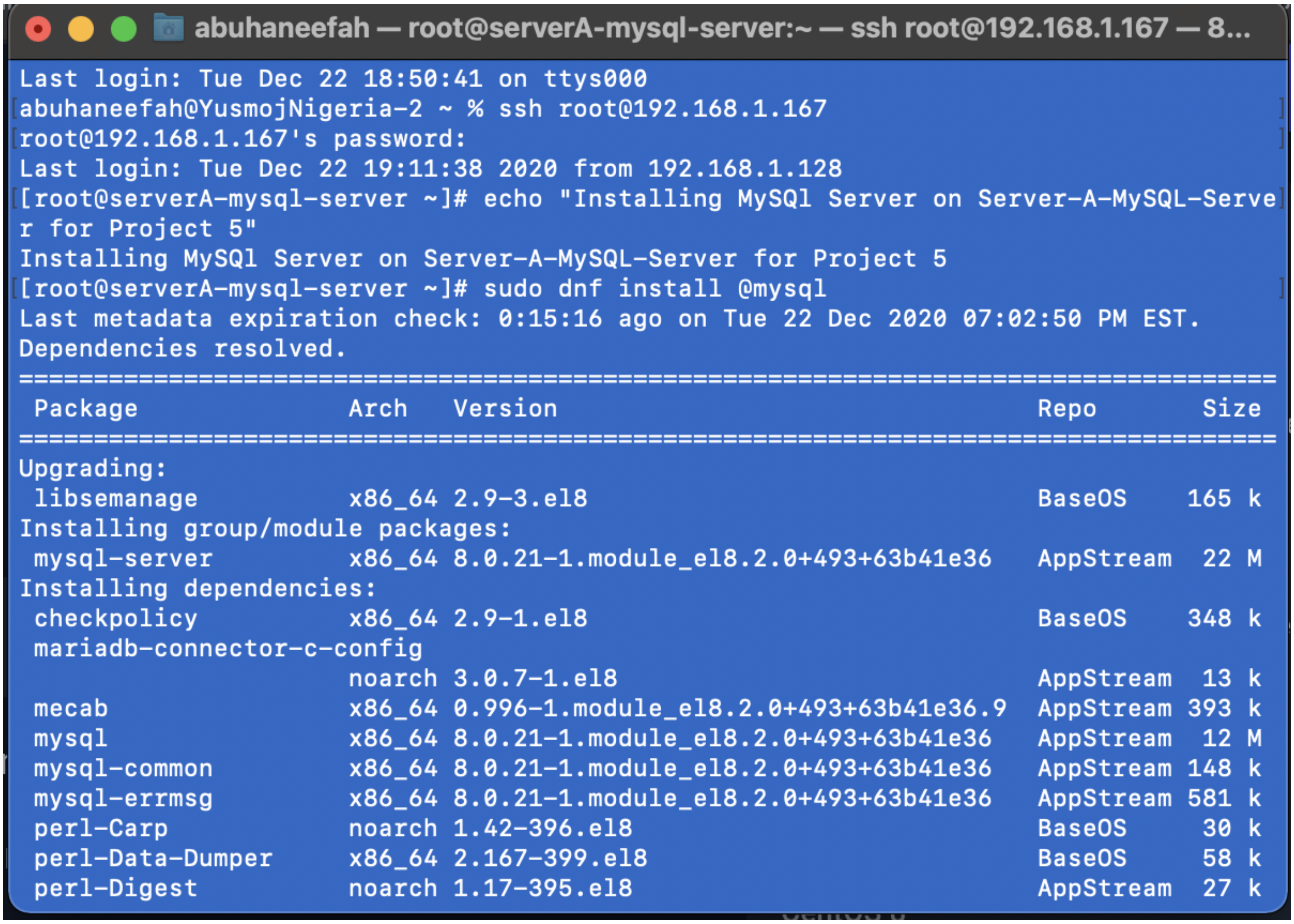
MYQSL SERVER-CLIENT PROJECT

Step-by-step Procedure

- Installed Linux Server for both Client and Server
- Installed mysql on Server
  - Installing MySQL 8.0 on CentOS 8
  - package manager as root or user with sudo privileges:

```
sudo dnf install @mysql
```

