Cycle 08 - Homework Summary

1. PostgreSQL & Microsoft SQL Server Deployment on EC2

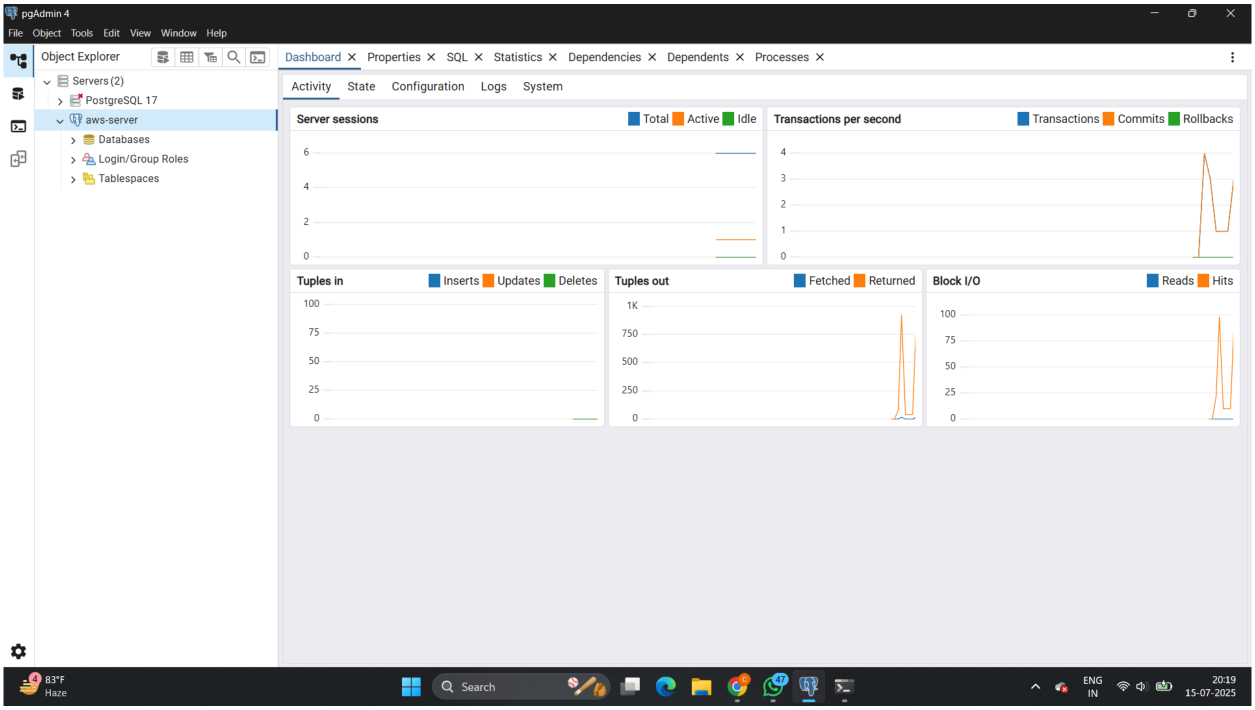
Objective:

Install and configure both PostgreSQL and Microsoft SQL Server on AWS EC2 instances for remote access.

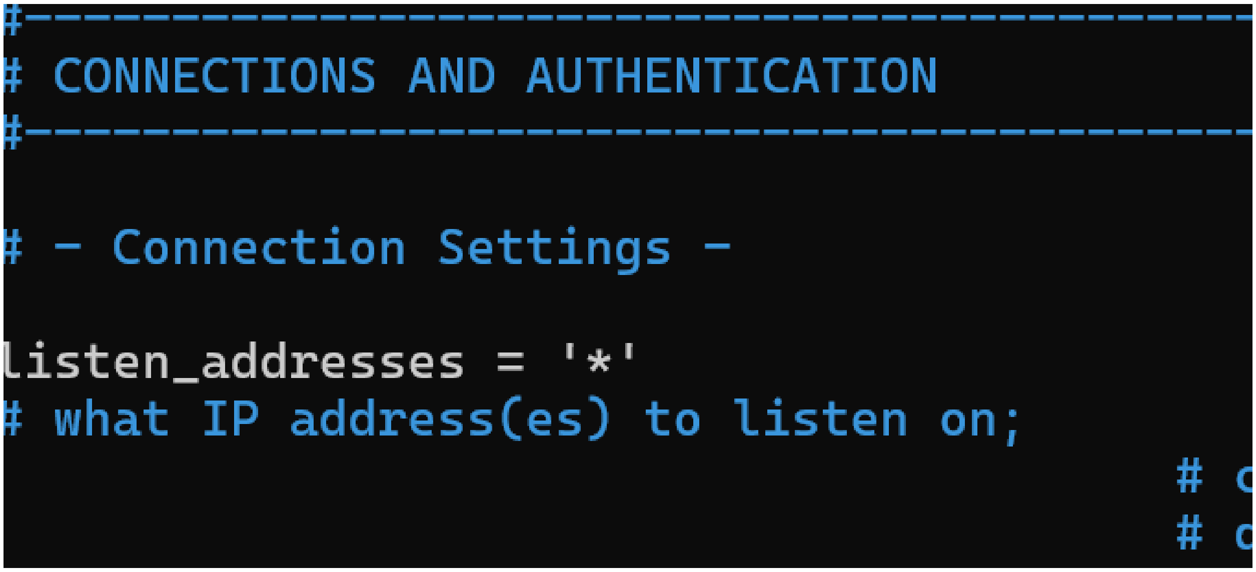
Key Findings:

