MySQL

mysql> create user 'M.Poortarigh'@'%' identified by '12!@qwQW';

mysql> grant all privileges on *.* to 'M.Poortarigh'@'%' with grant option;

mysql> flush privileges;

mysql> select * from information_schema.user_privileges;

mysql> select User();

mysql> show grants for 'poortarigh'@'localhost';

mysql> select * from mysql.user;

GRANT ALL ON yourdatabase.* TO youruser@'*' IDENTIFIED BY 'yourpassword';

ALTER USER 'maryam'@'localhost' IDENTIFIED WITH mysql_native_password BY '12!@qwQW';

With a single query we are changing the **auth_plugin** to **mysql_native_password**

mysql> alter user 'M.Poortarigh'@'%' identified with mysql_native_password by 'mynewpassword:12!@qwQW';

host : 194.5.195.125
username: M.Poortarigh
password : mynewpassword:12!@qwQW

root@vm8895219718:~# mysql systemctl restart

sudo service mysql stop

mysql> UPDATE mysql.user SET authentication_string =PASSWORD('password') WHERE User='root';

mysql> UPDATE mysql.user SET authentication_string = PASSWORD('new_password') WHERE User = 'root' AND Host = 'localhost';

```
SELECT user,authentication_string,plugin,host FROM mysql.user;
```

mysql> select user, host, password from user where user like 'poorta%';

Error :Can't connect to [local] MySQL server

https://dev.mysql.com/doc/refman/8.0/en/can-not-connect-to-server.html#:~:text=normally%20means%20that%20there%20is,firewall%20or%20port%20blocking%20service.

1. check the port (3306)
2. check your own system firewall
3. check mysql server connection management **(** `skip_networking` , `bind_address`)

maryam@poortarigh:$mysqladmin -h 194.5.195.125 --port=3306 version
mysqladmin: connect to server at '194.5.195.125' failed
error: 'Can't connect to MySQL server on '194.5.195.125' (110)'
Check that mysqld is running on 194.5.195.125 and that the port is 3306.
You can check this by doing 'telnet 194.5.195.125 3306'

maryam@poortarigh:$telnet 194.5.195.125 3306
Trying 194.5.195.125...
telnet: Unable to connect to remote host: Connection timed out

A MySQL client on Unix can connect to the mysqld server in two different ways:

1. By using a Unix socket file to connect through a file in the file system (default `/tmp/mysql.sock`), or

2. by using TCP/IP, which connects through a port number.

A Unix socket file connection is faster than TCP/IP, but can be used only when connecting to a server on the same computer. A Unix socket file is used if you do not specify a host name or if you specify the special host name `localhost`.

Error (2002)

The error (2002) `Can't connect to …` normally means that there is no MySQL server running on the system or that you are using an incorrect Unix socket file name or TCP/IP port number when trying to connect to the server. You should also check that the TCP/IP port you are using has not been blocked by a firewall or port blocking service.

Error (2003)

The error (2003) `Can't connect to MySQL server on 'server' (10061)` indicates that the network connection has been refused. You should check that there is a MySQL server running, that it has network connections enabled, and that the network port you specified is the one configured on the server.

ps xa | grep mysqld

(/usr/sbin/mysqld)

Start by checking whether there is a process named **mysqld** running on your server host. (Use ps xa | grep mysqld on Unix or the Task Manager on Windows.) If there is no such process, you should start the server.

If a **mysqld** process is running, you can check it by trying the following commands. The port number or Unix socket file name might be different in your setup. `host_ip` represents the IP address of the machine where the server is running.

```
shell> mysqladmin version
shell> mysqladmin variables
shell> mysqladmin -h `hostname` version variables
shell> mysqladmin -h `hostname` --port=3306 version
shell> mysqladmin -h host_ip version
shell> mysqladmin --protocol=SOCKET --socket=/tmp/mysql.sock version
```

Note the use of backticks rather than forward quotation marks with the hostname command; these cause the output of hostname (that is, the current host name) to be substituted into the **mysqladmin** command. If you have no hostname command or are running on Windows, you can manually type the host name of your machine (without backticks) following the `-h` option. You can also try `-h 127.0.0.1` to connect with TCP/IP to the local host.

Make sure that the server has not been configured to ignore network connections or (if you are attempting to connect remotely) that it has not been configured to listen only locally on its network interfaces. If the server was started with the **skip_networking** system variable enabled, it cannot accept TCP/IP connections at all. If the server was started with the **bind_address** system variable set to `127.0.0.1`, it listens for TCP/IP connections only locally on the loopback interface and does not accept remote connections.

https://dev.mysql.com/doc/refman/8.0/en/server-system-variables.html#sysvar_skip_networking

`skip_networking`

This variable controls whether the server permits TCP/IP connections. By default, it is disabled (permit TCP connections). If enabled, the server permits only local (non-TCP/IP) connections and all interaction with **mysqld** must be made using named pipes or shared memory (on Windows) or Unix socket files (on Unix).