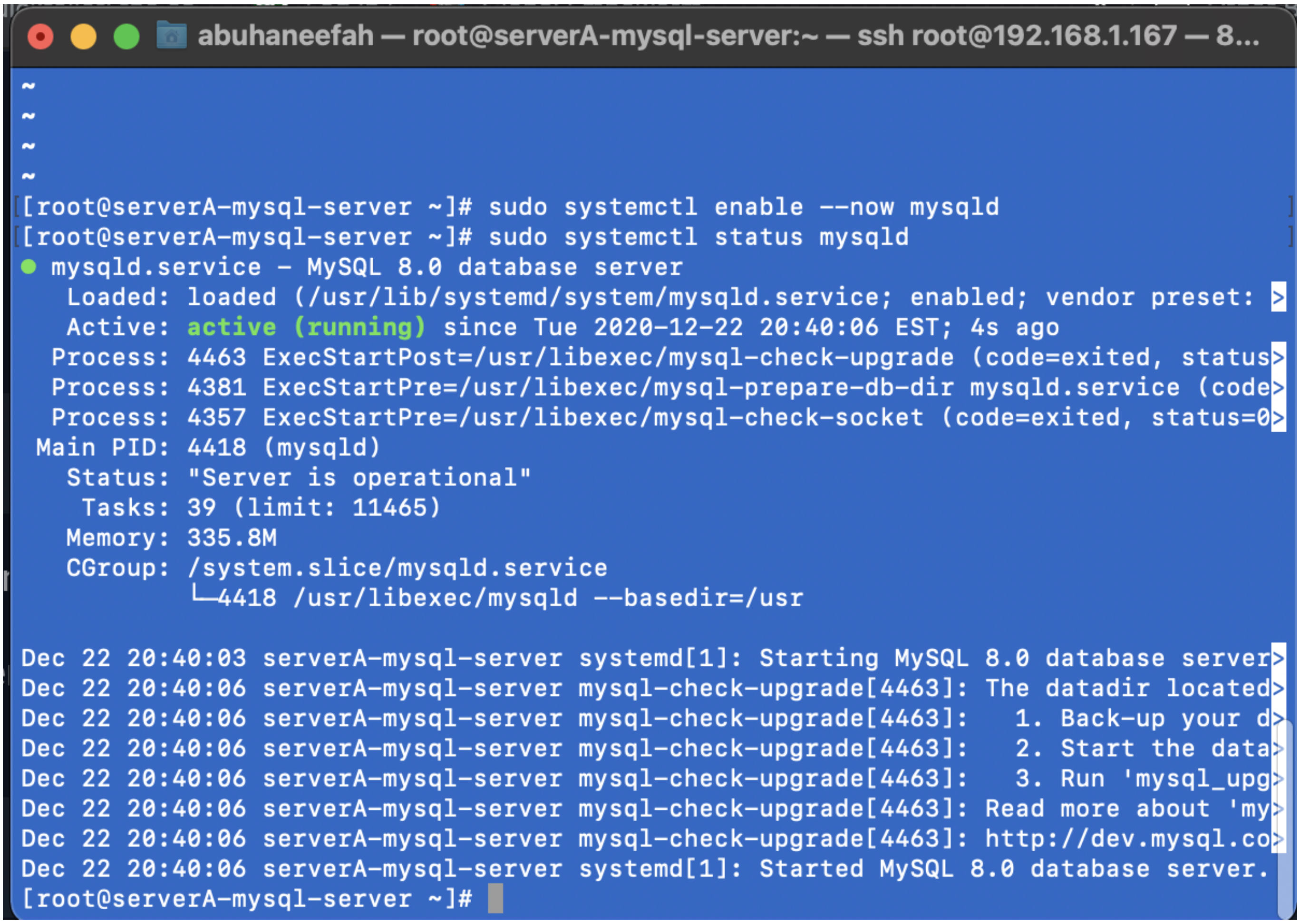
> The @mysql module installs MySQL and all dependencies.


abuhaneefah — root@serverA-mysql-server:~ — ssh root@192.168.1.167 — 8...

```
Last login: Tue Dec 22 18:50:41 on ttys000
[abuhaneefah@YusmojNigeria-2 ~ % ssh root@192.168.1.167
[root@192.168.1.167's password:
Last login: Tue Dec 22 19:11:38 2020 from 192.168.1.128
[[root@serverA-mysql-server ~]# echo "Installing MySQl Server on Server-A-MySQL-Serve
r for Project 5"
Installing MySQl Server on Server-A-MySQL-Server for Project 5
[[root@serverA-mysql-server ~]# sudo dnf install @mysql
Last metadata expiration check: 0:15:16 ago on Tue 22 Dec 2020 07:02:50 PM EST.
Dependencies resolved.
======================================================================================
 Package                 Arch    Version                                Repo       Size
======================================================================================
Upgrading:
 libsemanage             x86_64 2.9-3.el8                               BaseOS    165 k
Installing group/module packages:
 mysql-server            x86_64 8.0.21-1.module_el8.2.0+493+63b41e36   AppStream  22 M
Installing dependencies:
 checkpolicy             x86_64 2.9-1.el8                               BaseOS    348 k
 mariadb-connector-c-config
                         noarch 3.0.7-1.el8                             AppStream  13 k
 mecab                   x86_64 0.996-1.module_el8.2.0+493+63b41e36.9  AppStream 393 k
 mysql                   x86_64 8.0.21-1.module_el8.2.0+493+63b41e36   AppStream  12 M
 mysql-common            x86_64 8.0.21-1.module_el8.2.0+493+63b41e36   AppStream 148 k
 mysql-errmsg            x86_64 8.0.21-1.module_el8.2.0+493+63b41e36   AppStream 581 k
 perl-Carp               noarch 1.42-396.el8                            BaseOS     30 k
 perl-Data-Dumper        x86_64 2.167-399.el8                           BaseOS     58 k
 perl-Digest             noarch 1.17-395.el8                            AppStream  27 k
```