- **PostgreSQL**
  - Installed using `sudo apt install postgresql`
  - Default port: `5432`
  - Remote access enabled by modifying `postgresql.conf` to bind on `0.0.0.0` and updating `pg_hba.conf`
  - pgAdmin used as client tool and tested for remote connectivity

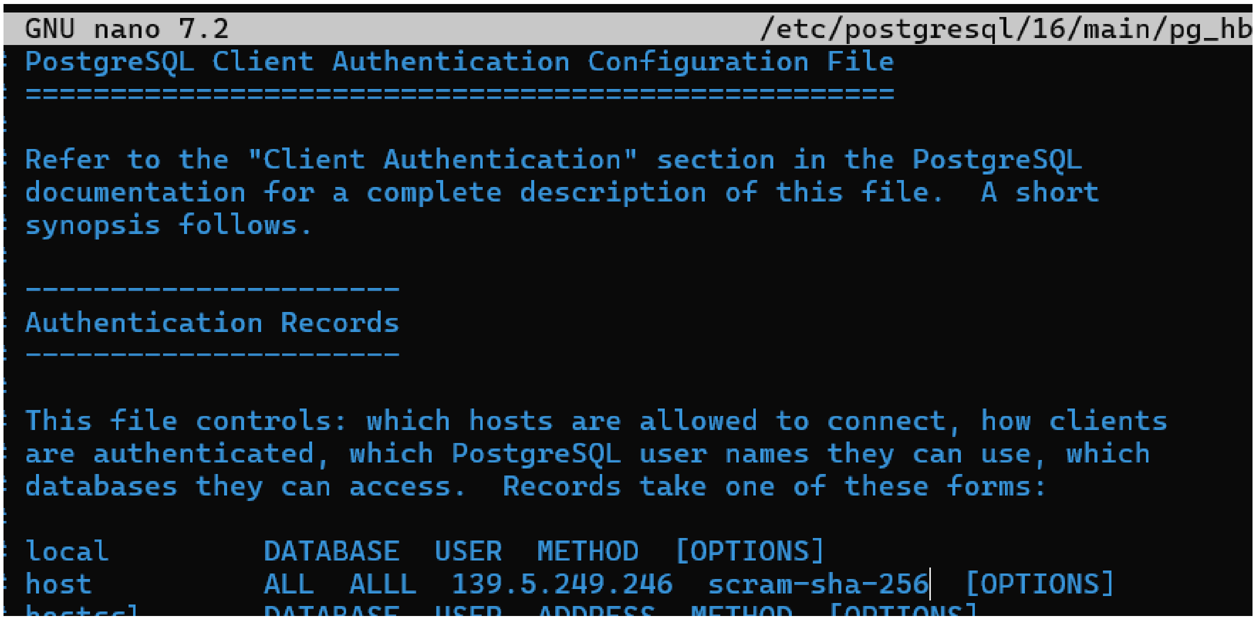
postgresql pgadmin console:-



CHANGES IN NANO:

