

Logging in with SSH

On this page:

- [Pawsey Hostnames](#)
- [Login nodes, compute nodes and data movers](#)
- [Logging in using a computer with Linux or macOS operating system](#)
- [Logging in using Visual Studio Code for Linux, macOS and Windows](#)
 - [Recommended settings in Visual Studio Code](#)
- [Alternative programs to log in from Windows](#)
 - [MobaXterm](#)
 - [Recommended settings in MobaXterm](#)
 - [Basic connection \(using the "local terminal" option\)](#)
 - [MobaXterm local terminal](#)
 - [Common Problems](#)
 - [VirtualBox](#)
 - [Putty](#)
 - [Cygwin](#)
- [Logging in with graphical interaction](#)

Interaction with Pawsey supercomputers is done remotely using SSH (Secure Shell version 2, SSH-2). This is an encrypted communication channel that allows command-line access to any of our computers. After communication is established with the login node, you can run commands or use a command-line text editor to edit files. SSH can also be used to "tunnel" GUI programs such as GUI text editors and debuggers. The general use of ssh is:

```
ssh [-options] username@hostname
```

Where the hostname can be any of the addresses listed in the following subsection. In order to authenticate the connection, there are two possibilities: (1) type the password in every connection, or (2) to use a ssh-key which is a secure method to avoid the need of typing the password every time. SSH-keys are also very useful to establish connections in automated scripts which, for example, can be used for file transfers. In this current page we will explain the general usage of SSH and authenticate by typing a password. The use of SSH-keys is explained separately in [Logging in with SSH keys](#).

Pawsey Hostnames

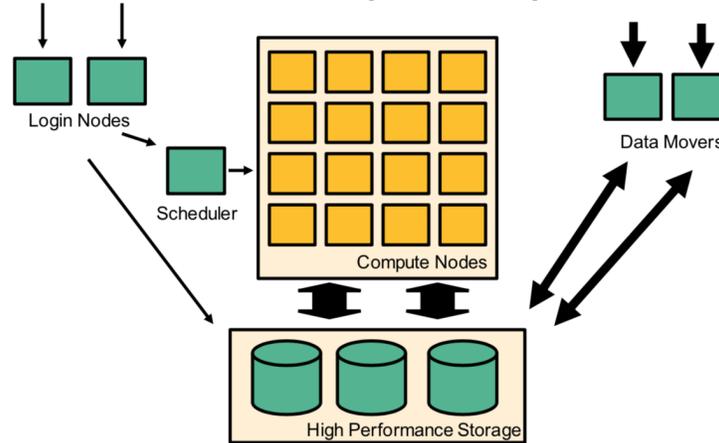
Generally, the hostname is the name of the system (in lowercase) appended by ".pawsey.org.au":

System	Hostname
Magnus	magnus.pawsey.org.au
Zeus	zeus.pawsey.org.au
Galaxy	galaxy.pawsey.org.au
Topaz	topaz.pawsey.org.au
Data mover	hpc-data.pawsey.org.au

Login nodes, compute nodes and data movers

When a remote connection is established with any of our systems, it is to a login node. Login nodes are like the "front desk" of the system, and they are where users manage their workflows, edit files, and submit jobs to the scheduler in order to be executed in the compute nodes. The compute nodes are where the main computations are processed and they can be accessed through the different queues or partitions managed by the scheduler. These concepts are explained in the following schematic:

Abstract Supercomputer



The hostnames of the systems listed in the previous subsection are often a pseudonym for the several login nodes that a system may have. When connecting remotely, what actually happens is that connections are randomly placed on a login node from where you interact with the rest of the system. The real name of the login node ("magnus-2" in the example below) can be easily obtained by executing the Linux command "hostname" after the ssh connection has been established:

```
localComputer:~> ssh mickeymouse@magnus.pawsey.org.au
Password:
Last login: Tue Jun 12 11:07:13 2018 from 130.116.145.55
#####
#           Pawsey Supercomputing Centre           #
#   Empowering cutting-edge research for Australia's future   #
#   #
#   This service is for authorised clients only.             #
#   It is a criminal offence to:                          #
#     - Obtain access to data without permission             #
#     - Damage, delete, alter or insert data without permission #
#   #
#####
.
.
.
=====
By using Pawsey facilities you agree to the Conditions of use available at
https://support.pawsey.org.au/documentation/display/US/Conditions+of+Use
=====
mickeymouse@magnus-2:~> hostname
magnus-2
```