- Once the installation is completed, I start the MySQL service and enabled it to automatically start on boot by running the following command:

```
sudo systemctl enable --now mysqld
```


abuhaneefah — root@serverA-mysql-server:~ — ssh root@192.168.1.167 — 8...

```
~
~
~
~
[root@serverA-mysql-server ~]# sudo systemctl enable --now mysqld
[root@serverA-mysql-server ~]# sudo systemctl status mysqld
● mysqld.service - MySQL 8.0 database server
   Loaded: loaded (/usr/lib/systemd/system/mysqld.service; enabled; vendor preset: >
   Active: active (running) since Tue 2020-12-22 20:40:06 EST; 4s ago
  Process: 4463 ExecStartPost=/usr/libexec/mysql-check-upgrade (code=exited, status>
  Process: 4381 ExecStartPre=/usr/libexec/mysql-prepare-db-dir mysqld.service (code>
  Process: 4357 ExecStartPre=/usr/libexec/mysql-check-socket (code=exited, status=0>
 Main PID: 4418 (mysqld)
   Status: "Server is operational"
    Tasks: 39 (limit: 11465)
   Memory: 335.8M
   CGroup: /system.slice/mysqld.service
           └─4418 /usr/libexec/mysqld --basedir=/usr

Dec 22 20:40:03 serverA-mysql-server systemd[1]: Starting MySQL 8.0 database server>
Dec 22 20:40:06 serverA-mysql-server mysql-check-upgrade[4463]: The datadir located>
Dec 22 20:40:06 serverA-mysql-server mysql-check-upgrade[4463]:   1. Back-up your d>
Dec 22 20:40:06 serverA-mysql-server mysql-check-upgrade[4463]:   2. Start the data>
Dec 22 20:40:06 serverA-mysql-server mysql-check-upgrade[4463]:   3. Run 'mysql_upg>
Dec 22 20:40:06 serverA-mysql-server mysql-check-upgrade[4463]: Read more about 'my>
Dec 22 20:40:06 serverA-mysql-server mysql-check-upgrade[4463]: http://dev.mysql.co>
Dec 22 20:40:06 serverA-mysql-server systemd[1]: Started MySQL 8.0 database server.
[root@serverA-mysql-server ~]#
```

- To check whether the MySQL server is running, I used:

```
sudo systemctl status mysqld
```

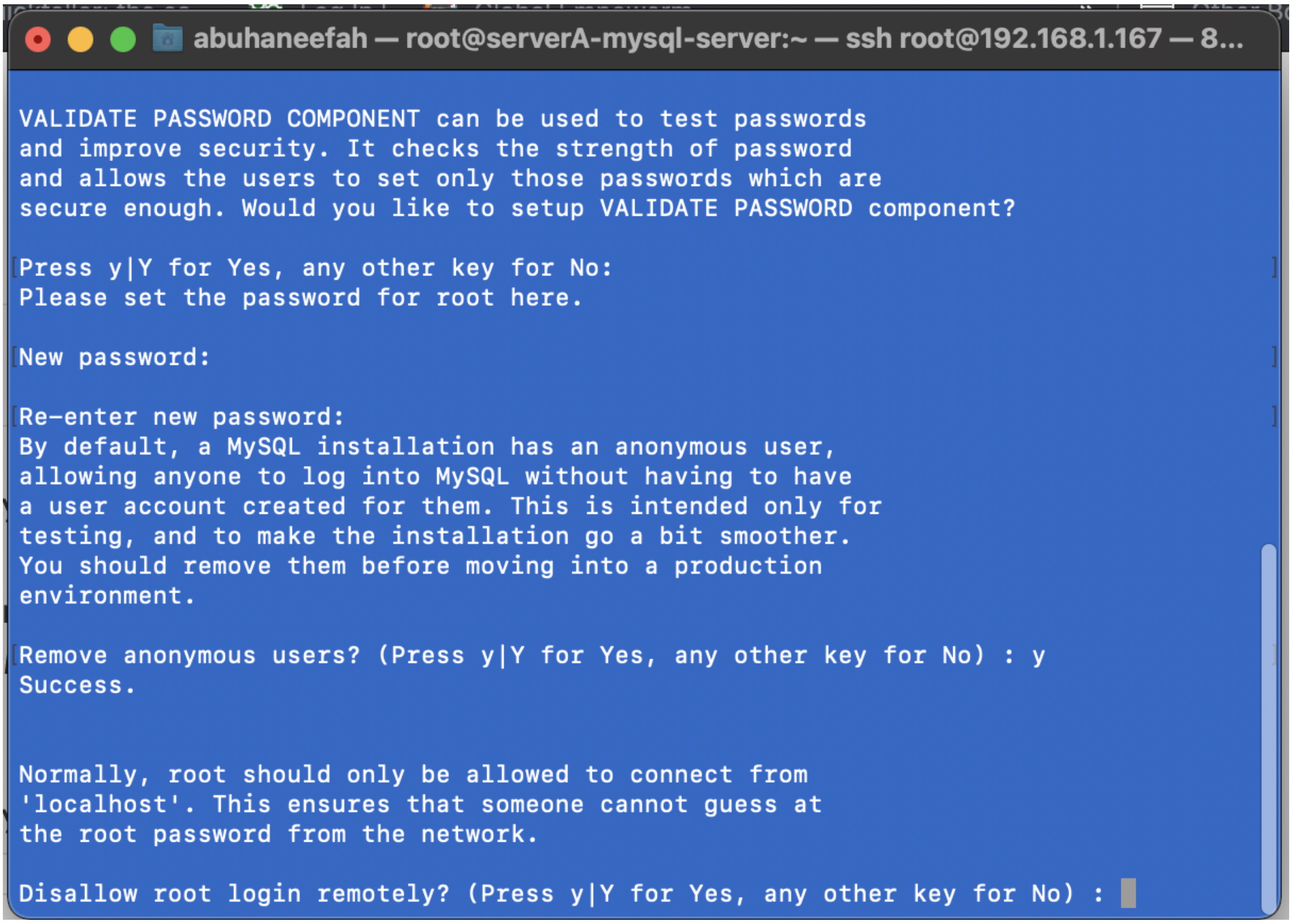
> Below is the output:

```
mysqld.service - MySQL 8.0 database server
Loaded: loaded (/usr/lib/systemd/system/mysqld.service; enabled; vendor
preset: disabled)
Active: active (running) since Thu 2020-12-24 22:09:39 EST; 15s ago
...
```

