

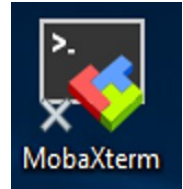
## Accessing Research systems remotely from Windows systems

Unlike the department's teaching Linux farm, which can be accessed directly from home machines (desktop or laptop), connecting to research systems on the departmental internal network is a two-stage process, using a *Secure Shell* connection (SSH). Windows systems usually require an SSH client to be explicitly installed (as this is not included as part of a typical installation).

The software that we recommend is called **MobaXterm**, and this can be obtained from <https://mobaxterm.mobatek.net/download-home-edition.html>

Download the Installer edition zip archive, open it, and run the MSI installer found there. Then follow the Setup Wizard as usual.

This will typically install an icon on your desktop (as well as an entry in the Start Menu)



### Increased security measures

Unfortunately, following an increase in security exploits targetting universities, the departmental research systems are now only accessible if you first set up a VPN connection via the University's Cisco AnyConnect system. This needs to be done before attempting to connect to the SSH gateway systems described below. See <https://www.liverpool.ac.uk/csd/vpn/> for more details. Note that this will also trigger a Duo 2-Factor Authentication challenge.

Alternative it is possible to set up a *departmental VPN connection* to the departmental internal network, which avoids the need to go via the SSH gateway systems altogether. This approach is documented separately – see the document [VPN \(Windows\)](#) in the section [Guides](#) on the departmental intranet server for more details.

### Configuring MobaXterm

The first time you use MobaXterm, you will need to supply details of the connection to set up. This information will then be saved, and can be used again for subsequent connections. MobaXterm supports a variety of different types of session – in this document we will only consider a simple SSH connection via the departmental SSH gateways.

There are two such systems – named `ssh1.csc.liv.ac.uk` and `ssh2.csc.liv.ac.uk`.

The available software and configuration of these two systems are slightly different – however their primary purpose is simply to provide a stepping stone to other systems within the department. They can handle simple processing tasks, but anything more demanding should be run on more powerful or specialist equipment. If either machine is not available, please use the other one (and inform the technical staff of the problem).

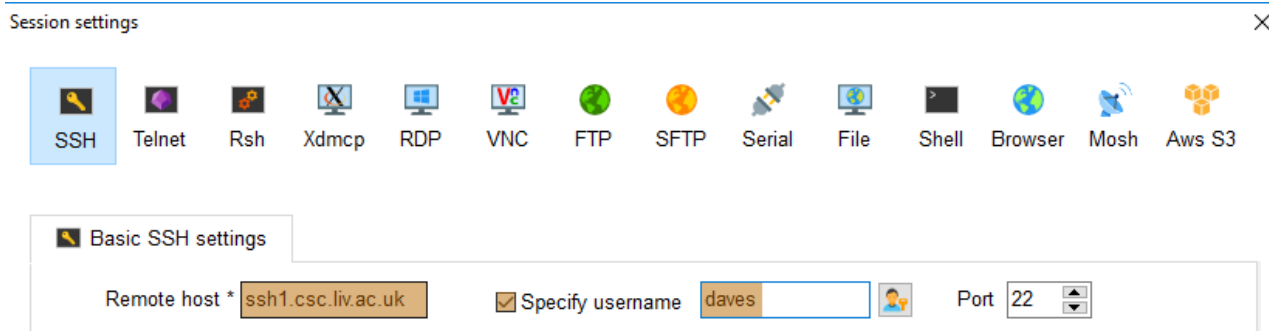


Click the *Session* icon (*left*) in the buttons bar across the top of the window. Then click the *SSH* session button (*right*) and fill in the fields as follows:

Remote Host      `ssh{X}.csc.liv.ac.uk`  
Check the box *Specify username* and enter your **departmental** username



The rest of the settings (*Advanced*, *Terminal*, etc) can be left at the default values.

