TUTORIAL

How to Connect to a MySQL Server Remotely with MySQL Workbench

MySQL

By Jon Schwenn

Published on October 21, 2016 215.6k

Introduction

Your database server contains tables full of important data. Querying this data graphically on your local computer is the easiest way to interact with your database. But connecting remotely to your database server usually entails configuring MySQL to listen on every interface, restricting access to port `3306` with your firewall, and configuring user and host permissions for authentication. And allowing connections to MySQL directly can be a security concern.

Using tools like HeidiSQL for Windows, Sequel Pro for macOS, or the cross-platform MySQL Workbench, you can connect securely to your database over SSH, bypassing those cumbersome and potentially insecure steps. This brief tutorial will show you how to connect to a remote database using MySQL Workbench.

Prerequisites

To complete this tutorial, you will need:

- A server running MySQL that is accessible via SSH. For example, you can follow the tutorial How To Install MySQL on Ubuntu 14.04 to get up and running quickly.
- MySQL Workbench installed on your local machine, which is available for all major platforms, including Windows, macOS, Ubuntu Linux, RedHat Linux, and Fedora. Visit the MySQL Workbench Downloads page to download the installer for your operating system.

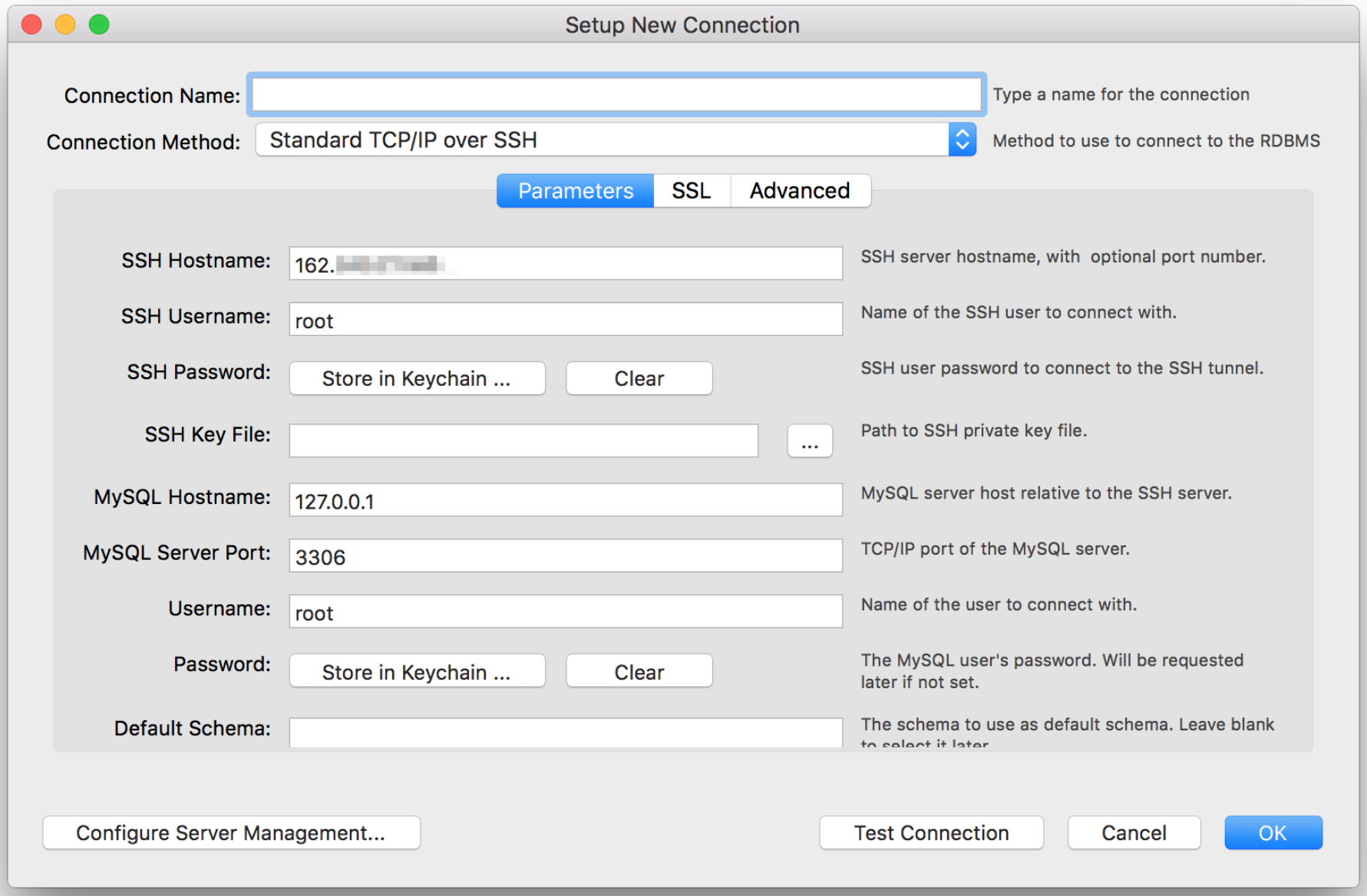
You will also need the following information about the database server you plan to use:

- The public IP address of the server running MySQL.
- The server's SSH Port if configured differently than port `22`.
- A user account with SSH access to the server, with a password or public key.
- The username and password for the MySQL account you wish to use.

Connecting to the Database Server With SSH

Once you've installed MySQL Workbench on your computer, launch the program. Create a new connection by clicking the **+** icon next to **MySQL Connections** in the main window.

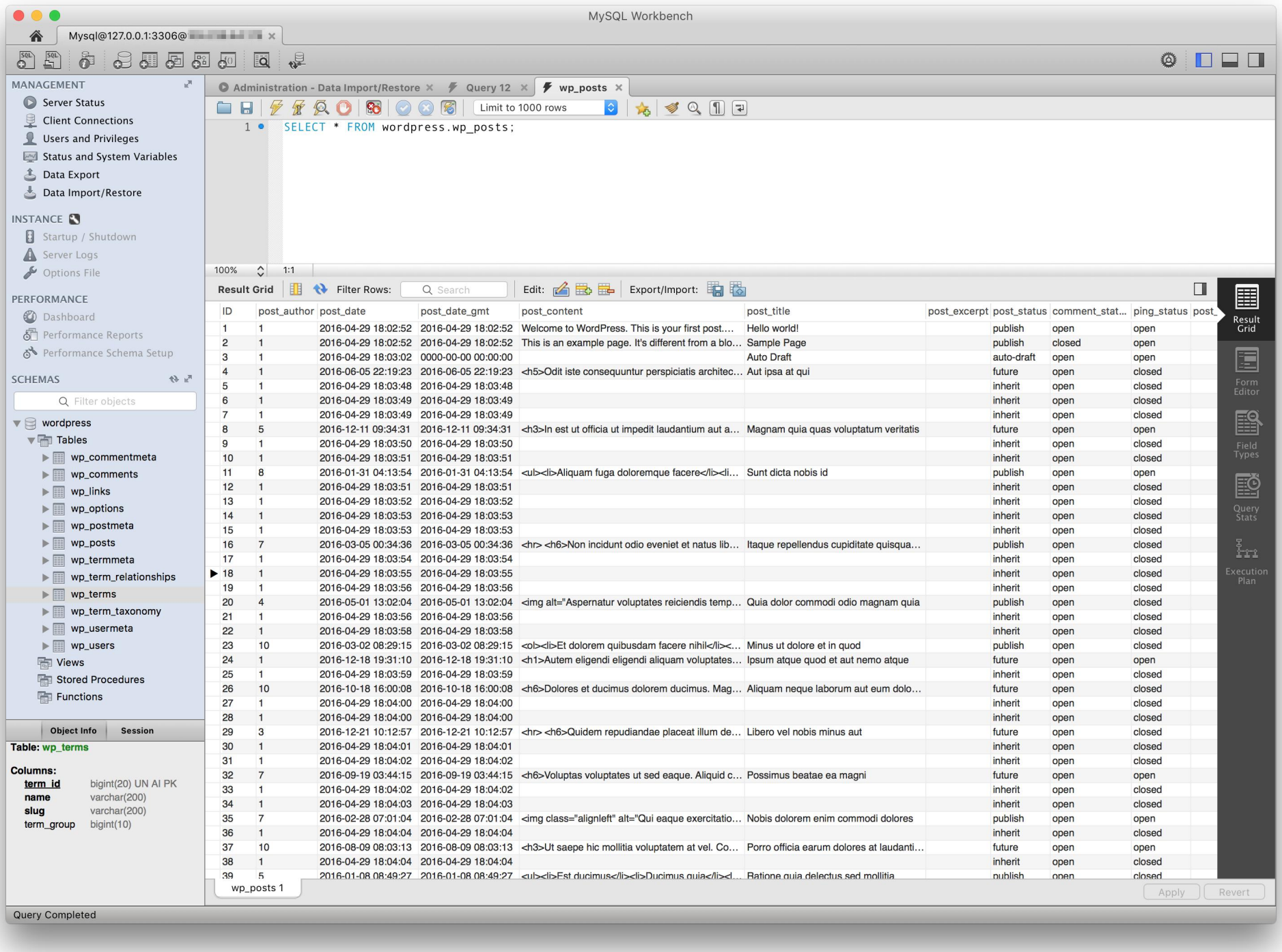
You'll be presented with the **Connect to Database** window, which looks like the follwing figure:



To create the connection, enter the following details:

1. For **Connection Name**, enter any name you'd like that helps you identify the connection you're making later. This might be something like `database_for_myapp` or something more descriptive.
2. Change the **Connection Method** to **Standard TCP/IP over SSH**.
3. For **SSH Hostname**, enter your MySQL server's IP address. If your server accepts SSH connections on a different port, enter the IP address, followed by a colon and port number.
4. For **SSH Username**, enter the username you use to log into the server via SSH.
5. For **SSH Password**, enter the password you use for your SSH user. If you use public keys instead of passwords, select an SSH key for authentication.
6. For **MySQL Hostname** and **MySQL Server Port**, use the default values.
7. For **Username**, enter the MySQL username.
8. For **Password**, you can either enter the password or leave it blank. If you do not store the MySQL password in MySQL Workbench, a prompt will request the password each time you attempt to connect to the database.
9. Choose **Test Connection** to ensure your settings are correct.
0. Choose **OK** to create the connection.

Once you've connected to your database, you can view the details of the MySQL instance, including database status, current connections, and database configuration, as well as users and permissions. MySQL Workbench also supports importing and exporting of MySQL dump files so you can quickly back up and restore your database.



To manage your connections, select the **Database** menu and choose the **Connect to Database** option, or press `⌘U` on the Mac or `CTRL+U` on Windows and Linux systems. To connect to a different database, create a new connection using the same process you used for your first connection.

Conclusion

Using MySQL Workbench to access your remote MySQL database through an SSH tunnel is a simple and secure way to manage your databases from the comfort of your local computer. Using the connection method in this tutorial, you can bypass multiple network and security configuration changes normally required for a remote MySQL connection.

Still looking for an answer?

Ask a question

Search for more help

RELATED

Now Available: Managed MySQL Databases
Product

How To Run a PHP Job Multiple Times in a Minute with Crontab on Ubuntu 20.04
Tutorial

How To Store and Retrieve Data in MariaDB Using Python on Ubuntu 18.04
Tutorial

Comments

10 Comments

Leave a comment...

Sign In to Comment

ekekis January 30, 2017
0
Hello, it did not work!!!
I have run many tutorials, but the same error: lost connection to mysql server at ‘reading initial communication packet’ system error 0
I do not know what to do to fix it.
I need help!

Reply Report

> jeffmjack May 18, 2019
> 0
> Make sure you’ve changed the Connection Method setting at the top from “Standard (TCP/IP)” to “Standard TCP/IP over SSH”.
>
> Reply Report

WackyApps March 25, 2017
0
I have same problem. Not connecting to remote server. Do I have to enable remote connection On on mysql database?