- Securing MySQL: To secure the mysql server, I ran the mysql_secure_installation script that performs several security-related operations and sets the MySQL root password:

```
sudo mysql_secure_installation
```

> I followed the system prompt to complete the secure system validation process.


abuhaneefah — root@serverA-mysql-server:~ — ssh root@192.168.1.167 — 8...

```
VALIDATE PASSWORD COMPONENT can be used to test passwords
and improve security. It checks the strength of password
and allows the users to set only those passwords which are
secure enough. Would you like to setup VALIDATE PASSWORD component?

Press y|Y for Yes, any other key for No:
Please set the password for root here.

New password:

Re-enter new password:
By default, a MySQL installation has an anonymous user,
allowing anyone to log into MySQL without having to have
a user account created for them. This is intended only for
testing, and to make the installation go a bit smoother.
You should remove them before moving into a production
environment.

Remove anonymous users? (Press y|Y for Yes, any other key for No) : y
Success.


Normally, root should only be allowed to connect from
'localhost'. This ensures that someone cannot guess at
the root password from the network.

Disallow root login remotely? (Press y|Y for Yes, any other key for No) :
```

- Perform restart with

```
sudo systemctl restart mysqld
```

- To access the MYSQL Database, I used the below command

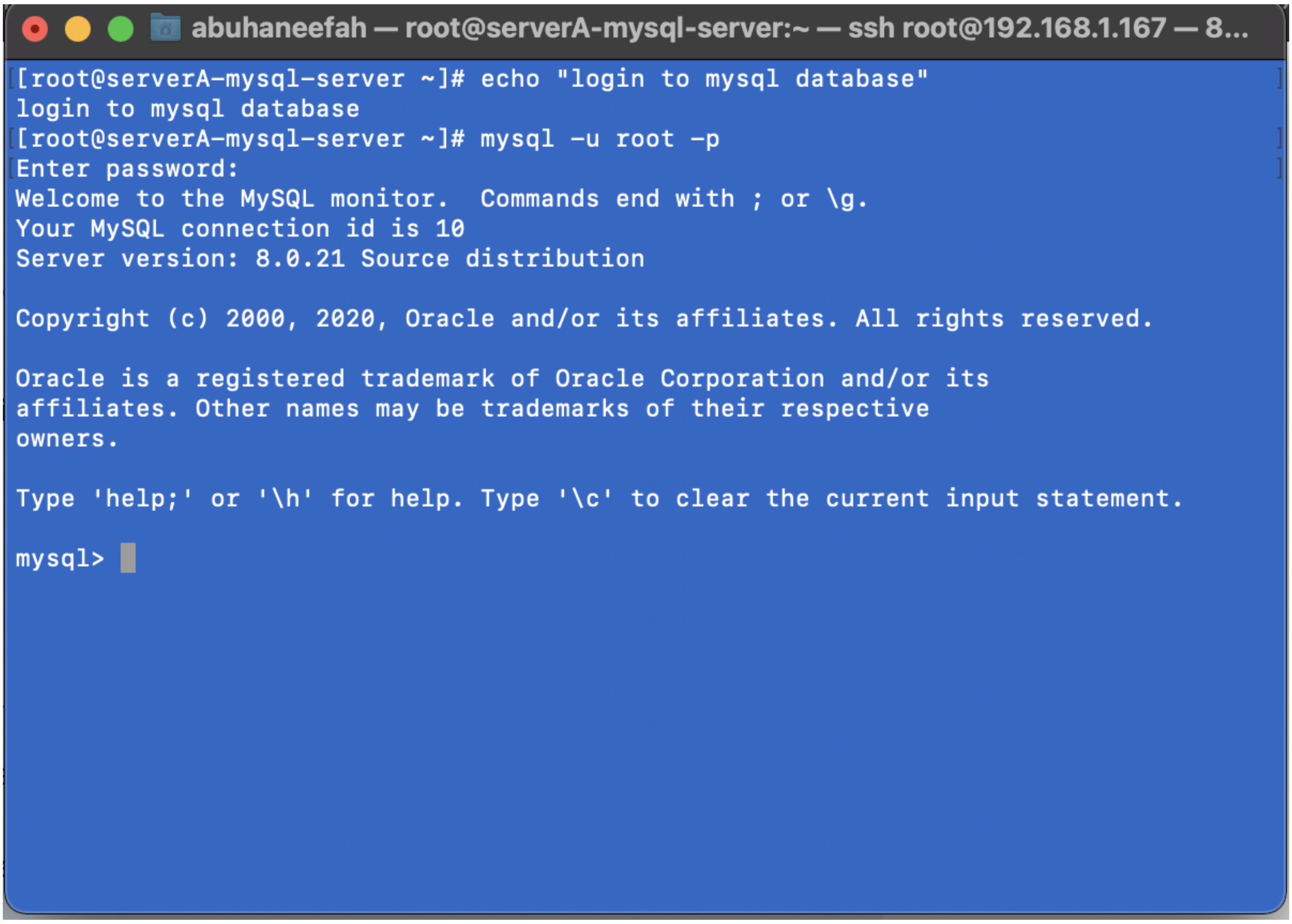
```
mysql -u root -p
```

- Enter the root password

> This is the output:

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 10
Server version: 8.0.21 Source distribution

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights
reserved.

