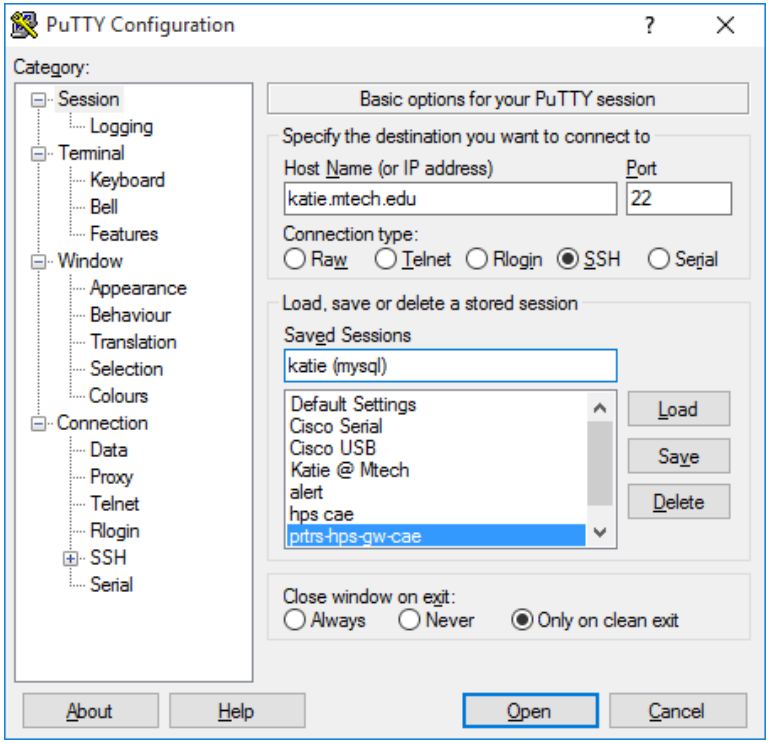
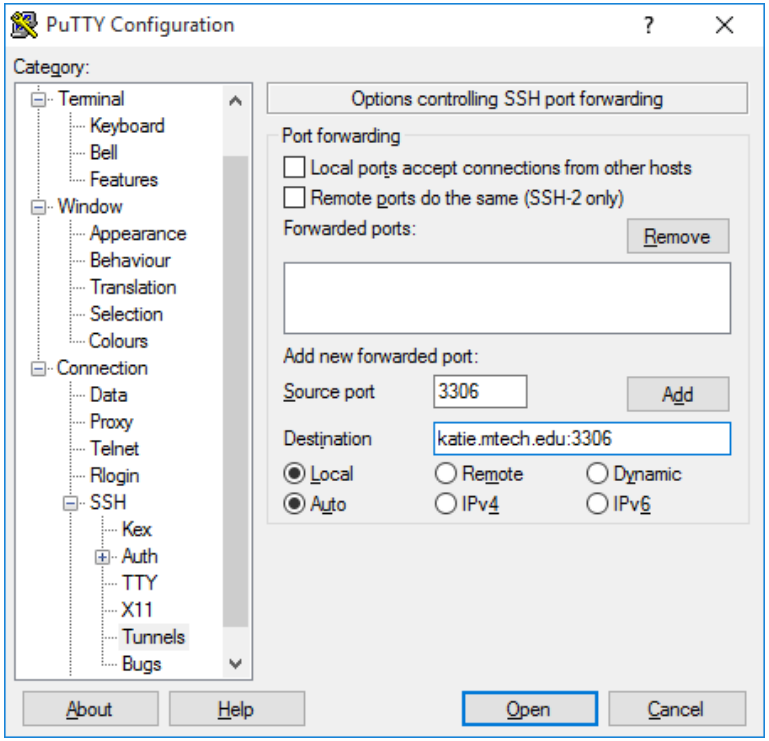
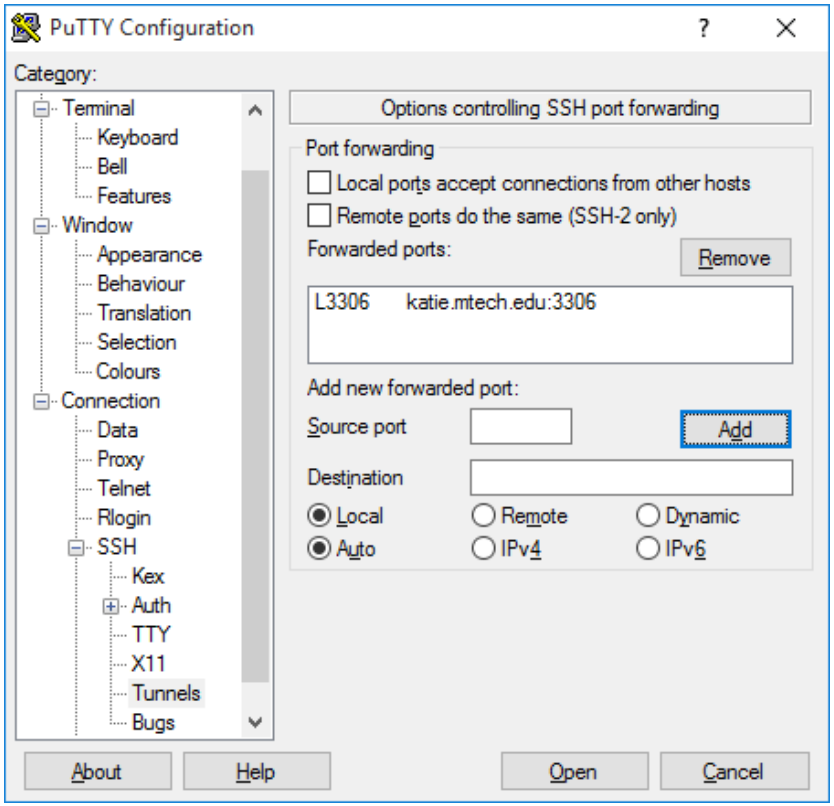
You must define a [putty session] that includes a tunnel specification, save that session, then, when you want to run your C# application, you must first login to Katie using the save session that has the tunnel configuration and remain logged into Katie the entire time you are using the C# application. If you logout of Katie, you tear down your ssh session to Katie, and the tunnels it establishes.