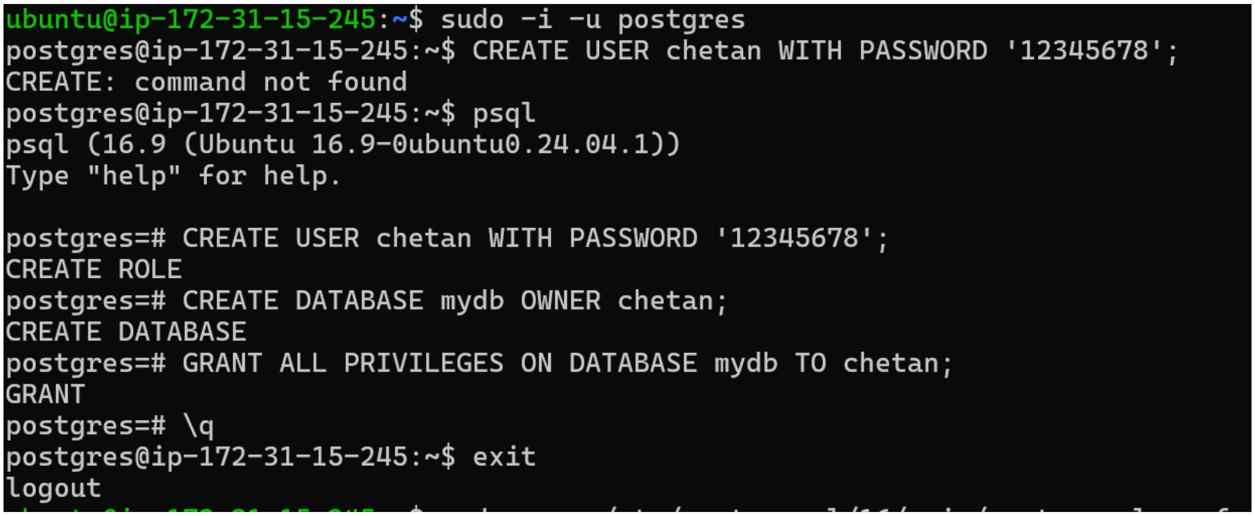


```
#------------------------------------------
# CONNECTIONS AND AUTHENTICATION
#------------------------------------------


# - Connection Settings -

listen_addresses = '*'
# what IP address(es) to listen on;
                                       # c
                                       # c
```



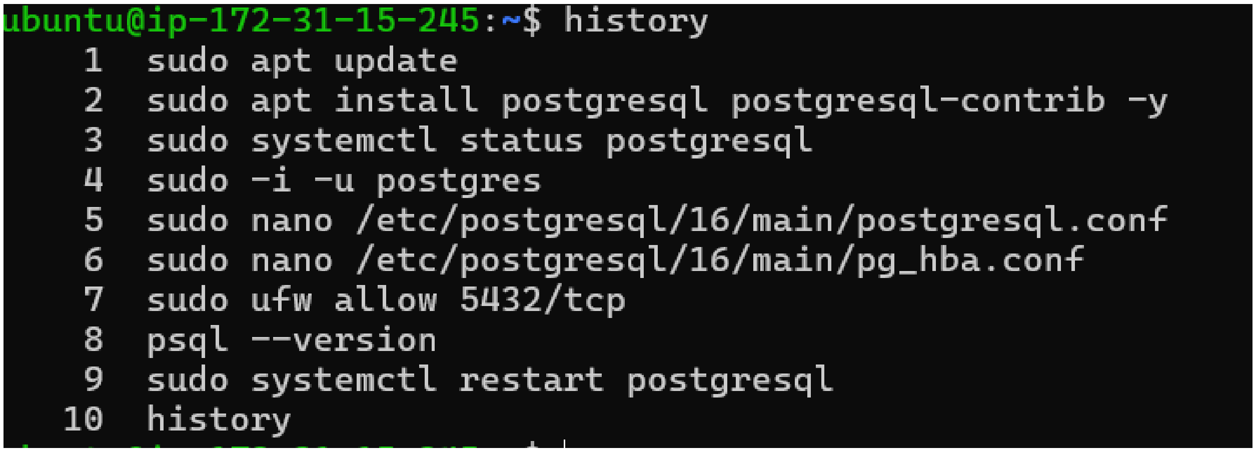
```
 GNU nano 7.2                         /etc/postgresql/16/main/pg_hb
# PostgreSQL Client Authentication Configuration File
# ===================================================
#
# Refer to the "Client Authentication" section in the PostgreSQL
# documentation for a complete description of this file.  A short
# synopsis follows.
#
# ----------------------
# Authentication Records
# ----------------------
#
# This file controls: which hosts are allowed to connect, how clients
# are authenticated, which PostgreSQL user names they can use, which
# databases they can access.  Records take one of these forms:
#
# local         DATABASE  USER  METHOD  [OPTIONS]
# host          ALL  ALLL  139.5.249.246  scram-sha-256  [OPTIONS]
# hostssl       DATABASE  USER  ADDRESS  METHOD  [OPTIONS]
```