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>
```



abuhaneefah — root@serverA-mysql-server:~ — ssh root@192.168.1.167 — 8...

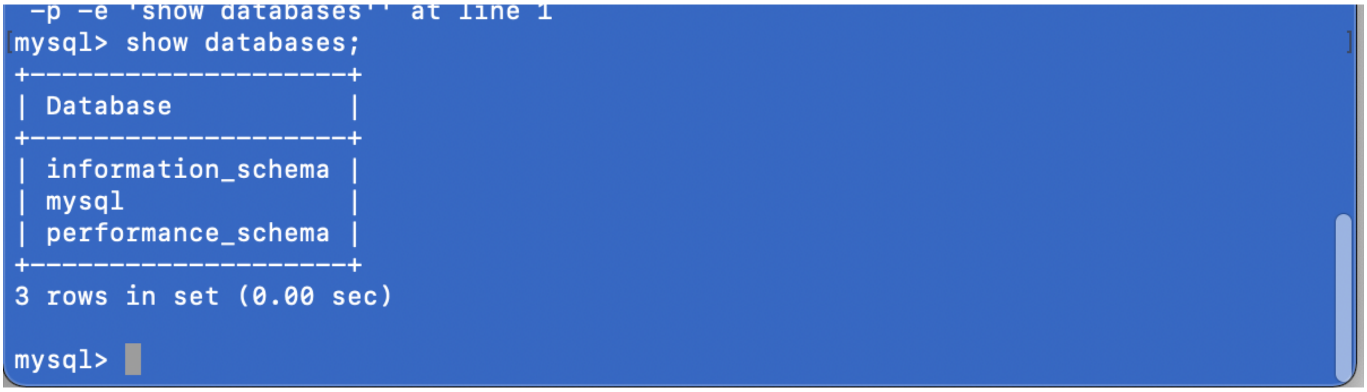
```
[root@serverA-mysql-server ~]# echo "login to mysql database"
login to mysql database
[root@serverA-mysql-server ~]# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 10
Server version: 8.0.21 Source distribution

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

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>
```

- To view the database you've created simply issue the following command:

```
SHOW DATABASES;
```

> This is the output;

```
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
4 rows in set (0.01 sec)
```



```
 -p -e 'show databases'' at line 1
[mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
+--------------------+
3 rows in set (0.00 sec)

mysql> 
```

Creating and configuring mysql instance on Server to allow remote connection

Configuring the database instance This section discusses how to create a new database instance. Although a new database instance is recommended. To configure a MySQL database instance:

- I logged in to the database server as root or any user.
- Entered the MySQL root user's password when prompted.
- Entered the following commands in the order shown to create a database instance named project5 with username leye:

1. Create new database

```
create database project5;
```

> Output:

```
Query OK, 1 row affected (0.05 sec)
```

2. Create a new user

```
create user 'leye'@'localhost' IDENTIFIED BY 'p@ssw0rd';
```

> Output:

```
Query OK, 0 rows affected (0.02 sec)
```

3. Grant access to the new user on the database

```
GRANT ALL ON project5.* TO 'leye'@'localhost';
```

> Output:

```
Query OK, 0 rows affected (0.02 sec)
```

4. Flush privilege

```
FLUSH PRIVILEGES;
```

> Output:

```
Query OK, 0 rows affected (0.01 sec)
```

- Enter exit to quit the command prompt.
- To view the database you've created simply issue the following command:

```
SHOW DATABASES;
```

> This is the output;

```
+---------------------+
| Database            |
+---------------------+
| information_schema |
| mysql               |
| performance_schema |
| project5            |
| sys                 |
+---------------------+
5 rows in set (0.00 sec)
```

- Verify the database:

```
mysql -u leye -p
```

Granting remote access to the mysql Server user on the Client Server

- Login the the msql server as root

