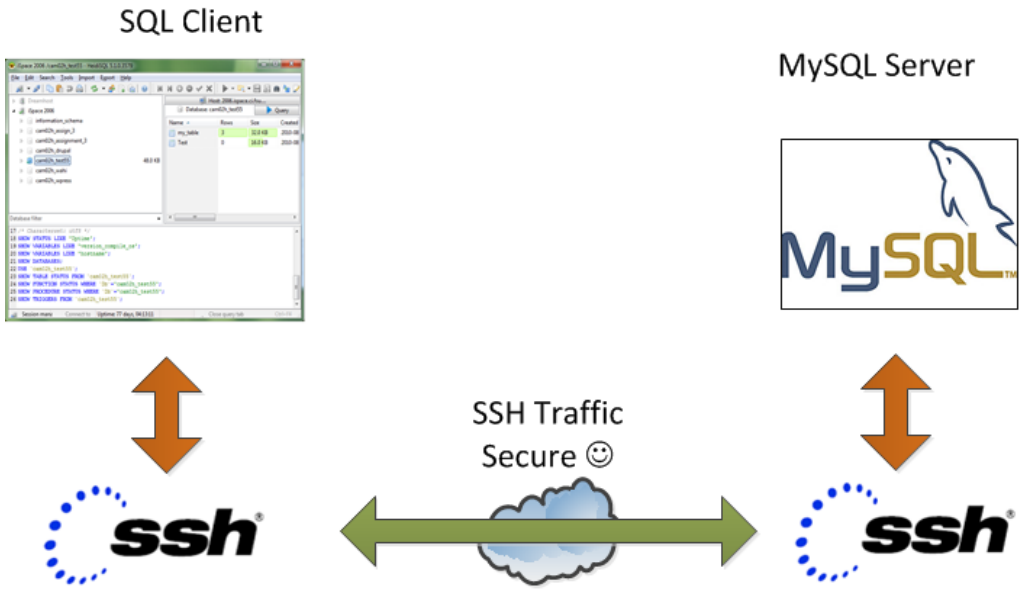
Create SSH Tunnel to MySQL server to access remote filtered MySQL port 3306 host through localhost port 3308

Author : admin



On our **Debian / CentOS / Ubuntu Linux and Windows servers** we're **running multiple MySQL servers** and **our customers sometimes need to access this servers.**
This is usually problem because **MySQL Db servers are running in a DMZ Zone with a strong firewall** and besides that **for security reasons SQLs are configured to only listen for connections coming from localhost**, I mean in config files across our **Debian Linux servers and CentOS / RHEL Linux machines** the **/etc/mysql/my.cnf** and **/etc/my.cnf** the setting for **bind-address** is *127.0.0.1*:

```
[root@centos ~]# grep -i bind-address /etc/my.cnf
bind-address            = 127.0.0.1
##bind-address  = 0.0.0.0
```

For source code developers which are accessing development SQL servers only through a VPN secured DMZ Network there are few MySQL servers witha allowed access remotely from all hosts, e.g. on those I have configured:

[root@ubuntu-dev ~]# grep -i bind-address /etc/my.cnf

bind-address = 0.0.0.0

However though clients insisted to have remote access to their ***MySQL Databases*** but since this is pretty

unsecure, **we decided not to configure MySQLs to listen to all available IP addresses / network**

interfaces.
MySQl acess is allowed only through **PhpMyAdmin** accessible via Cleint's Web interface which on some servers is **CPanel** and on other Kloxo (*This is open source CPanel like very nice webhosting platform*).

