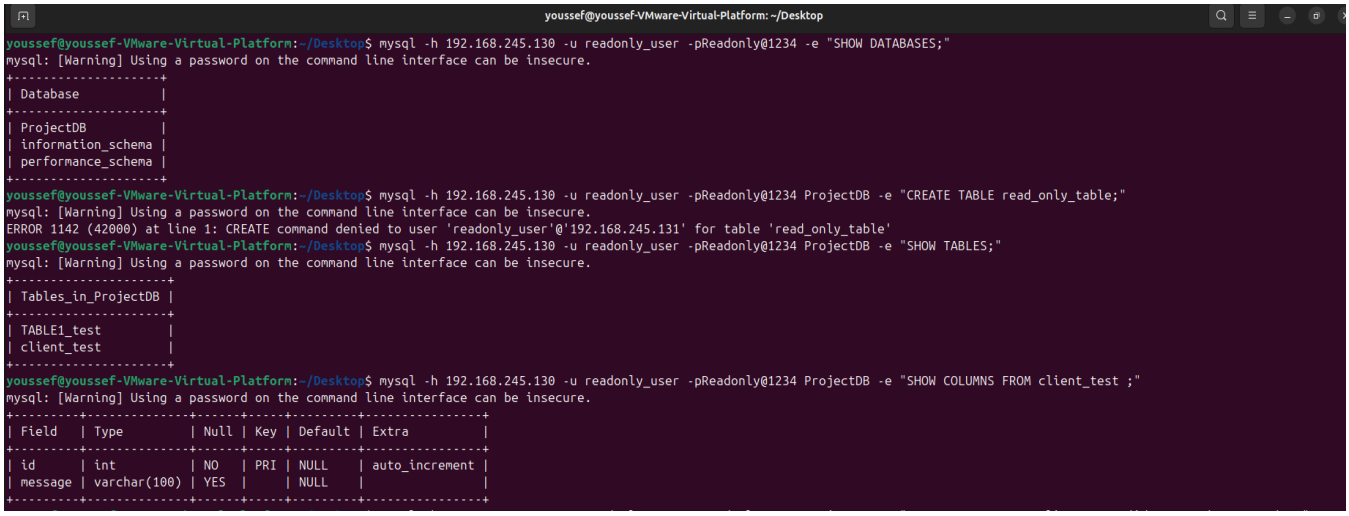
Pinging the Server IP

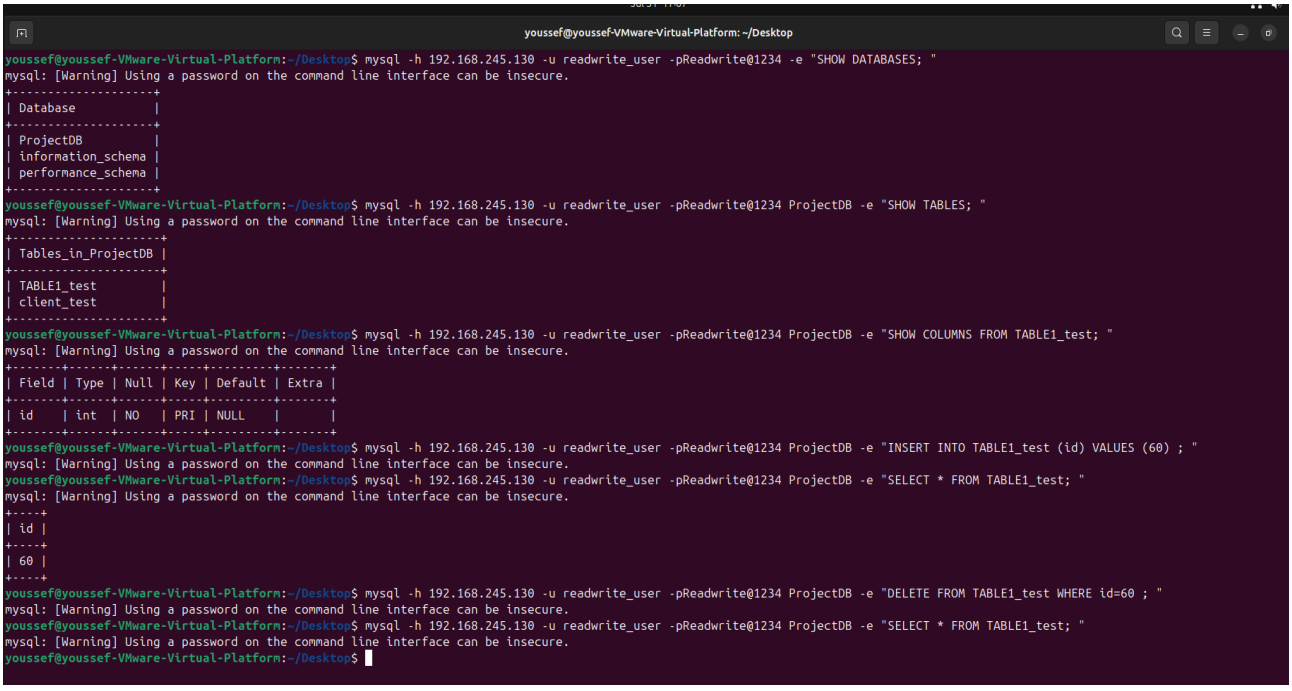


Executing the Client Script

```
youssef@youssef-VMware-Virtual-Platform:~/Desktop$ ./client1.sh
Installing MySQL client
Hit:1 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:2 http://eg.archive.ubuntu.com/ubuntu noble InRelease
Hit:3 http://eg.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:4 http://eg.archive.ubuntu.com/ubuntu noble-backports InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
156 packages can be upgraded. Run 'apt list --upgradable' to see them.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
mysql-client is already the newest version (8.0.42-0ubuntu0.24.04.2).
0 upgraded, 0 newly installed, 0 to remove and 156 not upgraded.
Testing connection to MySQL server at 192.168.245.130
mysql: [Warning] Using a password on the command line interface can be insecure.
+--------------------+
| Database           |
+--------------------+
| ProjectDB          |
| information_schema |
| performance_schema |
+--------------------+
Successfully connected to MySQL server as readonly_user
Running a sample SELECT query on ProjectDB with the READ_ONLY_USER
mysql: [Warning] Using a password on the command line interface can be insecure.
+--------------------+
| Tables_in_ProjectDB |
+--------------------+
| TABLE1_test         |
+--------------------+
Running a sample INSERT query on ProjectDB with the READ_ONLY_USER
mysql: [Warning] Using a password on the command line interface can be insecure.
ERROR 1142 (42000) at line 1: CREATE command denied to user 'readonly_user'@'192.168.245.131' for table 'client_test'
CORRECT, Failed to create table with read_only_user
Testing CREATE operation with read-write user
mysql: [Warning] Using a password on the command line interface can be insecure.
Successfully connected and executed write query as readwrite_user
Testing INSERT operation with read-write user
mysql: [Warning] Using a password on the command line interface can be insecure.
INSERT successful as readwrite_user
Client script completed.
youssef@youssef-VMware-Virtual-Platform:~/Desktop$
```

MANUAL TEST AFTER EXECUTING THE SCRIPT





Questions for Understanding

1. What is the purpose of having read-only and read-write users?