```
mysql -u root -p
```

- Grant permission to the desired user to use for remote connection on the client server

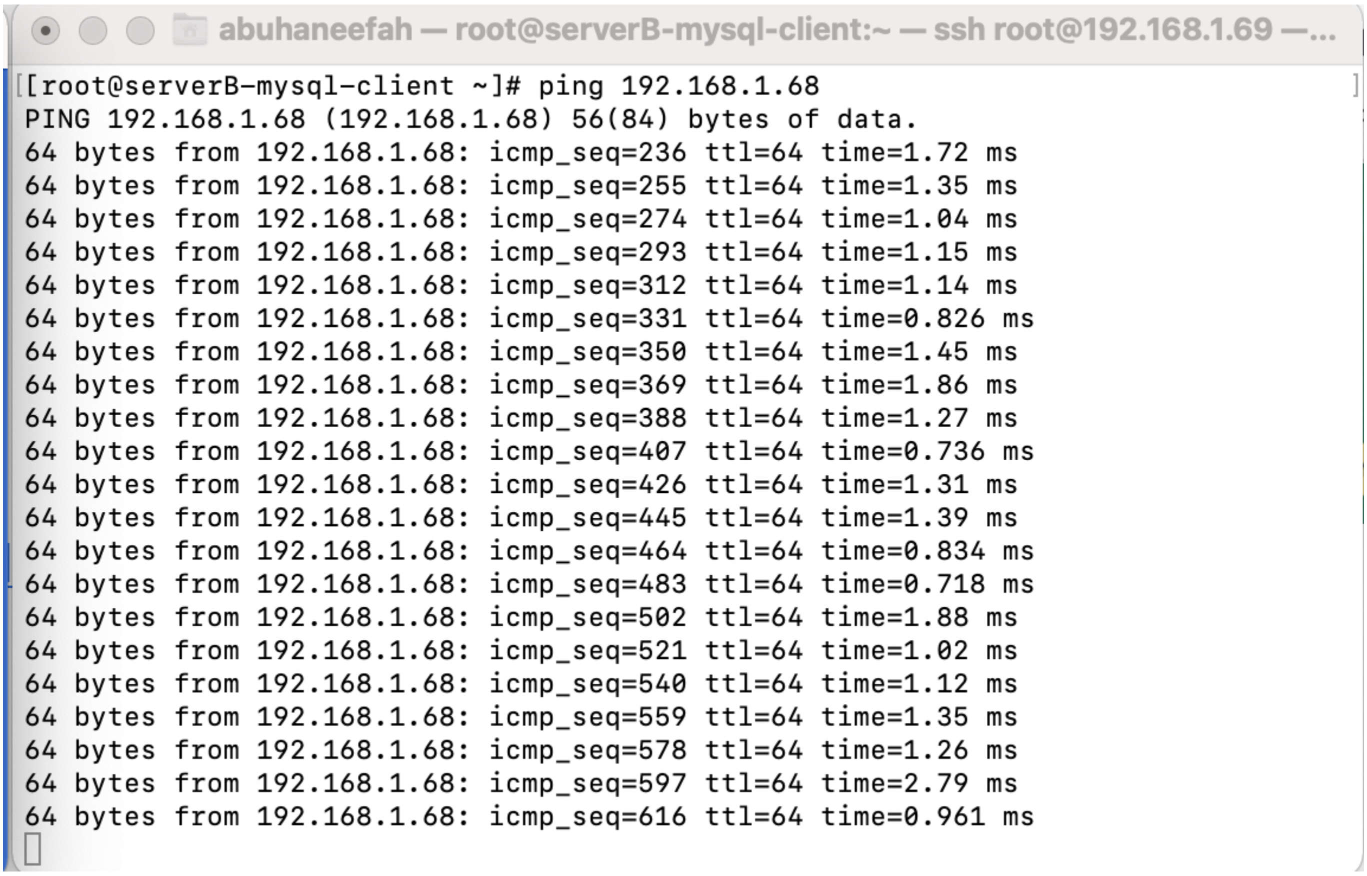
```
GRANT ALL PRIVILEGES ON *.* TO ‘leye’@‘192.168.1.69’;
```

- Use the flush privilege command

```
FLUSH PRIVILEGES;
```

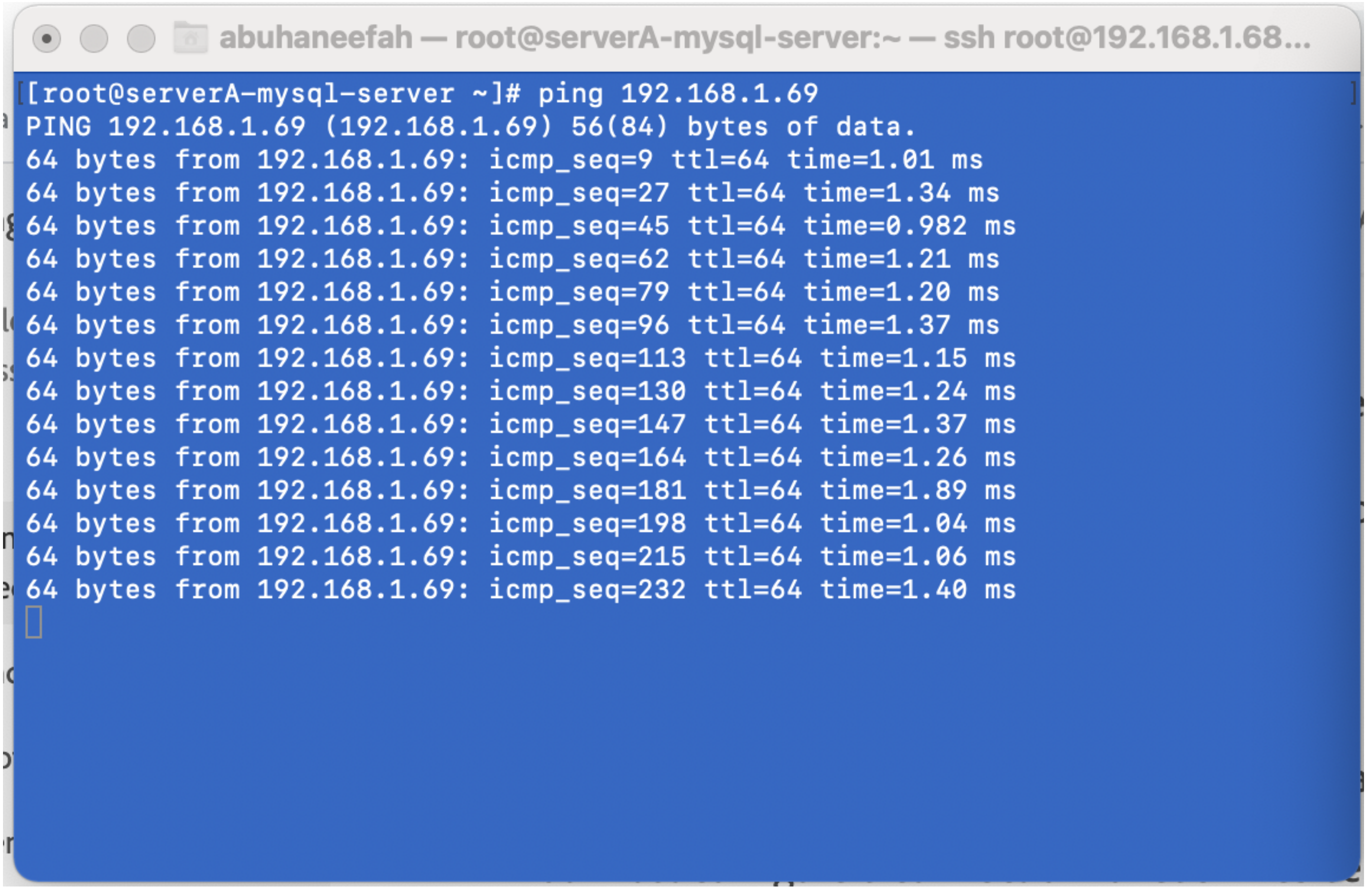
- Ping both server and client IP to confirm connectivity

