Azure MySQL Flexible Server Setup & Connection Guide (v8.4)

1. Prerequisites

Azure CLI installed

Python 3.x installed

pip installed

Internet access

2. Install Azure CLI & Login

# Install Azure CLI (if not already installed)

curl -sL https://aka.ms/InstallAzureCLIDeb | sudo bash

# Login to Azure

az login

3. Create Resource Group

az group create --name mysql-rg --location eastus2

4. Create MySQL Flexible Server (v8.4)

az mysql flexible-server create \

--resource-group mysql-rg \

--name mysql84-demo-ruthwik \

--location eastus2 \

--admin-user adminuser \

--admin-password "SecureP@ss123!" \

--version 8.4 \

--tier Burstable \

--sku-name Standard\_B1s \

--storage-size 20 \

--public-access "0.0.0.0-255.255.255.255"

5. Configure Azure CLI Extensions

# Allow extension installation without prompt

az config set extension.use\_dynamic\_install=yes\_without\_prompt

# Install required extension

az extension add --name rdbms-connect

6. Add Firewall Rule for Your IP

MY\_IP=$(curl -s ifconfig.me)

az mysql flexible-server firewall-rule create \

--name mysql84-demo-ruthwik \

--resource-group mysql-rg \

--rule-name "allow\_my\_ip" \

--start-ip-address $MY\_IP \

--end-ip-address $MY\_IP

# Wait for firewall rules to propagate

sleep 120

7. Connect via MySQL CLI

az mysql flexible-server connect \

--name mysql84-demo-ruthwik \

--resource-group mysql-rg \

--admin-user adminuser \

--admin-password "SecureP@ss123!" \

--interactive

8. Create Test Database (in MySQL Shell)

sql

Copy

Edit

CREATE DATABASE testdb;

exit;

9. Test Python Connection

a. Install Required Python Package

pip install mysql-connector-python

b. Download SSL Certificate

curl -o DigiCertGlobalRootCA.crt.pem https://cacerts.digicert.com/DigiCertGlobalRootCA.crt.pem

c. Create test\_connection.py

python

Copy

Edit

import mysql.connector

config = {

"host": "mysql84-demo-ruthwik.mysql.database.azure.com",

"user": "adminuser",

"password": "SecureP@ss123!",

"database": "testdb",

"ssl\_ca": "DigiCertGlobalRootCA.crt.pem"

}

try:

cnx = mysql.connector.connect(\*\*config)

print("✅ Connection successful!")

cnx.close()

except Exception as e:

print(f"❌ Connection failed: {e}")

d. Run the Connection Test

python test\_connection.py

10. (Optional) Disable SSL for Testing Only

⚠️ Not recommended for production use.

az mysql flexible-server update \

--name mysql84-demo-ruthwik \

--resource-group mysql-rg \

--ssl-enforcement Disabled

11. Troubleshooting Checklist

✅ Verify resource group:

az group show --name mysql-rg

✅ Check server status:

az mysql flexible-server show \

--name mysql84-demo-ruthwik \

--resource-group mysql-rg

✅ List firewall rules:

az mysql flexible-server firewall-rule list \

--name mysql84-demo-ruthwik \

--resource-group mysql-rg

✅ Test basic connectivity (port 3306):

nc -zv mysql84-demo-ruthwik.mysql.database.azure.com 3306