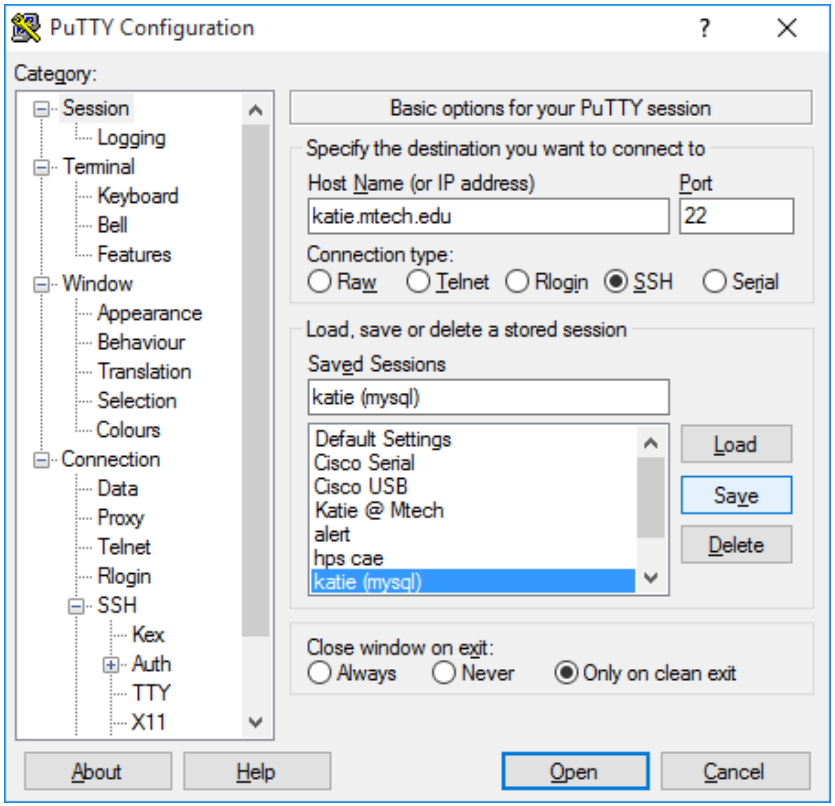
Above you see a new session being defined and named “Katie (mysql)” to differentiate it from any other Katie ssh sessions you might have saved.



