

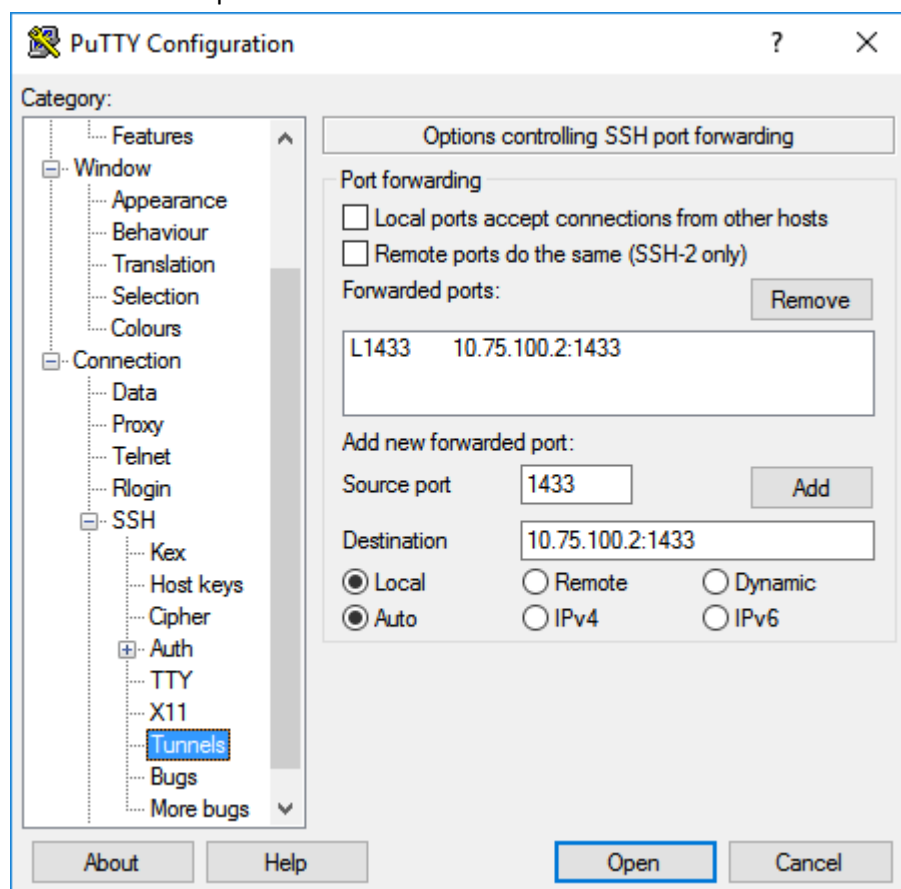
# Technical Notes: Connecting to SQL Server from Off Campus

## Overview

- Use putty or command-line ssh to set up port-forwarding from port 1433 on your machine to port 1433 on the IT Department's SQL Server instance.
- Establish an ssh connection to kate.ict.op.ac.nz. Establishing the connection will initiate the port forwarding.
- Connect to SQL Server by pointing SSMS at 127.0.0.1, but otherwise logging on as normal.