```
ping 192.168.1.68
```


abuhaneefah — root@serverB-mysql-client:~ — ssh root@192.168.1.69 —...

```
[root@serverB-mysql-client ~]# ping 192.168.1.68
PING 192.168.1.68 (192.168.1.68) 56(84) bytes of data.
64 bytes from 192.168.1.68: icmp_seq=236 ttl=64 time=1.72 ms
64 bytes from 192.168.1.68: icmp_seq=255 ttl=64 time=1.35 ms
64 bytes from 192.168.1.68: icmp_seq=274 ttl=64 time=1.04 ms
64 bytes from 192.168.1.68: icmp_seq=293 ttl=64 time=1.15 ms
64 bytes from 192.168.1.68: icmp_seq=312 ttl=64 time=1.14 ms
64 bytes from 192.168.1.68: icmp_seq=331 ttl=64 time=0.826 ms
64 bytes from 192.168.1.68: icmp_seq=350 ttl=64 time=1.45 ms
64 bytes from 192.168.1.68: icmp_seq=369 ttl=64 time=1.86 ms
64 bytes from 192.168.1.68: icmp_seq=388 ttl=64 time=1.27 ms
64 bytes from 192.168.1.68: icmp_seq=407 ttl=64 time=0.736 ms
64 bytes from 192.168.1.68: icmp_seq=426 ttl=64 time=1.31 ms
64 bytes from 192.168.1.68: icmp_seq=445 ttl=64 time=1.39 ms
64 bytes from 192.168.1.68: icmp_seq=464 ttl=64 time=0.834 ms
64 bytes from 192.168.1.68: icmp_seq=483 ttl=64 time=0.718 ms
64 bytes from 192.168.1.68: icmp_seq=502 ttl=64 time=1.88 ms
64 bytes from 192.168.1.68: icmp_seq=521 ttl=64 time=1.02 ms
64 bytes from 192.168.1.68: icmp_seq=540 ttl=64 time=1.12 ms
64 bytes from 192.168.1.68: icmp_seq=559 ttl=64 time=1.35 ms
64 bytes from 192.168.1.68: icmp_seq=578 ttl=64 time=1.26 ms
64 bytes from 192.168.1.68: icmp_seq=597 ttl=64 time=2.79 ms
64 bytes from 192.168.1.68: icmp_seq=616 ttl=64 time=0.961 ms
```