PROSTGRES COMMANDS:

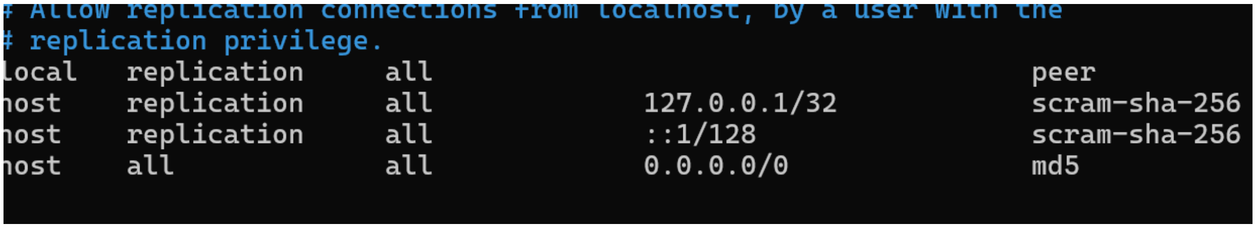


```
ubuntu@ip-172-31-15-245:~$ sudo -i -u postgres
postgres@ip-172-31-15-245:~$ CREATE USER chetan WITH PASSWORD '12345678';
CREATE: command not found
postgres@ip-172-31-15-245:~$ psql
psql (16.9 (Ubuntu 16.9-0ubuntu0.24.04.1))
Type "help" for help.

postgres=# CREATE USER chetan WITH PASSWORD '12345678';
CREATE ROLE
postgres=# CREATE DATABASE mydb OWNER chetan;
CREATE DATABASE
postgres=# GRANT ALL PRIVILEGES ON DATABASE mydb TO chetan;
GRANT
postgres=# \q
postgres@ip-172-31-15-245:~$ exit
logout
```

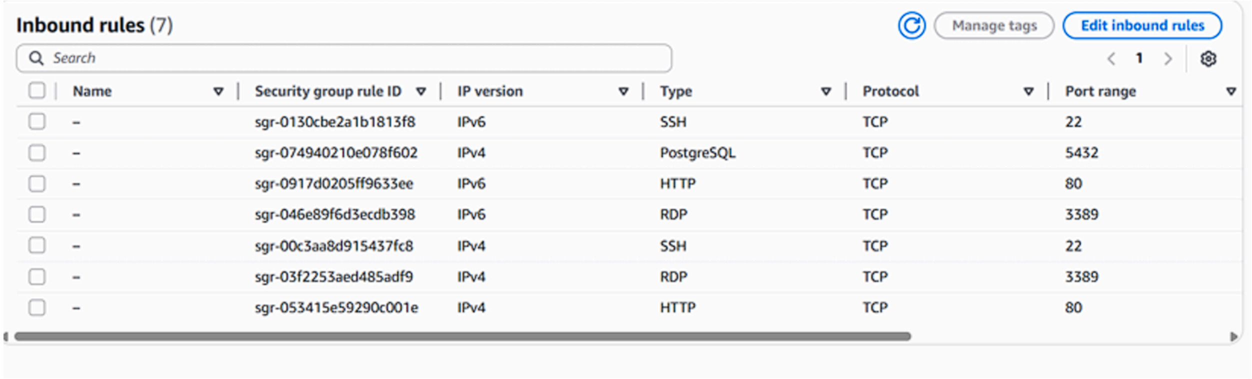
UNBUNTO COMMANDS:



```
ubuntu@ip-172-31-15-245:~$ history
    1  sudo apt update
    2  sudo apt install postgresql postgresql-contrib -y
    3  sudo systemctl status postgresql
    4  sudo -i -u postgres
    5  sudo nano /etc/postgresql/16/main/postgresql.conf
    6  sudo nano /etc/postgresql/16/main/pg_hba.conf
    7  sudo ufw allow 5432/tcp
    8  psql --version
    9  sudo systemctl restart postgresql
   10  history
```



```
# Allow replication connections from localhost, by a user with the
# replication privilege.
local   replication     all                                     peer
host    replication     all             127.0.0.1/32            scram-sha-256
host    replication     all             ::1/128                 scram-sha-256
host    all             all             0.0.0.0/0               md5
```



Inbound rules (7)

□	Name	Security group rule ID	IP version	Type	Protocol	Port range
□	–	sgr-0130cbe2a1b1813f8	IPv6	SSH	TCP	22
□	–	sgr-074940210e078f602	IPv4	PostgreSQL	TCP	5432
□	–	sgr-0917d0205ff9633ee	IPv6	HTTP	TCP	80
□	–	sgr-046e89f6d3ecdb398	IPv6	RDP	TCP	3389
□	–	sgr-00c3aa8d915437fc8	IPv4	SSH	TCP	22
□	–	sgr-03f2253aed485adf9	IPv4	RDP	TCP	3389
□	–	sgr-053415e59290c001e	IPv4	HTTP	TCP	80

- Microsoft sql is not done due to requirements mismatch .

