

# How to Access a MariaDB Database From Another Host?

written by sysadmin | 29 November 2025

[The previous article](#) already explained how to access a MariaDB database from localhost. This article will explain how to access the MariaDB database from another host.

## Problem

How to access a MariaDB database from another host?

## Solution

To access the MariaDB database, you can follow the format below:

```
mariadb -h your_server_ip -u username -P port_number -p
```

By default, the MariaDB database uses port 3306, so if your MariaDB database uses port 3306, then there is no need to write the port when executing the command to access the MariaDB database. I have a MariaDB database server with an IP of 192.168.56.101 and an Ubuntu server with an IP of 192.168.56.11. I want to access the MariaDB database via the Ubuntu server using the user john and run the command:

```
mariadb -h 192.168.56.101 -u john -p
```

But I got an error like below:

ERROR 2002 (HY000): Can't connect to server on '192.168.56.101' (115)

```
sysadmin@ubuntu2404:/etc/mysql$ mariadb -h 192.168.56.101 -u john -p
Enter password:
ERROR 2002 (HY000): Can't connect to server on '192.168.56.101' (115)
sysadmin@ubuntu2404:/etc/mysql$
```



Error when accessing the MariaDB from another host

Below are the steps to access the MariaDB database:

### 1. Open the port

Open port 3306 on both servers. If you use RockyLinux/AlmaLinux/RHEL for your servers, use the command below:

```
firewall-cmd --add-port=3306/tcp --permanent  
firewall-cmd --reload
```

But if you use Ubuntu/Debian for your server, type the command below:

```
sudo ufw allow 3306
```

### 2. Check the MariaDB version

You should know that to access the MariaDB database, you need a mariadb-client package whose version is the same as the MariaDB database version. If the mariadb-client version is different from the mariadb database version, then there will usually be an error. To install the mariadb-client package, you can go to [this page](#).

### 3. Grant access

On the database server, run a query with the format below:

```
GRANT option ON db_name.* TO username@"ip_address" IDENTIFIED BY "password";
```

For example, if user john wants to access the entire MariaDB database from the host with IP 192.168.56.11, then use the query below:

```
GRANT ALL ON *.* TO john@"192.168.56.11" IDENTIFIED BY "123456";
```

```
MariaDB [(none)]> GRANT ALL ON *.* TO john@'192.168.56.11' IDENTIFIED BY '123456';  
Query OK, 0 rows affected (0.002 sec)
```

```
MariaDB [(none)]> SELECT User,Host FROM mysql.user;
```

| User        | Host          |
|-------------|---------------|
| john        | 192.168.56.11 |
| mariadb.sys | localhost     |
| mysql       | localhost     |
| root        | localhost     |

4 rows in set (0.001 sec)

```
MariaDB [(none)]> 
```



Grant access for user john

### Warning

If you want user john to be able to access only the Zabbix database, use the query below:

```
GRANT ALL ON zabbix.* TO john@"192.168.56.101" IDENTIFIED BY "qwerty";
```

And use the query below if you want user john to be able to view only the Zabbix database:

```
GRANT SELECT ON zabbix.* TO john@"192.168.56.101" IDENTIFIED BY "qwerty";
```

## 4. Configure the file

Go to the file `/etc/mysql/mariadb.conf.d/50-server.cnf` and change the bind-address item to be as below:

```
bind-address          = 0.0.0.0
```

### Warning

You can also change the bind-address item in the `/etc/mysql/my.cnf` file by adding the following script to the file:

```
[mysqld]  
bind-address = 0.0.0.0
```

## 5. Restart MariaDB

After that, restart the MariaDB service using the command below:

```
sudo systemctl restart mariadb
```

Then, try to access it again, and you should be able to access the database like in the picture below:

```
sysadmin@ubuntu2404:~$ mariadb -h 192.168.56.101 -u john -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 35
Server version: 12.1.2-MariaDB-ubu2404 mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

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

MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| db_office |
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
5 rows in set (0.001 sec)

MariaDB [(none)]> 
```

Succeed to access MariaDB from another host

## Note

If you want your MariaDB database to only be accessed from a certain IP, for example, IP 192.168.56.11, then use the command below if your distro uses Ubuntu/Debian:

```
sudo ufw allow from 192.168.56.11 to any port 3306
```

But, if you use RockyLinux/AlmaLinux/RHEL, type the below command to open port 3306 only from IP 192.168.56.11

```
firewall-cmd --zone=public --add-rich-rule 'rule family=ipv4 source address=192.168.56.11 port port=3306 protocol=tcp accept'
firewall-cmd --reload
firewall-cmd --list-rich-rules
```

## References

[mariadb.com](https://mariadb.com)  
[tencentcloud.com](https://tencentcloud.com)  
[webdock.io](https://webdock.io)

---

## How to Access a PostgreSQL Database From Another Host?

written by sysadmin | 29 November 2025

I want to access a PostgreSQL database from another host.

### Problem

How to access a PostgreSQL database from another host?