Command-Line Format	`--skip-networking[={OFF|ON}]`
System Variable	`skip_networking`

`bind_address`

Command-Line Format	`--bind-address=addr`
System Variable	`bind_address`

It describes aspects of how the MySQL server manages client connections including Network Interfaces and Connection Manager Threads.

The MySQL server listens on one or more network sockets for TCP/IP connections. Each socket is bound to one address, but it is possible for an address to map onto multiple network interfaces. To specify how the server should listen for TCP/IP connections, set the `bind_address` system variable at server startup. The server also has an `admin_address` system variable that enables administrative connections on a dedicated interface.

Check to make sure that there is no firewall blocking access to MySQL. Your firewall may be configured on the basis of the application being executed, or the port number used by MySQL for communication (3306 by default). Under Linux or Unix, check your IP tables (or similar) configuration to ensure that the port has not been blocked.

If you do a normal install of MySQL on Debian, it will be configured to block external connections to the database. This means that you still need to tell MySQL that external access is OK. To do this, you need to update the bind address for MySQL.

This is configured in `my.cnf`, which, on Debian based systems, is located in `/etc/mysql/my.cnf`.

In there, find the section that says

[mysqld]

In there, you must make sure that

- the line `skip-networking` is either commented (comments start with a '#') or not there, and
- Bind-address is set to either `0.0.0.0` (which it is if there is no line bind-address) or to your server's IP-address.

After doing this, you should restart your MySQL service.

Starting the Server

https://dev.mysql.com/doc/refman/8.0/en/starting-server.html

systemctl status mysql.service

journalctl -xe

sudo service mysql start

sudo ps aux | grep -E mysql

sudo kill -9 pid

Start the MySQL server like this if your installation includes mysqld_safe:

```
shell> bin/mysqld_safe --user=mysql &
```

Note

For Linux systems on which MySQL is installed using RPM packages, server startup and shutdown is managed using systemd rather than mysqld_safe, and **mysqld_safe** is not installed.

systemd is a Linux initialization system and service manager that includes features like on-demand starting of daemons, mount and automount point maintenance, snapshot support, and processes tracking using Linux control groups. systemd provides a logging daemon and other tools and utilities to help with common system administration tasks.

Start the server like this if your installation includes systemd support:

```
shell> systemctl start mysqld
```

Substitute the appropriate service name if it differs from `mysqld` (for example, `mysql` on SLES systems).

Error log path:

/ var / log / mysql / error.log

https://dev.mysql.com/doc/refman/8.0/en/using-systemd.html

Managing MySQL Server with systemd

If you install MySQL using an RPM or Debian package on the following Linux platforms, server startup and shutdown is managed by systemd:

Debian family platforms:

1. Debian platforms

2. Ubuntu platforms

https://dev.mysql.com/doc/refman/8.0/en/mysqladmin.html

mysqladmin — A MySQL Server Administration Program

"mysqladmin" is a command-line interface for administrators to perform server administration tasks. It support a number of commonly used commands like:

- "mysqladmin shutdown" - Shuts down the server.
- "mysqladmin ping" - Checks if the server is alive or not.
- "mysqladmin status" - Displays several important server status values.
- "mysqladmin version" - Displays version information of the server.
- "mysqladmin create databaseName" - Creates a new database.
- "mysqladmin drop databaseName" - Drops an existing database.

Here are some reasons the `Can't connect to local MySQL server` error might occur:

1. **mysqld** is not running on the local host. Check your operating system's process list to ensure the **mysqld** process is present.

"mysqld" is MySQL server daemon program which runs quietly in background on your computer system. Invoking "mysqld" will start the MySQL server on your system. Terminating "mysqld" will shutdown the MySQL server.

which mysqld or

dpkg --get-selections | grep mysql or

ps - aux | grep mysqld or

sudo service mysql status

2. Someone has removed the Unix socket file that mysqld uses (`/tmp/mysql.sock` by default). For example, you might have a cron job that removes old files from the `/tmp` directory. You can always run mysqladmin version to check whether the Unix socket file that mysqladmin is trying to use really exists. The fix in this case is to change the cron job to not remove `mysql.sock` or to place the socket file somewhere else.

maryam@poortarigh:$ sudo find / -name "*.sock" or
`netstat -ln | grep mysql`

('/run/mysqld/mysqld.sock')

3. You have started the **mysqld** server with the `--socket=/path/to/socket` option, but forgotten to tell client programs the new name of the socket file. If you change the socket path name for the server, you must also notify the MySQL clients. You can do this by providing the same `--socket` option when you run client programs. You also need to ensure that clients have permission to access the `mysql.sock` file. To find out where the socket file is, you can do:

```
shell> netstat -ln | grep mysql
```

```
shell> mysqladmin --protocol=SOCKET --socket=/run/mysqld/mysqld.sock version
```

Error : mysqladmin: connect to server at 'localhost' failed
error: 'Access denied for user 'maryam'@'localhost' (using password: NO)'

4. You are using Linux and one server thread has died (dumped core). In this case, you must kill the other mysqld threads (for example, with kill) before you can restart the MySQL server.