2. MySQL Deployment & Remote Access

Objective:

Deploy MySQL on Ubuntu EC2 and configure secure remote access.

Steps Followed:

1. Installed MySQL Server

   `sudo apt install mysql-server`

2. Ran secure setup

   `mysql_secure_installation` used to set password and remove insecure settings

3. Created a remote user

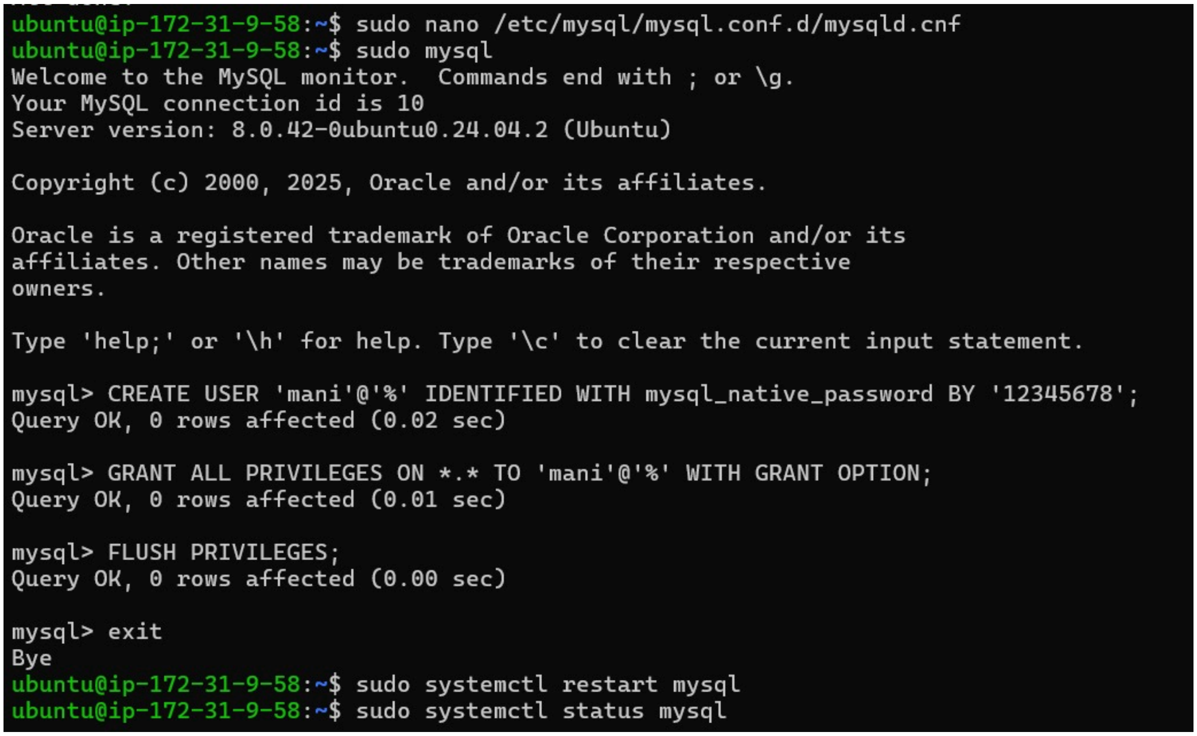
   ```
   sql
   CopyEdit
   CREATE USER 'adminuser'@'%' IDENTIFIED BY 'StrongPassword!';
   GRANT ALL PRIVILEGES ON *.* TO 'adminuser'@'%' WITH GRANT OPTIO
   N;
   FLUSH PRIVILEGES;
   ```

