If you try to connect to your mysql server from remote machine, and run into error like below, this article is for you.

ERROR 1130 (HY000): Host ‘1.2.3.4’ is not allowed to connect to this MySQL server

Change mysql config

Start with editing mysql config file

```
vim /etc/mysql/my.cnf
```

Comment out following lines.

```
#bind-address           = 127.0.0.1  #skip-networking
```

If you do not find skip-networking line, add it and comment out it.

Restart mysql server.

```
service mysql restart
```

Change GRANT privilege

You may be surprised to see even after above change you are not getting remote access or getting access but not able to all databases.

By default, mysql username and password you are using is allowed to access mysql-server locally. So need to update privilege.

Run a command like below to access from all machines.

```
mysql> GRANT ALL PRIVILEGES ON *.* TO 'USERNAME'@'%' IDENTIFIED BY
'PASSWORD' WITH GRANT OPTION;
```

Run a command like below to give access from specific IP.

```
mysql> GRANT ALL PRIVILEGES ON *.* TO 'USERNAME'@'1.2.3.4' IDENTIFIED BY
'PASSWORD' WITH GRANT OPTION;
```

You can replace 1.2.3.4 with your IP. You can run above command many times to GRANT access from multiple IPs.

You can also specify a separate USERNAME & PASSWORD for remote access.

You can check final outcome by:

```
SELECT * from information_schema.user_privileges where grantee like "'USERNAME'%";
```

Finally, you may also need to run:

```
mysql> FLUSH PRIVILEGES;
```

Test Connection

From terminal/command-line:

```
mysql -h HOST -u USERNAME -pPASSWORD
```

If you get a mysql shell, don’t forget to run show databases; to check if you have right privileges from remote machines.

Bonus-Tip: Revoke Access

If you accidentally grant access to a user, then better have revoking option handy.

Following will revoke all options for USERNAME from all machines:

```
mysql> REVOKE ALL PRIVILEGES, GRANT OPTION FROM 'USERNAME'@'%';
```

Following will revoke all options for USERNAME from particular IP:

```
mysql> REVOKE ALL PRIVILEGES, GRANT OPTION FROM 'USERNAME'@'1.2.3.4';
```

Its better to check information_schema.user_privileges table after running REVOKE command.

If you see USAGE privilege after running REVOKE command, its fine. It is as good as no privilege at all. I am not sure if it can be revoked.

```
+------------------------+--------------+------------------------+--------------+ | GRANTEE                | TABLE_CATALOG | PRIVILEGE_TYPE          | IS_GRANTABLE |
+------------------------+--------------+------------------------+--------------+ | 'USERNAME'@'%'         | def           | USAGE                   | NO           |
+------------------------+--------------+------------------------+--------------+
```

To get a list of MySQL users:

```
mysql> select user,host from mysql.user;  +------------------+--------------+ | user             | host
  |  +------------------+--------------+ | root             | %            | | root             | 127.0.0.1    | |
debian-sys-maint | localhost    | | root             | localhost    | | root             | stack.domain |
+------------------+--------------+  8 rows in set (0.01 sec)
```

The above output shows a list of existing MySQL accounts. Note that a MySQL account has two components: user and host. This allows the same user to use different MySQL accounts depending on which host they are connecting from. "%" is a wildcard character interpreted as "any" host.

To find the privilege(s) granted to a particular MySQL account:

```
mysql> show grants for 'root'@'%';  +-----------------------------------------------+ | Grants for
root@%                                  |  +-----------------------------------------------+ | GRANT ALL
PRIVILEGES ON *.* TO 'root'@'%' .... |  +-----------------------------------------------+  1 row in set
(0.00 sec)
```

As above, when you specify a particular MySQL account to check for privileges, use an account name constructed by user name and host name concatenated by "@".

If you want to add a new database called foo for user bar and remote IP 202.54.10.20 then you

need to type the following commands at mysql> prompt:

```
mysql> CREATE DATABASE foo;
 mysql> GRANT ALL ON foo.* TO bar@'202.54.10.20' IDENTIFIED BY 'PASSWORD';
```

How Do I Grant Access To An Existing Database?

Let us assume that you are always making connection from remote IP called 202.54.10.20 for database called webdb for user webadmin, To grant access to this IP address type the following command At mysql> prompt for existing database, enter:

```
 mysql> update db set Host='202.54.10.20' where Db='webdb';
 mysql> update user set Host='202.54.10.20' where user='webadmin';
```