IMPORTANT:

Many users share the login nodes at the same time. Therefore, OPERATIONS THAT GENERATE HEAVY LOADS ON LOGIN NODES SHOULD BE AVOIDED. You should NOT run programs or build/compile tools on login nodes if these operations would take more than a few minutes. Heavy workloads should be sent to the compute nodes.

Also, DO NOT TRANSFER, COPY OR DELETE LARGE AMOUNTS OF FILES/DATA THROUGH THE LOGIN NODES. Heavy file operations should be performed through the data mover system: [hpc-data.pawsey.org.au](https://support.pawsey.org.au/documentation/display/US/Conditions+of+Use) (whose nodes are explicitly devoted for that), or submitted to the "copyq", which uses the nodes in the data mover system.

Use of the login nodes that may have adverse effects on the service may be dealt with under the [Conditions of Use](#).

Logging in using a computer with Linux or macOS operating system

All Linux and MacOS distributions come installed with a terminal application that can be used for SSH access to the login nodes. Linux users will have different terminals available depending on their distribution and window manager (e.g. GNOME Terminal in GNOME, Konsole in KDE). Consult your Linux distribution's documentation for details on how to load a terminal.

MacOS users can use the "Terminal" application, located in the Utilities folder within the Applications folder. "iTerm2" (which needs to be installed separately) is also very popular.

Once a terminal window has been chosen and opened in the local system, execute the ssh command. In the following example username "mickeymouse" is connecting into Zeus:

```
localComputer:~> ssh mickeymouse@zeus.pawsey.org.au
Password:
Last login: Tue Jun 12 10:00:20 2018 from 130.116.145.55
#####
#                                     #
#           Pawsey Supercomputing Centre           #
#           Empowering cutting-edge research for Australia's future           #
#                                     #
#           This service is for authorised clients only.           #
#           It is a criminal offence to:           #
#           - Obtain access to data without permission           #
#           - Damage, delete, alter or insert data without permission           #
#                                     #
#####
.
.
.
=====
mickeymouse@zeus-1:~>
```

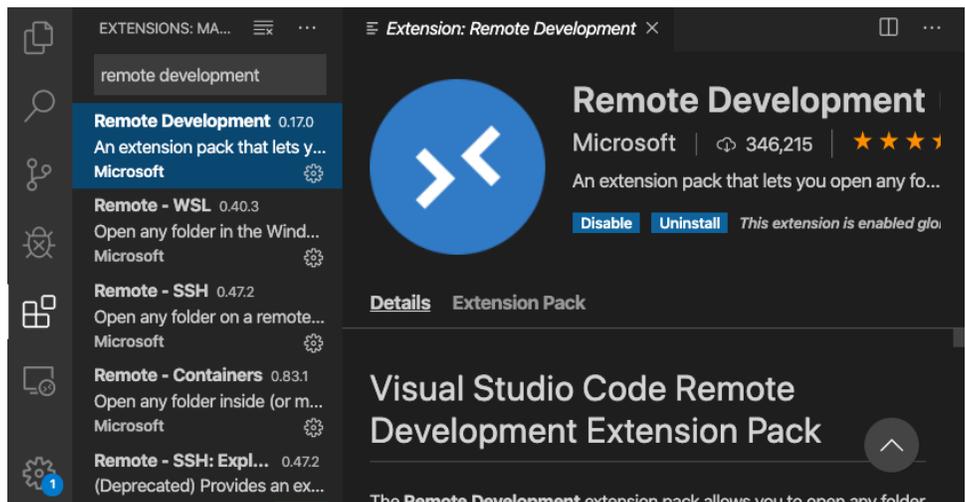
Logging in using Visual Studio Code for Linux, macOS and Windows

Visual Studio Code is a free and open source code editing software that can be deployed on Linux, macOS and Windows. It has an integrated terminal within its user interface that removes the need to switch between command-line tasks and code editing. The default terminal shell is bash on Linux and macOS, and PowerShell on Windows. *For Windows users to configure the default terminal shell to bash, an installation of Git from <https://git-scm.com/downloads> is required before installation of Visual Studio Code.