4. Modified bind-address to allow remote access

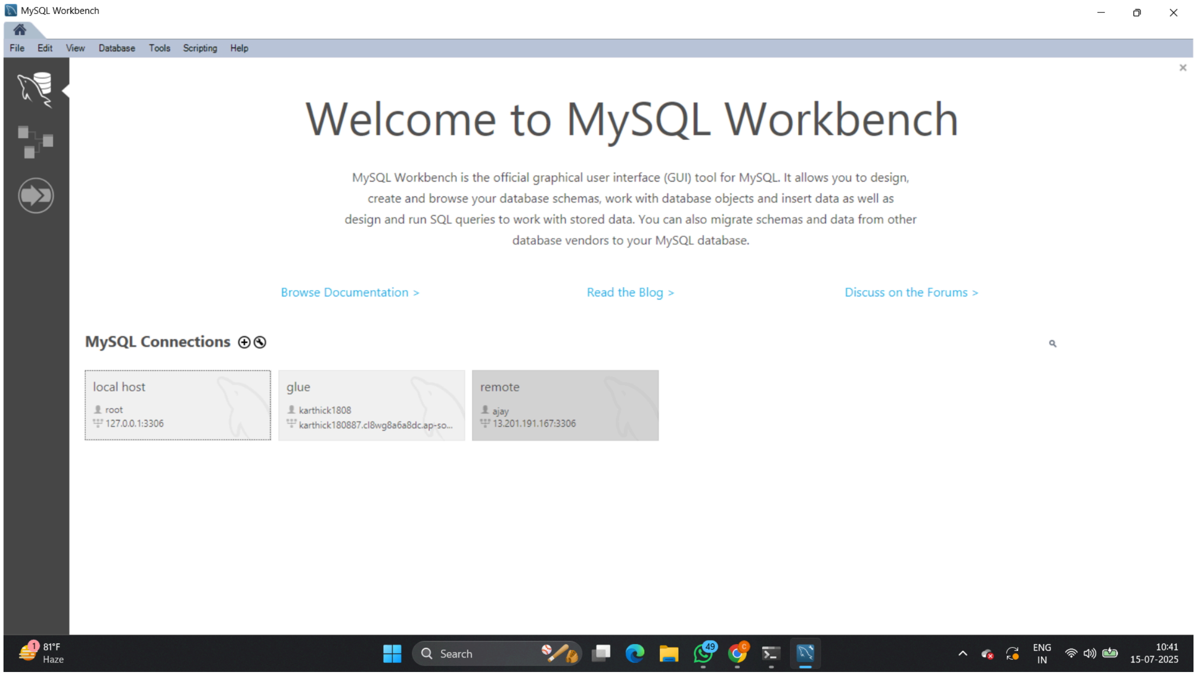
   `/etc/mysql/mysql.conf.d/mysqld.cnf` → `bind-address = 0.0.0.0`

5. Tested remote connection

Used MySQL Workbench to connect using EC2 Public IP and remote user



```
ubuntu@ip-172-31-9-58:~$ sudo nano /etc/mysql/mysql.conf.d/mysqld.cnf
ubuntu@ip-172-31-9-58:~$ sudo mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 10
Server version: 8.0.42-0ubuntu0.24.04.2 (Ubuntu)

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 'mani'@'%' IDENTIFIED WITH mysql_native_password BY '12345678';
Query OK, 0 rows affected (0.02 sec)

mysql> GRANT ALL PRIVILEGES ON *.* TO 'mani'@'%' WITH GRANT OPTION;
Query OK, 0 rows affected (0.01 sec)

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

mysql> exit
Bye
ubuntu@ip-172-31-9-58:~$ sudo systemctl restart mysql
ubuntu@ip-172-31-9-58:~$ sudo systemctl status mysql
```