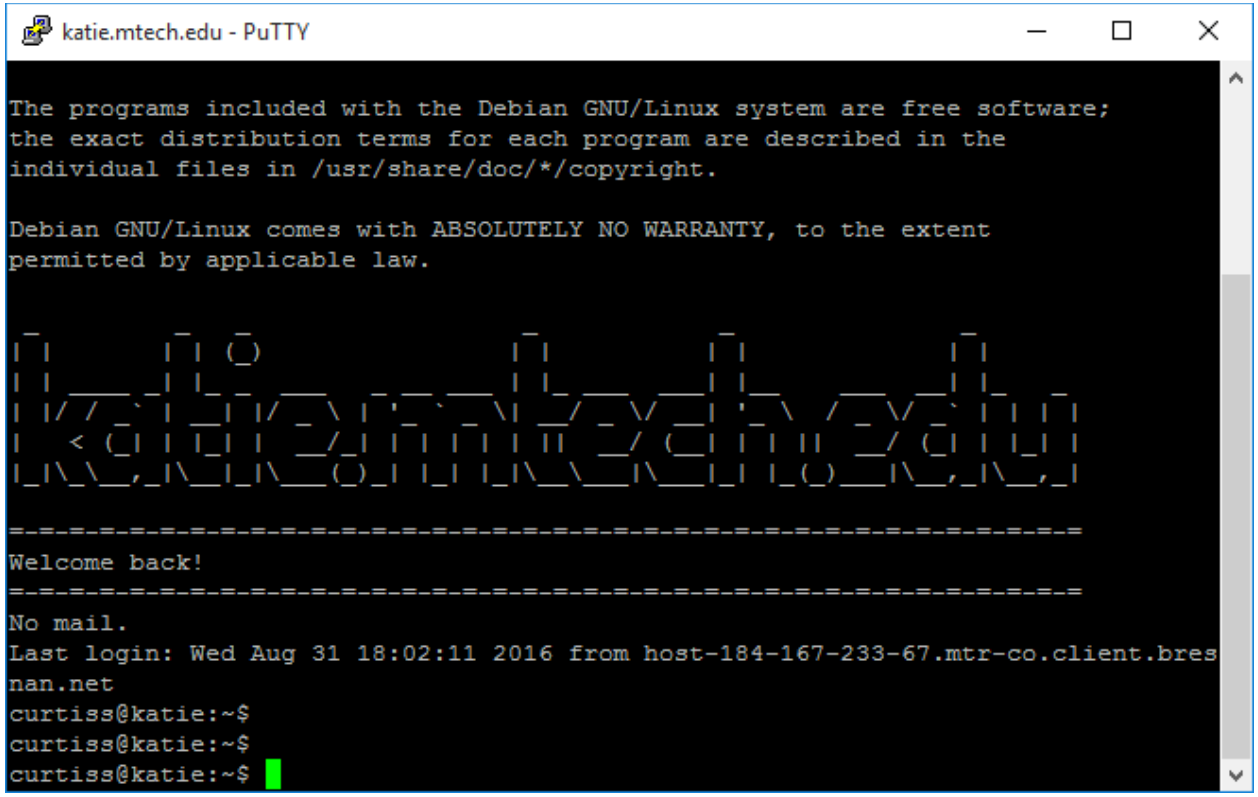
Above you see the [Connection->SSH->Tunnels] panel that shows what you would put in the Source Port and Destination fields. Note the destination has the hostname followed by a colon followed by the port number, all without any spaces. You would then click the [Add] button to store the tunnel specification.



Above you see what the tunnel specification looks like following you clicking the [Add] button.