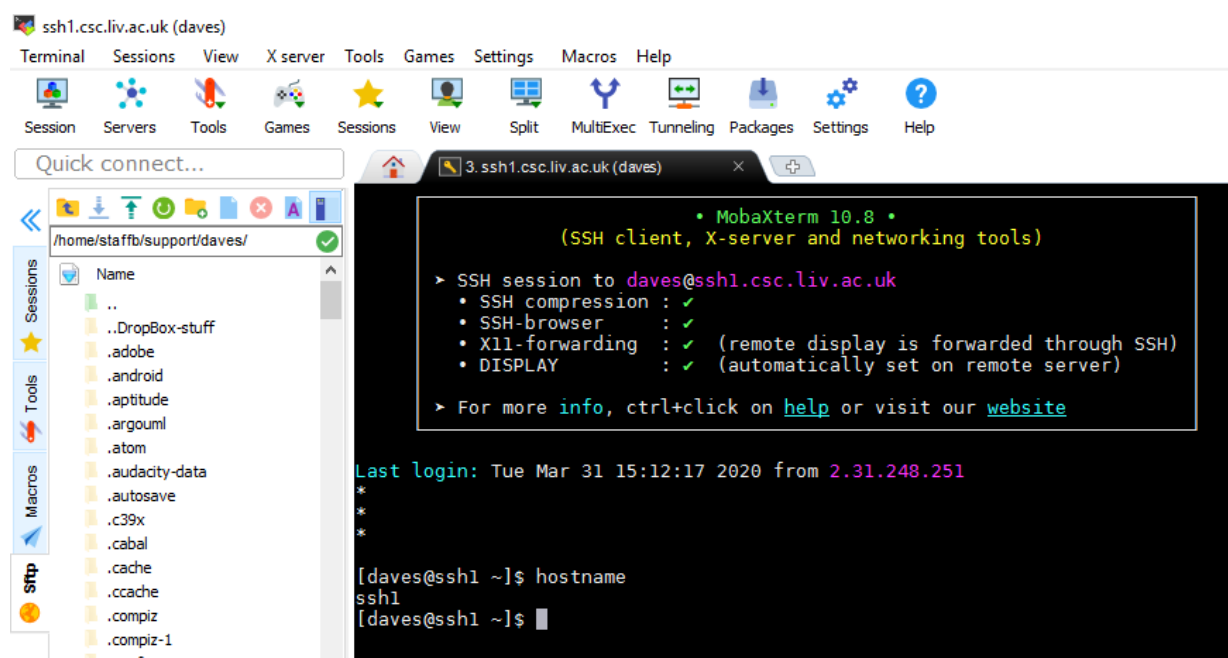


Then click OK

You will be asked to supply your **departmental** password (and press Return).



This will log you into the chosen system, and you can run simple Linux commands on that machine. For text-based commands, the output will be shown in the same **MobaXterm** terminal window – graphical commands should open up a new window on your local display.



Note that immediately after logging in, you will be informed of the time and source (IP address or hostname) of the last successful login session. If you do not recognise this connection, please inform the CS technical team immediately by emailing [CSC-HELPDESK@liverpool.ac.uk](mailto:CSC-HELPDESK@liverpool.ac.uk). You may also wish to change your password.

When reporting this, or any other issue with the system, it is extremely helpful to include details of which Linux machine you were using (which should now appear as part of the command prompt), and the date & time any problem occurred. If you're not sure which system you're on, the command `hostname` will also show this information (see above).

Having connected to the SSH gateway machine, the next step is to make a second *SSH* connection to the target system. This will commonly be a personal desktop machine (typically running Linux or MacOS), or a dedicated research group server – see your research group lead or supervisor for more information about any such resources available to you.

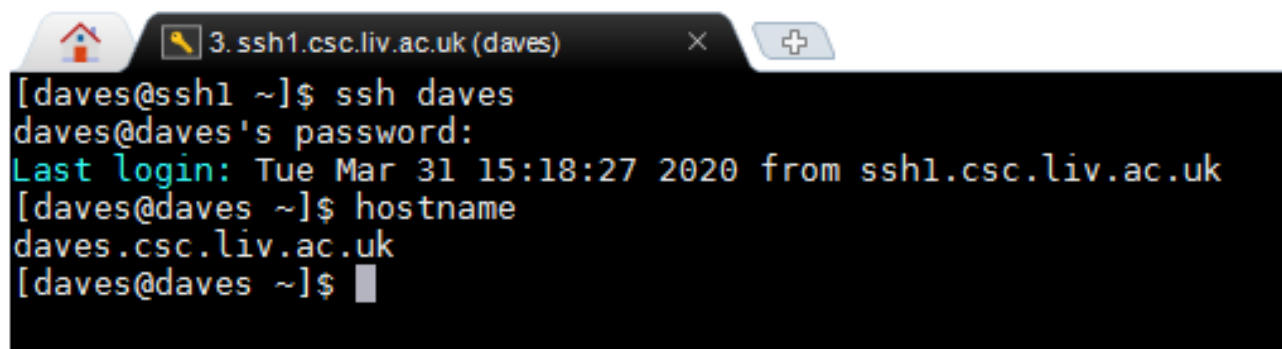
The department also has a number of general-purpose Linux systems available for staff/PhD use. These include:

O/S	Host names	
Scientific Linux	lxfarm01 .. lxfarm08	<i>RedHat-based enterprise O/S – currently v7.9 Student teaching Linux provision</i>
CentOS	centos01 .. centos04	<i>RedHat-based enterprise O/S – currently v8.4</i>
Fedora	fedora01 .. fedora08	<i>RedHat-based development OS - updated annually Currently Fedora 32 (01..04) and Fedora 34 (05..08)</i>
Ubuntu	ubuntu01 .. ubuntu08	<i>Debian-based LTS release – updated biennially Currently v18.04 (01..04) and v20.04 (05..08)</i>

Connect to the selected machine using the command: `ssh {host}`

Note that there is no need to specify an explicit username (unless this is on the target system).

If graphical commands will need to be run (with the output appearing on your local display), then use the command: `ssh -X {host}`

A screenshot of a terminal window titled '3. ssh1.csc.liv.ac.uk (daves)'. The terminal shows a user named 'daves' logging in via SSH. The prompt is '[daves@ssh1 ~]\$'. The user enters 'ssh daves', and the prompt changes to 'daves@daves's password:'. The user then enters their password, and the terminal shows 'Last login: Tue Mar 31 15:18:27 2020 from ssh1.csc.liv.ac.uk'. The prompt changes to '[daves@daves ~]\$'. The user then enters 'hostname', and the terminal shows 'daves.csc.liv.ac.uk'. The prompt changes to '[daves@daves ~]\$' with a cursor at the end.

When you have finished with the session, use the command `exit` to disconnect from the internal system, and return you to the SSH gateway machine. Then type `exit` again to disconnect from that, and return you to the local machine. You can then close the **Terminal** window, as usual.

Note that closing the **Terminal** window *without* first `exit`ing cleanly may leave processes running on the internal and/or gateway machines, and could cause problems in the future.