### Solution

I have 2 servers, the PostgreSQL server IP is 192.168.56.101, and the Ubuntu server IP is 192.168.56.11. I want to access the PostgreSQL database from the Ubuntu server. By default, use the format below if you want to access PostgreSQL:

```
psql -h ip_db -u username -d db_name -p port_number
```

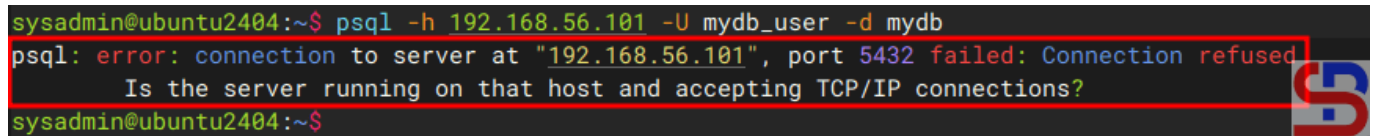
By default, the PostgreSQL database uses port 5432, so if

you access your PostgreSQL using port 5432, you don't need to write the port. I accessed my PostgreSQL by writing the command:

```
psql -h 192.168.56.101 -u postgres -d mydb
```

But I got an error like the image below:

```
psql: error: connection to server at '192.168.56.101', port 5432 failed:
Connection refused
Is the server running on that host and accepting TCP/IP connections?
```

A terminal window screenshot showing a command prompt on a system named 'sysadmin@ubuntu2404'. The user enters the command 'psql -h 192.168.56.101 -U mydb\_user -d mydb'. The output shows a red error message: 'psql: error: connection to server at "192.168.56.101", port 5432 failed: Connection refused' followed by the question 'Is the server running on that host and accepting TCP/IP connections?'. The prompt returns to 'sysadmin@ubuntu2404:~\$'.

```
sysadmin@ubuntu2404:~$ psql -h 192.168.56.101 -U mydb_user -d mydb
psql: error: connection to server at "192.168.56.101", port 5432 failed: Connection refused
Is the server running on that host and accepting TCP/IP connections?
sysadmin@ubuntu2404:~$
```

Error when accessing the PostgreSQL database

The following are the steps to access a PostgreSQL database from another host:

## 1. Open port

Open port 5432 on each server. If you use RockyLinux/AlmaLinux/RHEL for your servers, use the command below:

```
firewall-cmd --add-port=5432/tcp --permanent
firewall-cmd --reload
```

But if you use Ubuntu/Debian for your server, type the command below:

```
sudo ufw allow 5432
```

## 2. Config postgresql.conf

In the database server, go to **/etc/postgresql/18/main/postgresql.conf** file if you use PostgreSQL 18, but back up the file first:

```
sudo cp /etc/postgresql/18/main/postgresql.conf
/etc/postgresql/18/main/postgresql.conf.ori
```

After that, change the script below in the file:

```
#listen_addresses = 'localhost'
```

to

```
listen_addresses = '*'
```

### 3. Configure pg\_hba.conf

After that, go to **/etc/postgresql/18/main/pg\_hba.conf** if you use PostgreSQL 18, but backup the file first:

```
sudo cp /etc/postgresql/18/main/pg_hba.conf  
/etc/postgresql/18/main/pg_hba.conf.ori
```

Then, add your IP host (in this case, IP is 192.168.56.11) like the below script:

```
host      all             all             192.168.56.11/32      scram-sha-256
```

#### Warning

You must change the file **/etc/postgresql/18/main/postgresql.conf** if you want to use scram-sha-256 encryption as described in [this article](#).

### 4. Restart PostgreSQL

Restart postgresql service using the command below:

```
sudo systemctl restart postgresql
```

Now, try to access the PostgreSQL again, and you should be able to access the database like the command below:

```

sysadmin@docker:~$ psql -h localhost -U mydb_user -d mydb
Password for user mydb_user:
psql (18.0 (Ubuntu 18.0-1.pgdg24.04+3))
SSL connection (protocol: TLSv1.3, cipher: TLS_AES_256_GCM_SHA384, compression: off, ALPN: postgresql)
Type "help" for help.

mydb=> INSERT INTO employee (name,age,city) VALUES ('queen',23,'Florida');
INSERT 0 1
mydb=> select * from employee;
 name | age | city 
-----+----+-----
 bob  |  21 | New York
 John  |  22 | Chicago
 Steve |  22 | Kansas
 trump |  20 | Orlando
 paul  |  20 | Texas
 queen |  23 | Florida
(6 rows)

```

Do CRUD on the database

## Note

If you have an error when you want to display data, like the image below:

**ERROR: permission denied for table employee**

```

sysadmin@ubuntu2404:~$ psql -h 192.168.56.101 -U mydb_user -d mydb
Password for user mydb_user:
psql (18.0 (Ubuntu 18.0-1.pgdg24.04+3))
SSL connection (protocol: TLSv1.3, cipher: TLS_AES_256_GCM_SHA384, compression: off, ALPN: postgresql)
Type "help" for help.

mydb=> SELECT * FROM employee;
ERROR: permission denied for table employee
mydb=>

```

Error when querying in the postgresql

Then you have to change the owner of the database and the tables in the database. You can follow how to change it via [this page](#).

## References

[dev.to](https://dev.to)  
[dba.stackexchange.com](https://dba.stackexchange.com)  
[prisma.io](https://prisma.io)  
[tigerdata.com](https://tigerdata.com)