Welcome to MySQL Workbench

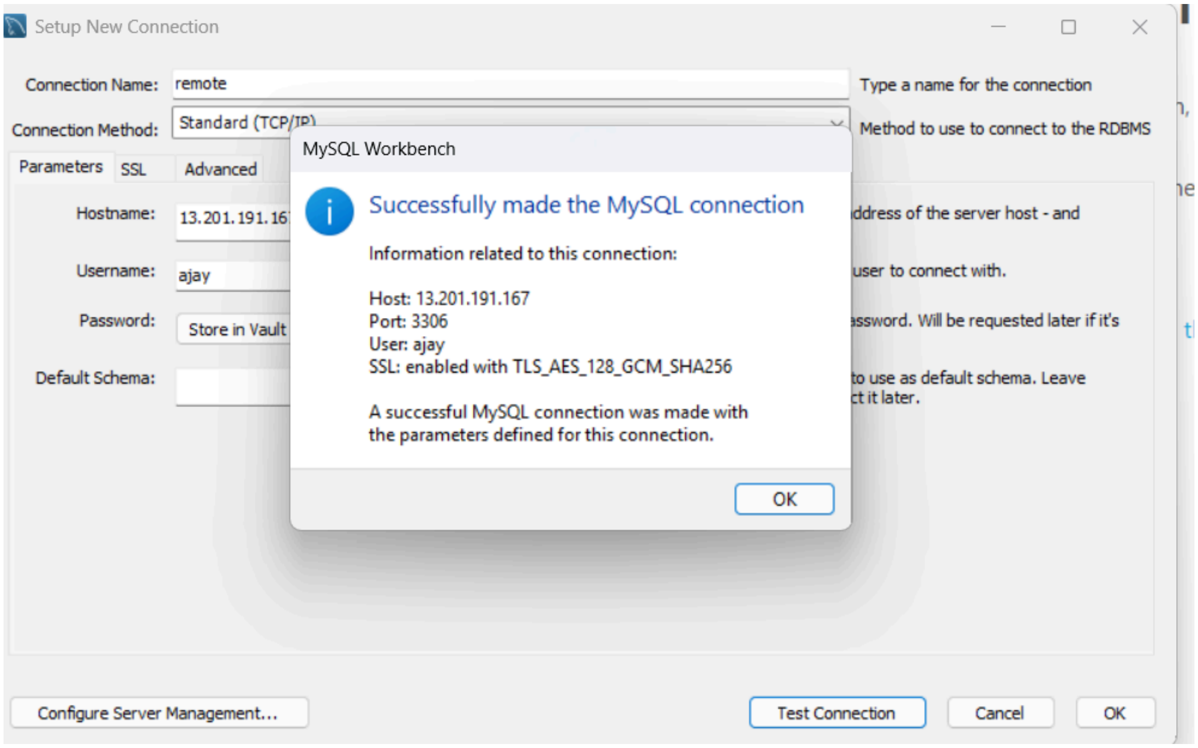
MySQL Workbench is the official graphical user interface (GUI) tool for MySQL. It allows you to design, create and browse your database schemas, work with database objects and insert data as well as design and run SQL queries to work with stored data. You can also migrate schemas and data from other database vendors to your MySQL database.

MySQL Connections

local host
root
127.0.0.1:3306

glue
karthick1808
karthick180887.cl8wg8a6a8dc.ap-so...

remote
ajay
13.201.191.167:3306



Setup New Connection

Connection Name: remote — Type a name for the connection

Connection Method: Standard (TCP/IP) — Method to use to connect to the RDBMS

Hostname: 13.201.191.16 — ddress of the server host - and

Username: ajay — user to connect with.

Password: Store in Vault — ssword. Will be requested later if it's

Default Schema: — to use as default schema. Leave ct it later.

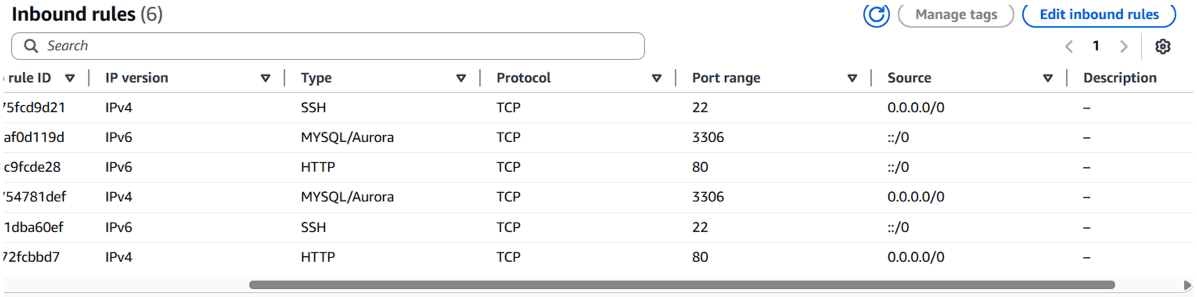
MySQL Workbench

Successfully made the MySQL connection

Information related to this connection:

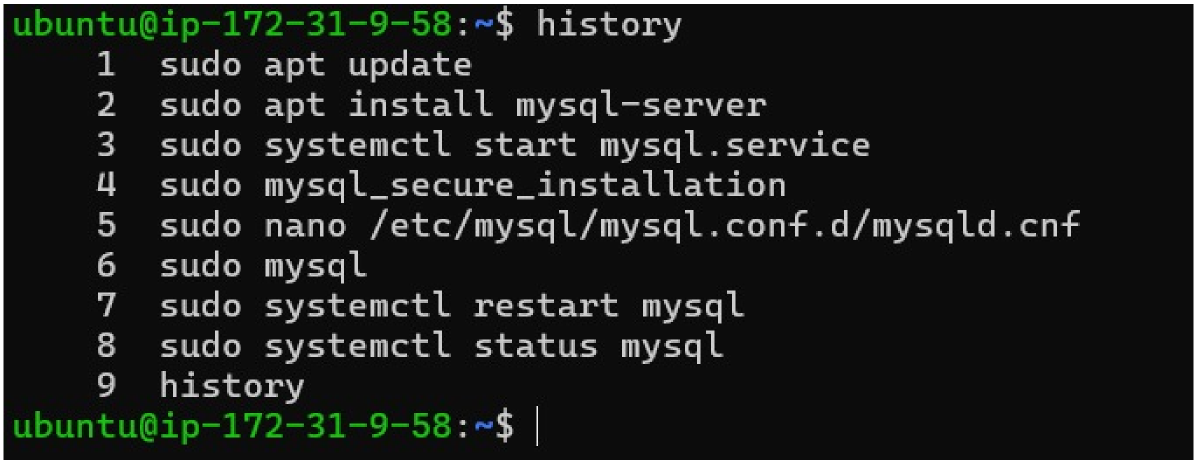
Host: 13.201.191.167
Port: 3306
User: ajay
SSL: enabled with TLS_AES_128_GCM_SHA256

A successful MySQL connection was made with the parameters defined for this connection.



Inbound rules (6)

rule ID	IP version	Type	Protocol	Port range	Source	Description
'5fcd9d21	IPv4	SSH	TCP	22	0.0.0.0/0	–
af0d119d	IPv6	MYSQL/Aurora	TCP	3306	::/0	–
c9fcde28	IPv6	HTTP	TCP	80	::/0	–
'54781def	IPv4	MYSQL/Aurora	TCP	3306	0.0.0.0/0	–
1dba60ef	IPv6	SSH	TCP	22	::/0	–
72fcbbd7	IPv4	HTTP	TCP	80	0.0.0.0/0	–



```
ubuntu@ip-172-31-9-58:~$ history
    1  sudo apt update
    2  sudo apt install mysql-server
    3  sudo systemctl start mysql.service
    4  sudo mysql_secure_installation
    5  sudo nano /etc/mysql/mysql.conf.d/mysqld.cnf
    6  sudo mysql
    7  sudo systemctl restart mysql
    8  sudo systemctl status mysql
    9  history
ubuntu@ip-172-31-9-58:~$ 
```

3. AWS Resource Cleanup

Actions Taken:

- All EC2 instances terminated after testing
- Unused security groups deleted, only default retained
- Verified no active instances in Mumbai (ap-south-1) region
- Confirmed only t2.micro or t3.micro instances were used
- Ensured Elastic IPs, volumes, and key pairs were cleaned up

4. DynamoDB (NoSQL) vs Relational Databases

Feature	DynamoDB (NoSQL)	MySQL/PostgreSQL (Relational)
Schema	Schema-less, flexible	Fixed schema with tables
Scalability	Auto-scalable, serverless	Vertical/horizontal scaling
Querying	Key-value access, limited	Full-featured SQL support
Use Cases	Real-time apps, gaming, IoT	Enterprise systems, analytics
Consistency	Eventually consistent (default)	Strong consistency by default

5. Jenkins Setup and Basics

- Installed Jenkins on Ubuntu EC2
- Default web access port: `8080`
- Initial password fetched from `/var/lib/jenkins/secrets/initialAdminPassword`
- Created and triggered a Freestyle project with simple shell command:

  ```
  bash
  CopyEdit
  echo "Hello from Jenkins!"
  ```

- Output verified in console logs

6. Advanced MySQL Security

Security Features Explored:

- Enabled `validate_password` plugin

  ```
  sql
  CopyEdit
  SHOW VARIABLES LIKE 'validate_password%';
  SET GLOBAL validate_password.policy = MEDIUM;
  ```

- Created role-based access control:

  ```
  sql
  CopyEdit
  CREATE ROLE readonly;
  GRANT SELECT ON *.* TO readonly;
  GRANT readonly TO 'reportuser'@'%';
  ```

- Tested user login with limited access using MySQL Workbench

---

7. AWS Billing Alerts Setup

- Created an AWS Budget with cost threshold set to `$0`
- Configured email alerts to trigger when usage crosses the limit
- Ensured budget applies to all linked services in Mumbai region