For some stubborn clients which wanted to have a mysql CLI and MySQL Desktop clients access to be able to easily analyze their databases with Desktop clients such as MySQL WorkBench there is a "hackers" like **work around to create and use a MySQL Tunnel to SQL server from their local Windows PCs** using standard OpenSSH Linux Client from Cygwin, MobaXterm which already comes with the SSH client pre-installed and has easy GUI interface to create SSH tunnels or eventually use Putty's Plink (Command Line Interface) to create the tunnel

Anyways the preferred and recommended (easiest) way to **achieve a tunnel between MySQL and local PC (nomatter whether Windows or Linux client system)** is to use standard **ssh** client and below command:

```
ssh -o ServerAliveInterval=10 -M -T -M -N -L 3308:localhost:3306 your-server.your-domain.com
```

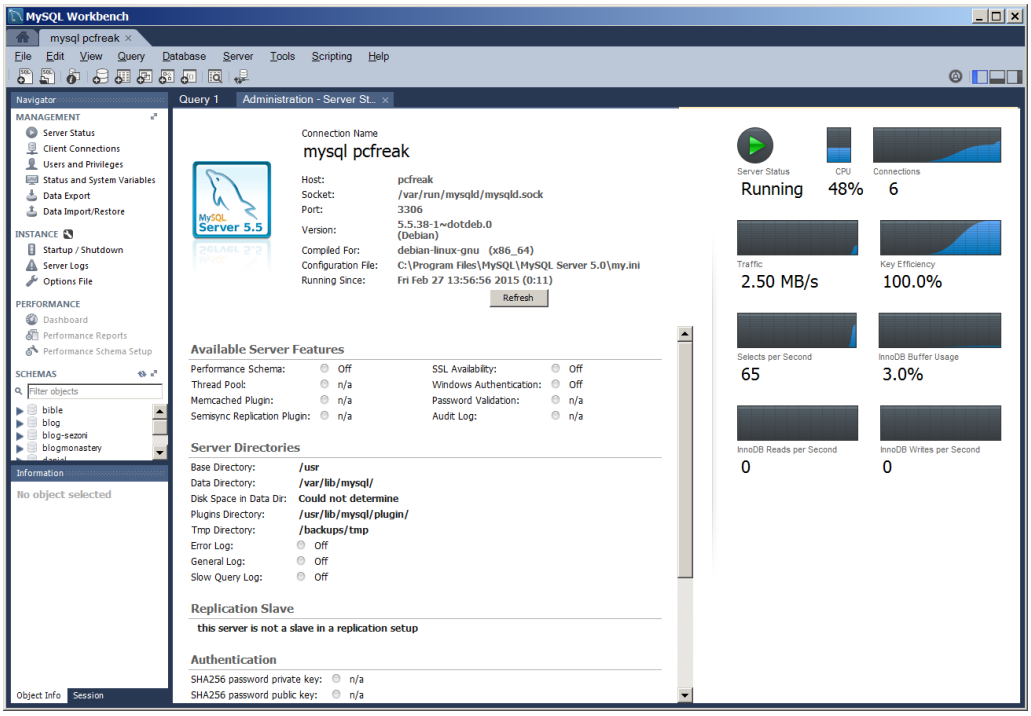
By default SSH tunnel will keep opened for 3 minutes and if not used it will automatically close to get around this issue, you might want to raise it to (lets say 15 minutes). To do so in home directory user has to add in**:**

```
~/.ssh/config

ServerAliveInterval 15
ServerAliveCountMax 4
```