5. The server or client program might not have the proper access privileges for the directory that holds the Unix socket file or the socket file itself. In this case, you must either change the access privileges for the directory or socket file so that the server and clients can access them, or restart mysqld with a `--socket` option that specifies a socket file name in a directory where the server can create it and where client programs can access it.

6. If you get the error message `Can't connect to MySQL server on some_host`, you can try the following things to find out what the problem is:

- Check whether the server is running on that host by executing `telnet some_host 3306` and pressing the Enter key a couple of times. (3306 is the default MySQL port number. Change the value if your server is listening to a different port.) If there is a MySQL server running and listening to the port, you should get a response that includes the server's version number. If you get an error such as `telnet: Unable to connect to remote host: Connection refused`, then there is no server running on the given port.

7. If the server is running on the local host, try using mysqladmin -h localhost variables to connect using the Unix socket file. Verify the TCP/IP port number that the server is configured to listen to (it is the value of the `port` variable.)

- If you are running under Linux and Security-Enhanced Linux (SELinux) is enabled, see Section 6.7, “SELinux”.

Error :Can't connect to MySQL server on (ip or domain name

https://stackoverflow.com/questions/21221220/cant-connect-to-mysql-server-on-ip-or-domain-name

Changes you can make in your own system to resolve the problem:

If you do a normal install of MySQL on Debian, it will be configured to block external connections to the database. This means that you still need to tell MySQL that external access is OK. To do this, you need to update the bind address for MySQL.

This is configured in `my.cnf`, which, on Debian based systems, is located in `/etc/mysql/my.cnf`.

In there, find the section that says

[mysqld]
In there, you must make sure that

- the line `skip-networking` is either commented (comments start with a '#') or not there, and
- Bind-address is set to either `0.0.0.0` (which it is if there is no line bind-address) or to your server's IP-address.

After doing this, you should restart your MySQL service. Then you need to create a user that is allowed remote access. This can be done with a SQL query:

GRANT ALL ON yourdatabase.* TO youruser@'*' IDENTIFIED BY 'yourpassword';

You can switch out the asterisk for the IP-address you will connect from, if it's the same every time.

Finally , you need to open port 3306 (the port MySQL uses) on your firewall. This usually isn't neccesary as it is already open on most systems, but it can be done using the following iptables command.

/sbin/iptables -A INPUT -i eth0 -p tcp --destination-port 3306 -j ACCEPT

`sudo /sbin/iptables-save (`service iptables save)

Error : Unrecognised service

Iptables is an extremely flexible firewall utility. Basically, it works based on a set of specified rules. These rules specify the action to take on server traffic.

So, when anyone tries to establish a connection to the server, *iptables* look for a rule in its list to match it to. If it doesn't find one, it takes the default action. The action can be ALLOW, DENY or REJECT traffic.

```
service iptables save
```

Usually, this command will work for RHEL/ Red Hat / CentOS. In Ubuntu, to save the changes in firewall rules, we use the following command

```
sudo /sbin/iptables-save
```

I was trying to find .cnf file hence I did the following:

sudo find / -name "*.cnf"

I edited **/etc/mysql/mysql.conf.d/mysqld.cnf** based on :

strace mysql ";" 2>&1 | grep cnf

and changed **`bind-address`** to my local IP address.

Go to /etc/mysql/my.cnf

vi /etc/mysql/my.cnf

Now you see and compare below if you find difference then update

```
# Default Homebrew MySQL server config
[mysqld]
```

```
# Only allow connections from localhost
bind-address = 0.0.0.0
```

Now press button => esc and :wq (vi commands)

Restart the MySQL =>

mysql systemctl restart

sudo service mysql stop

sudo service mysql start

https://stackoverflow.com/questions/41645309/mysql-error-access-denied-for-user-rootlocalhost

https://dba.stackexchange.com/questions/211604/mysql-user-password-vs-authentication-string

Error :

mysql> select user,authentication_string from mysql.user;
seems the root does not have any authentification string

Error : `mysqladmin --protocol=SOCKET --socket=/run/mysqld/mysqld.sock version`

Error : mysqladmin: connect to server at 'localhost' failed
error: 'Access denied for user 'maryam'@'localhost' (using password: NO)'