Reply Report

```
Authentication error, unhandled exception caught in tunnel manager, please refer to logs for details.
```

Looking in my logs: (obfuscated my IP)

```
17:23:40 [INF][     SSH tunnel]: Opening SSH tunnel to 45.55.111.111
17:23:40 [WRN][sshtunnel.py:_connect_ssh:288]: IOError, probably caused by file C:\Users\cutups\AppData\Roaming\MySQL\Workbench\ssh\known_
17:23:40 [ERR][sshtunnel.py:notify_exception_error:233]: Traceback (most recent call last):
  File "C:\Program Files (x86)\MySQL\MySQL Workbench 6.3 CE\sshtunnel.py", line 298, in _connect_ssh
    look_for_keys=has_key, allow_agent=has_key)
  File "C:\Program Files (x86)\MySQL\MySQL Workbench 6.3 CE/python/site-packages\paramiko\client.py", line 301, in connect
    t.start_client()
  File "C:\Program Files (x86)\MySQL\MySQL Workbench 6.3 CE/python/site-packages\paramiko\transport.py", line 461, in start_client
    raise e
SSHException: Incompatible ssh peer (no acceptable kex algorithm)

17:23:41 [INF][     SSH tunnel]: TunnelManager.wait_connection authentication error: Authentication error, unhandled exception caught in tu
17:23:41 [ERR][     SSH tunnel]: Authentication error opening SSH tunnel: Authentication error, unhandled exception caught in tunnel manage
```

Anybody else find a solution in this case?

Reply Report

bov188 June 16, 2017

0

Make sure you change the default bind-address in MySQL settings in your droplet.

A few other sources I found online said to change this in the `my.cnf` file located in `/etc/mysql/` however when I looked it was not located there.

Navigate to the `/etc/mysql/mysql.conf.d/` directory and edit the mysqld.cnf file by entering the following command `nano mysqld.cnf` (May need to use sudo command, can't remember)

Change the line that says `bind-address = 127.0.0.1` to `bind-address = 0.0.0.0`

This will allow your server to accept connections other IP addresses.

Hope this helps others

Reply Report

> **ldaniellgoo** February 28, 2018
>
> 0
>
> didn't work
>
> Reply Report

xzx7moodxzx February 8, 2018

0

thank you for this tutorial

Did you now how to connect into database in java program?

Reply Report

crispytx September 19, 2018

1

In the past I've always used the LAMP Stack and phpMyAdmin for all of my projects. However, I just started using LEMP instead and encountered problems getting phpMyAdmin to work with the new tech stack. So I decided to give MySQL Workbench a try, but ended up running into a whole bunch of problems getting it set up.

However, now that I have everything all figured out, I figured I should leave a comment with some information that is missing from this article that was critical for me getting this all setup:

If you are using the LEMP stack on DigitalOcean, you will likely need to create a username for a database administrator to access MySQL. So you'll still use 'root' for your ssh username, but you will need to create a username to provide as the MySQL database administrator. When using MySQL Workbench, you'll be asked for BOTH names

```
mysql> CREATE USER 'developer'@'%' IDENTIFIED BY 'dev_password';
mysql> GRANT ALL ON *.* TO 'developer'@'localhost';
mysql> GRANT ALL ON *.* TO 'developer'@'%';
$ sudo service mysql restart
```

REFERENCE: https://stackoverflow.com/questions/5755819/lost-connection-to-mysql-server-at-reading-initial-communication-packet-syste/38070979#38070979

Reply Report

weilei October 12, 2018

4 For anyone who has working SSH connection but still get connection errors after following the above. These two things helped me fix:

- Generate a private SSH key without passphrase
- When using Puttygen, import your existing private key, then export it as OpenSSH. I previously used simply save private key and it didn't work with Workbench (more info here https://www.digitalocean.com/community/questions/trying-to-connect-to-mysql-database-server-through-mysql-workbench-using-ssh)

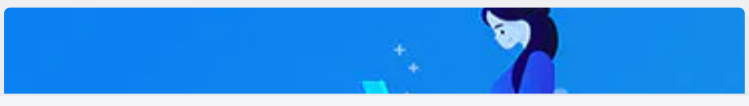
Reply Report

cibelewatt December 11, 2018

1 When your key file was generated by Putty it won't work with Workbench. You have to import it to Puttygen once again and then export it to the OpenSSH format. Then you can use this file at Workbench (you do not have to remove the passphrase in order for it to work).

Worked for me when I did it, hope it works for you!

Reply Report



you get paid; we donate to tech nonprofits.