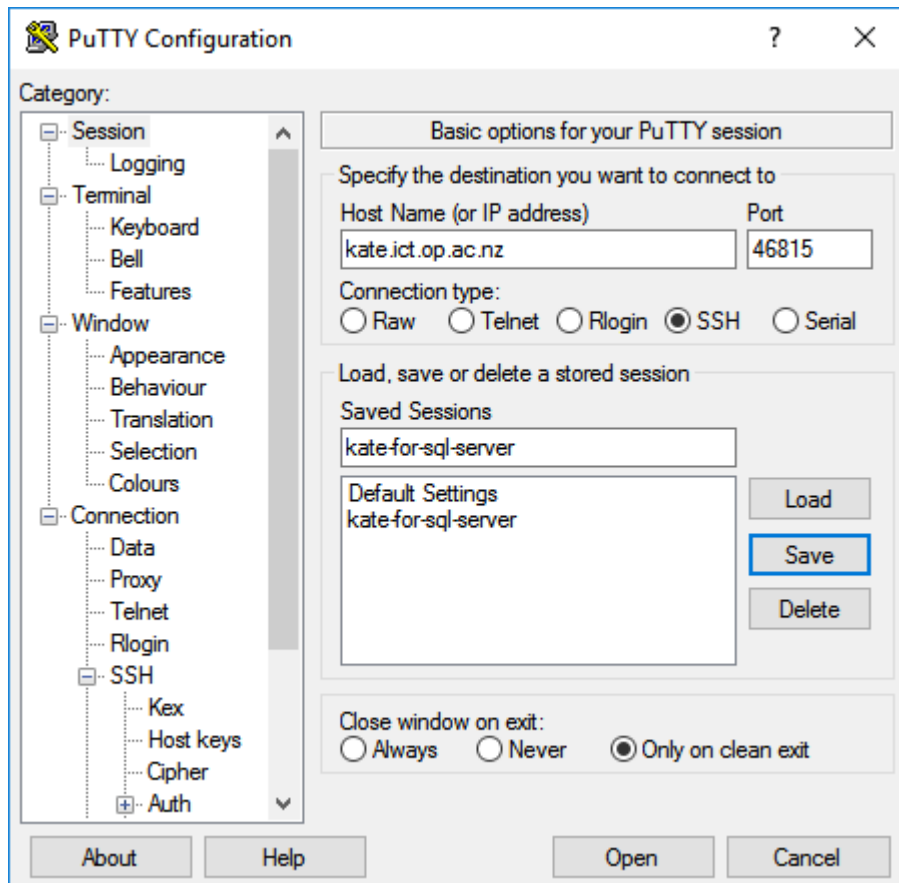
## Details

- Open up PuTTY's configuration pane and go to Connection->SSH->Tunnels
- Enter 1433 into the Source port field
- In Destination, enter 10.75.100.2:1433 (this is our SQL Server instance)
- Now click the add button beside the Source port field
- You should end up with this:

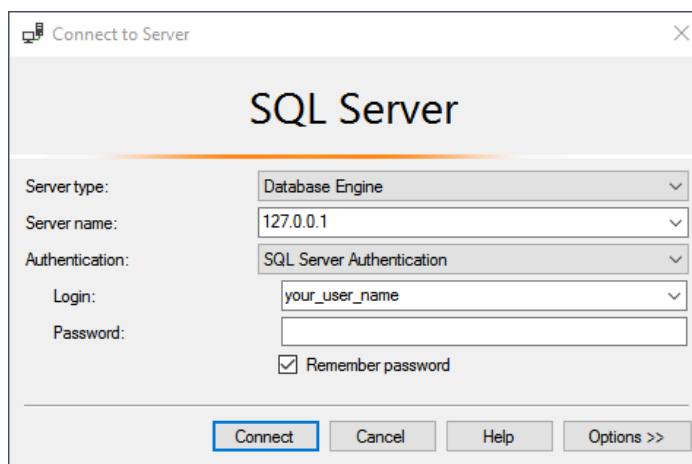


- Don't click Open yet!
- In the left pane, scroll back up to Session
- Under Host Name, enter kate.ict.op.ac.nz, Port 46815
- It's worth saving your session now: give it a name under Saved Sessions and click Save

- Things should look like this:



- Click the Open button and log in to kate normally.
- Fire up SSMS
- Enter 127.0.0.1 as the server name. User SQL Server Authentication.



- You should be able to connect to your database normally at this point.

## What just happened?

Port 1433 is the port that SQL Server listens on to accept queries. It's also the port that SMSS uses to send out queries. By forwarding 1433 on your machine to 1433 on the server, you're achieving two things: 1) Applications on your machine (e.g. SMSS) think that port 1433 on local host is behaving just like the server that they would normally talk to, and 2) the server thinks it's talking to kate – as far as the server is concerned, it's getting requests from kate, which is on the campus network and thus safe to talk to.

Extra note/reminder: we're using port 46815 on kate rather than the standard port 22 for ssh connections.