To implement principle of least privilege.

- Read-only users can query data but cannot modify it. Useful for reporting, analytics, or monitoring.
- Read-write users can perform inserts, updates, and deletes. Used by applications that need to modify the database.

Benefits:

- Prevents accidental or malicious changes.
- Limits damage if credentials are compromised.
- Easier to audit and control access.

2. Why is it important to automate server setup using Shell scripts?

- Ensures consistency across multiple environments (dev/test/prod).
- Saves time and effort during setup or reinstallation.
- Reduces human error.
- Enables repeatable deployments (in CI/CD pipelines or cloud setups).

3. What changes are required in the MySQL configuration to allow remote access?

- Edit my.cnf or mysqld.cnf and change:

  bind-address = 0.0.0.0 (to allow connections from any host)

- Create users with % or specific IP in the host field:

  CREATE USER 'user'@'%' IDENTIFIED BY 'password';

- Grant appropriate privileges:

  GRANT SELECT ON db.* TO 'readonly_user'@'%';

- Restart MySQL after config changes:

  sudo systemctl restart mysqld

4. What security risks exist when opening MySQL to external connections, and how can we mitigate them?

Risks:

- Unauthorized access if credentials are leaked.
- Brute-force attacks.
- Data interception if traffic is unencrypted.
- SQL injection if used with vulnerable apps.

Mitigations:

- Use strong passwords and rotate them regularly.
- Restrict access with firewall rules or only allow known IPs.
- Use SSL/TLS encryption for client-server connections.
- Avoid using % in host field if not necessary.
- Do not expose MySQL port (3306) publicly unless required.
- Regularly audit logs and permissions.

5. How would you secure the MySQL root account in a production environment?

- Set a strong, unique password.
- Use mysql_secure_installation to:
  1. Remove anonymous users
  2. Disallow remote root login
  3. Remove test database

- Restrict root to localhost only:
  ALTER USER 'root'@'%' IDENTIFIED WITH mysql_native_password BY 'newpass';
  REVOKE ALL PRIVILEGES ON *.* FROM 'root'@'%';
  DROP USER 'root'@'%';

- Use sudo and local socket (mysql -u root -p) instead of remote root login.
- Enable two-factor authentication or use IAM-based authentication in cloud platforms.