You can download Visual Studio Code from the following link: <https://code.visualstudio.com>.

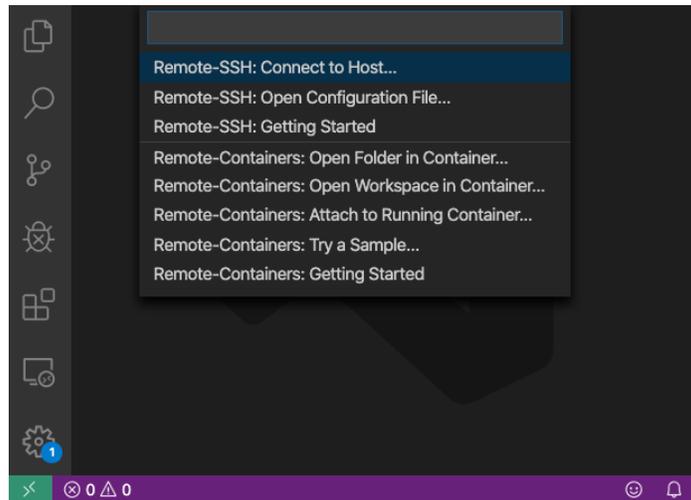
Recommended settings in Visual Studio Code

Although the integrated terminal on visual studio code can be used directly to SSH log in to Pawsey systems, a useful feature of Visual Studio Code is the Remote Development extension pack that is easily installed from the Marketplace within the application. It allows the ability to open remote directories and text files on Visual Studio Code for in-app code editing and building.

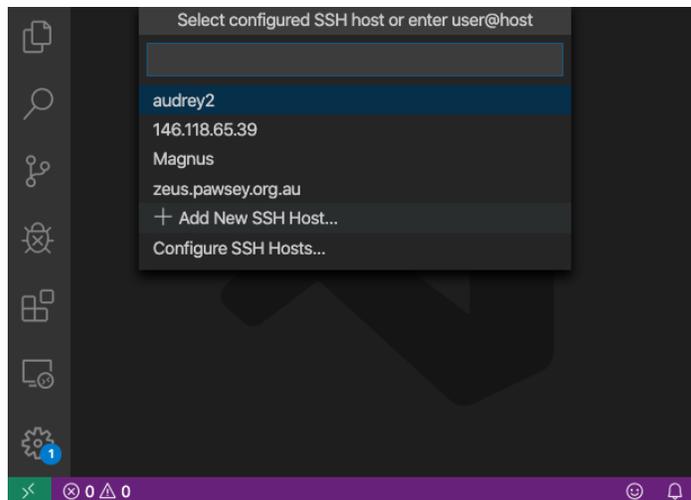


Once installed, connect to Pawsey systems as follows:

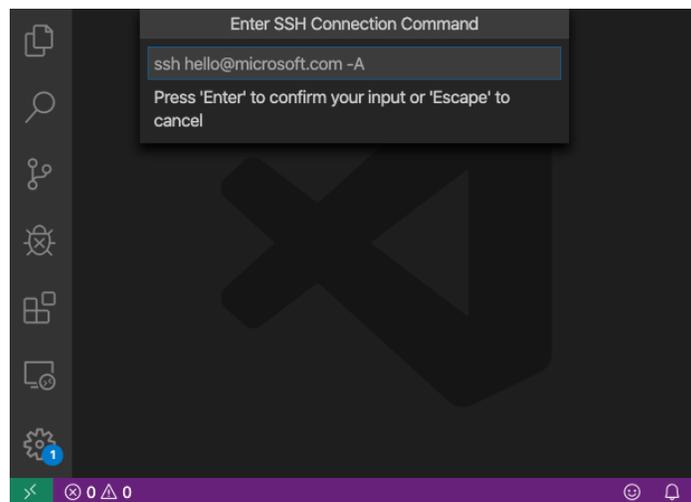
- Click on the green button in the bottom left corner of the screen and select the first option "Remote-SSH: Connect to Host..."