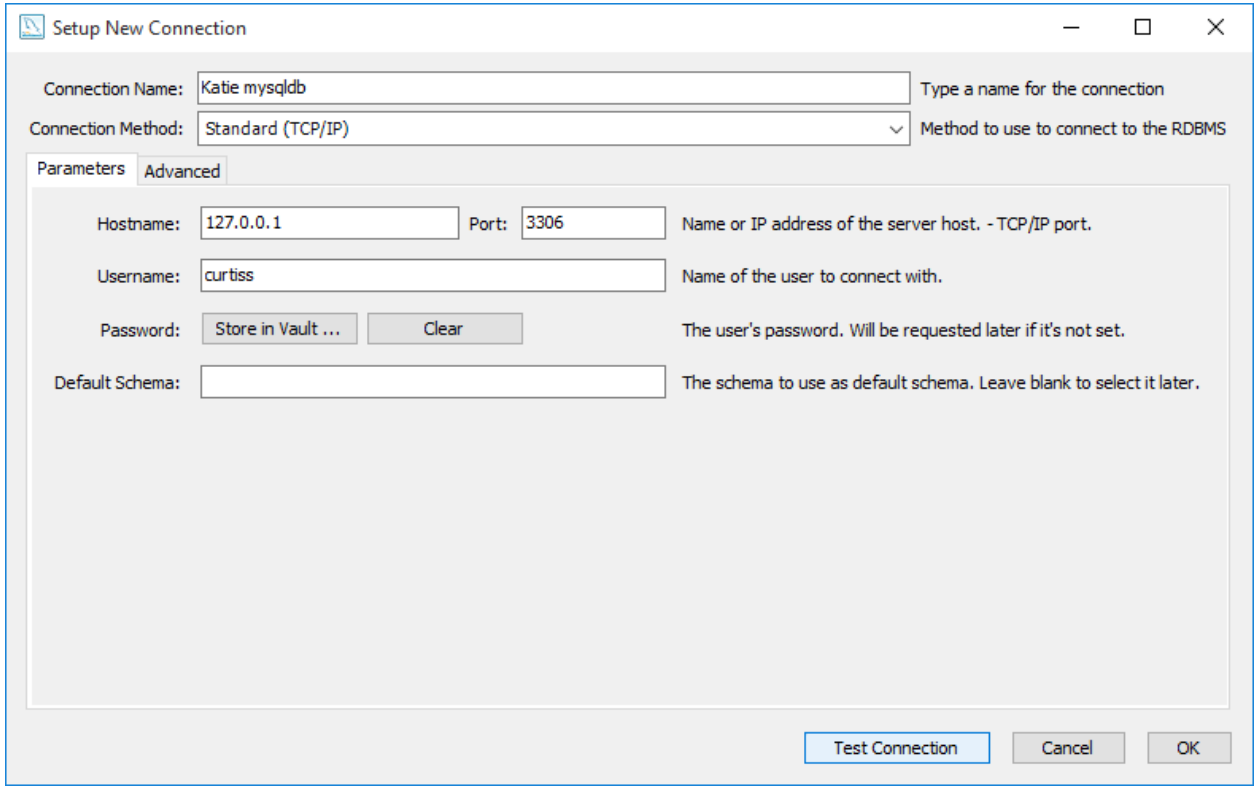


Above shows you clicking the [Session] link in the left panel, and then clicking the [Save] button to save the SSH session.



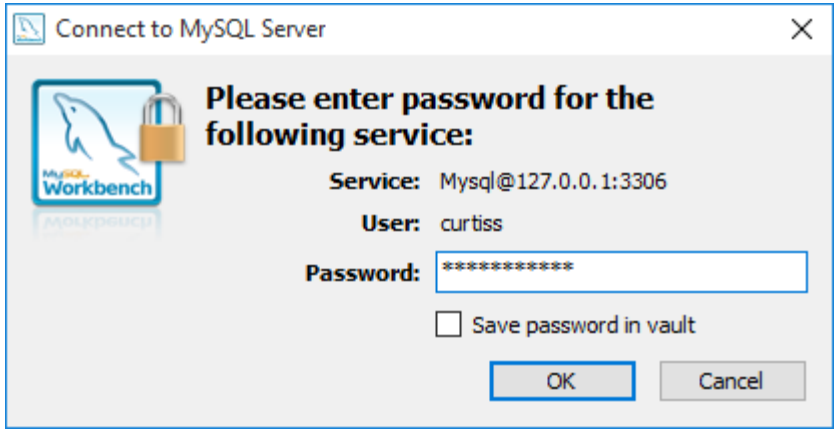
Above you will see the result of clicking the [open] button with the new saved session “Kaite (mysql)” highlighted. You will login to Katie, like normal, except behind the scenes, there is now an ssh-based tunnel for the 3306 port from your local machine (on which putty is running) and port 3306 on Katie (on which mysql server is running).

This tunnel will persist until you logoff Katie. At that point, the tunnel will be torn down and you will no longer have port 3306 access to mysql server running on Katie

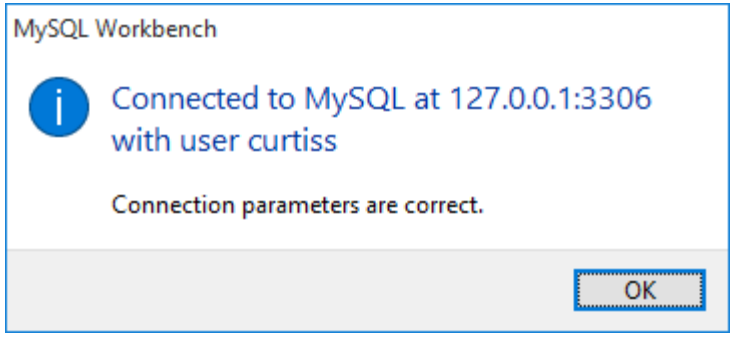


To show you this works, I launched MySql Workbench and selected the [New Connection] link. Here you see that I have named the Connection "Katie mysqldb" and specified the hostname as my localhost IP address 127.0.0.1 and the default (well-known) port of 3306. I then provided my username (for mysql server).

When I press [Test Connection] mysql workbench will try to connect to a mysql server running on the local machine (the one running mysql workbench – my laptop here at home) on port 3306. That connection will succeed as Putty has created that port on my local machine (my laptop) and is listening for traffic. When it obtains traffic, it will forward that traffic through the ssh tunnel it established to Katie.mtech.edu on its port 3306. On port 3306 on Katie, there is in fact a mysql database running, and so my connection will succeed.



Above shows the success of me authenticating (not to Katie, but) to mysql server running on Katie.mtech.edu through the ssh tunnel.



Your C# program will operate the same way. If that C# program attempts to connect to 127.0.0.1 (the localhost IP address) on port 3306, AFTER you have established the ssh session with the tunnel, it will connect to the mysql server running on Katie.