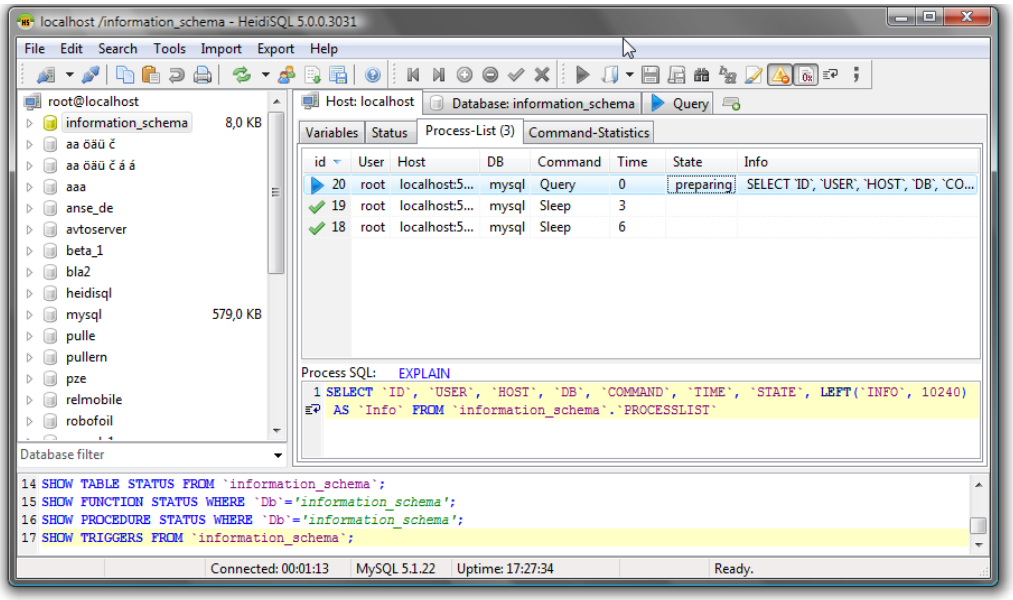
Note that sometimes it is possible ven though ssh tunnel timeout value is raised to not take affect if there is some NAT (Network Adress Translation) with low timeout setting on a firewall level. If you face constant SSH Tunnel timeouts you can use below **bash few lines code to auto-respawn SSH tunnel connection** (for Windows users use *MobaXterm* or install in advance *bash shell cygwin package*):

```
while true
do

ssh -o ServerAliveInterval=10 -M -T -M -N -L 3308:localhost:3306 your-server.your-

domain.com
  sleep 15
done
```

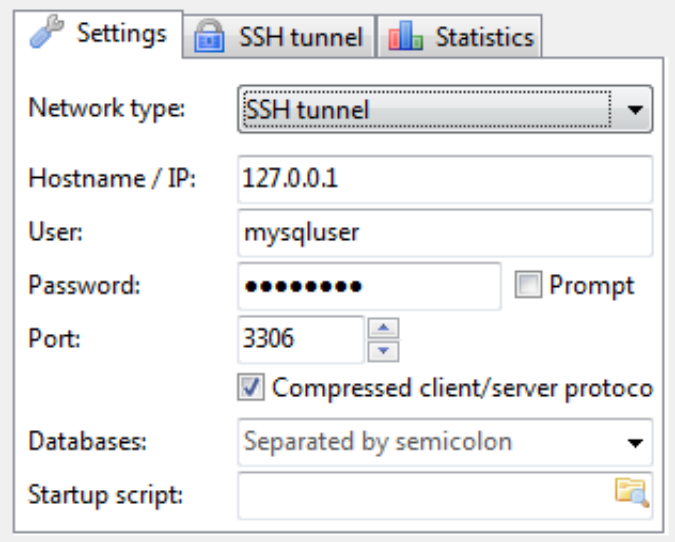
Below is MySQLBench screenshot connected through server where this blog is located after **establishing ssh tunnel to remote mysql server on port 3308 on localhost**



There is also another **alternative way to access remote firewall filtered mysql servers without running complex commands to Run a tunnel which we recommend for clients (sql developers / sql designers)** by using HeidiSQL (which is a useful tool for webdevelopers who has to deal with *MySQL and MSSQL hosted Dbs*).



To connect to remote MySQL server through a Tunnel using Heidi:



In the '**Settings**' tab

*1. In the dropdown list of '**Network type**', please select SSH tunnel*

*2. **Hostname/IP**: **localhost** (even you are connecting remotely)*

*3. **Username & Password**: your mysql user and password*

*Next, in the tab **SSH Tunnel**:*

*1. specify **plink.exe** or you need to download it and specify where it's located*

*2. **Host + port**: the remote IP of your SSH server(should be MySQL server as well), port 22 if you don't change anything*

*3. **Username & password**: SSH username (not MySQL user)*