```
ping 192.168.1.69
```



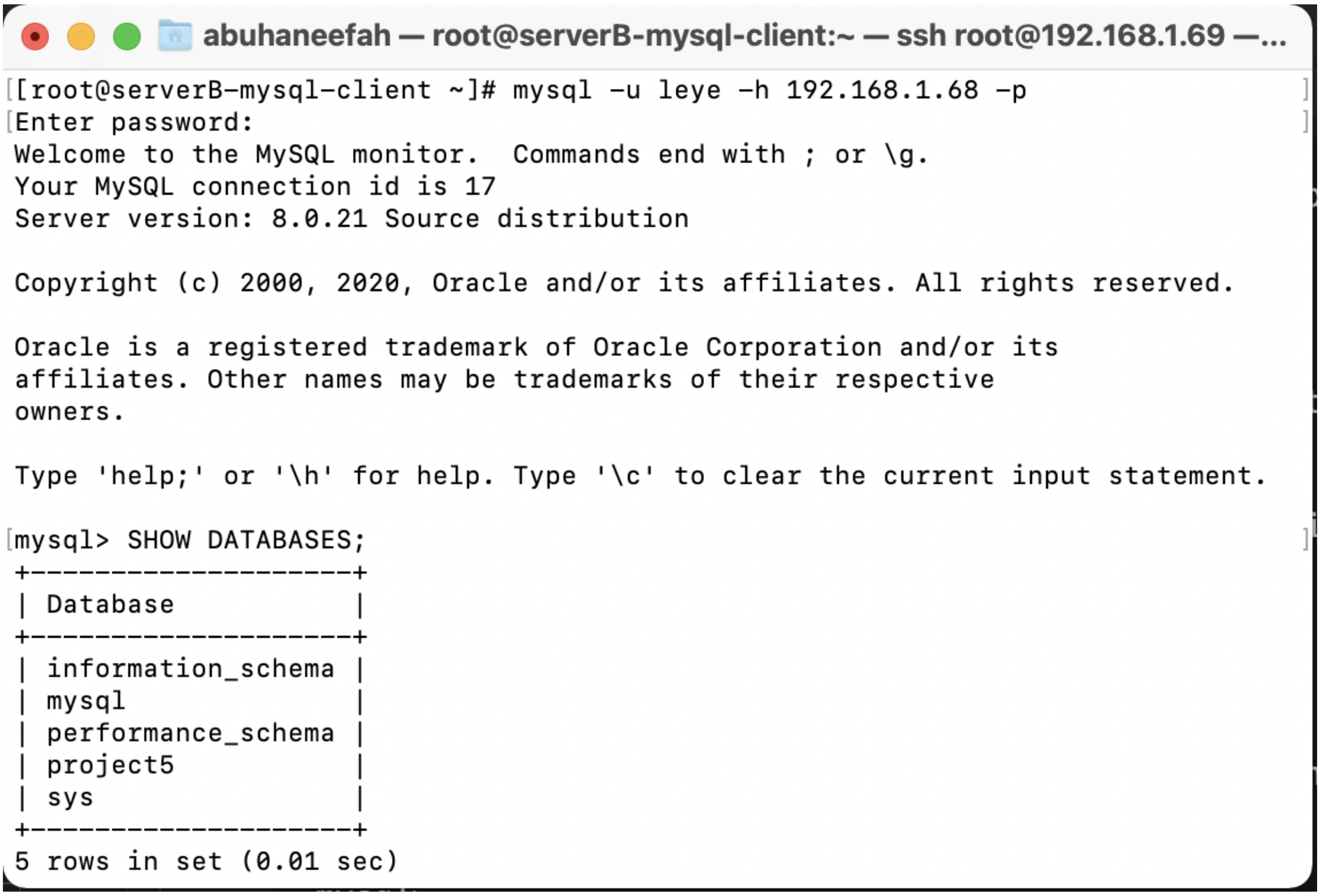
abuhaneefah — root@serverA-mysql-server:~ — ssh root@192.168.1.68...

```
[root@serverA-mysql-server ~]# ping 192.168.1.69
PING 192.168.1.69 (192.168.1.69) 56(84) bytes of data.
64 bytes from 192.168.1.69: icmp_seq=9 ttl=64 time=1.01 ms
64 bytes from 192.168.1.69: icmp_seq=27 ttl=64 time=1.34 ms
64 bytes from 192.168.1.69: icmp_seq=45 ttl=64 time=0.982 ms
64 bytes from 192.168.1.69: icmp_seq=62 ttl=64 time=1.21 ms
64 bytes from 192.168.1.69: icmp_seq=79 ttl=64 time=1.20 ms
64 bytes from 192.168.1.69: icmp_seq=96 ttl=64 time=1.37 ms
64 bytes from 192.168.1.69: icmp_seq=113 ttl=64 time=1.15 ms
64 bytes from 192.168.1.69: icmp_seq=130 ttl=64 time=1.24 ms
64 bytes from 192.168.1.69: icmp_seq=147 ttl=64 time=1.37 ms
64 bytes from 192.168.1.69: icmp_seq=164 ttl=64 time=1.26 ms
64 bytes from 192.168.1.69: icmp_seq=181 ttl=64 time=1.89 ms
64 bytes from 192.168.1.69: icmp_seq=198 ttl=64 time=1.04 ms
64 bytes from 192.168.1.69: icmp_seq=215 ttl=64 time=1.06 ms
64 bytes from 192.168.1.69: icmp_seq=232 ttl=64 time=1.40 ms
```

Remotely Connecting to from the Client Server

- After fulfilling all configuration processes, below is the output of the remote connection to the main server;

```
[root@serverB-mysql-client ~]# mysql -u leye -h 192.168.1.68 -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 17
Server version: 8.0.21 Source distribution

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights
reserved.

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> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| project5           |
| sys                |
+--------------------+
5 rows in set (0.01 sec)

mysql>
```


abuhaneefah — root@serverB-mysql-client:~ — ssh root@192.168.1.69 —...

```
[root@serverB-mysql-client ~]# mysql -u leye -h 192.168.1.68 -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 17
Server version: 8.0.21 Source distribution

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

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> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| project5           |
| sys                |
+--------------------+
5 rows in set (0.01 sec)
```

Issues Encountered and Resolution

1. I was unable to grant access to both root and my created user to access the server remotely.

> Error message below:

```
ERROR 1410 (42000): You are not allowed to create a user with GRANT
```

2. I modified the /etc/my.cnf to include **bind-address**

```
bind-address    0.0.0.0
```

This allow the server to accept connection from any IP address.

3. Added the destination port to the **IP-Table** of the Server.

```
sudo iptables -A INPUT -p tcp --destination-port 3306 -j ACCEPT
```

4. Added the destination port to the **Firewall** to allow TCP connection.

```
sudo firewall-cmd --permanent --zone=public --add-port=3306/tcp
```

5. Reloaded the **Firewall** to implement the new settings.

```
sudo firewall-cmd --reload
```

6. Then restarted the **mysql server** and connect remotely.