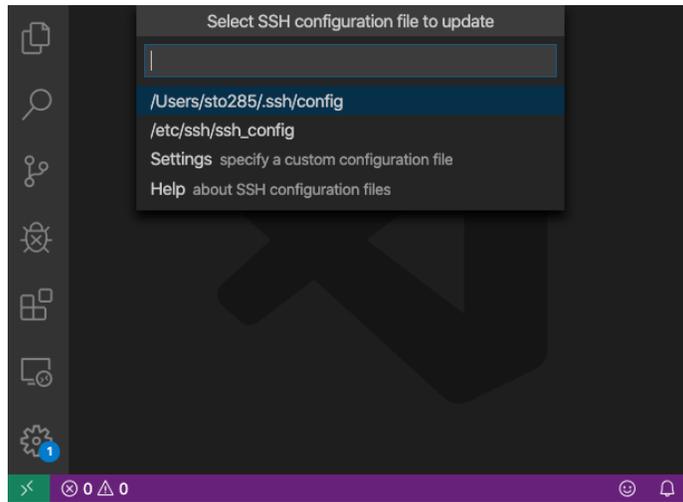
- Select "+ Add New SSH Host..."



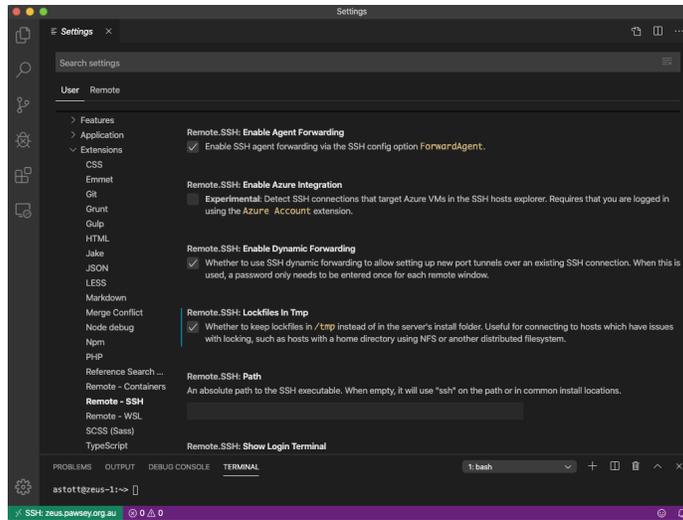
- Log in using your Pawsey username, e.g. ssh yourusername@zeus.pawsey.org.au



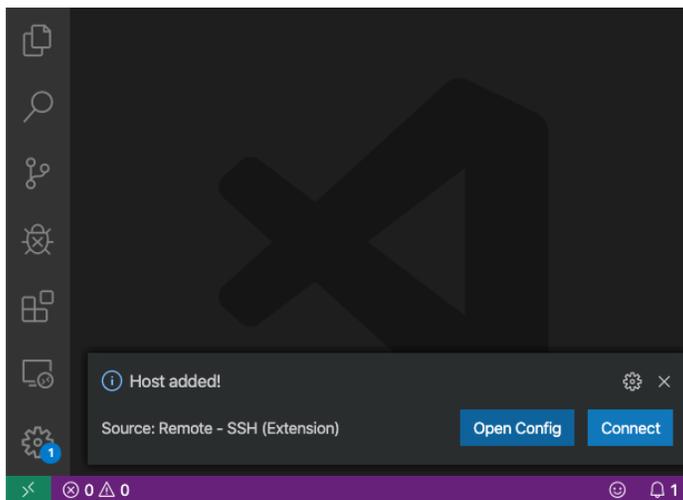
- Save the host details in the default config file, e.g. `/Users/yourcomputername/.ssh/config`

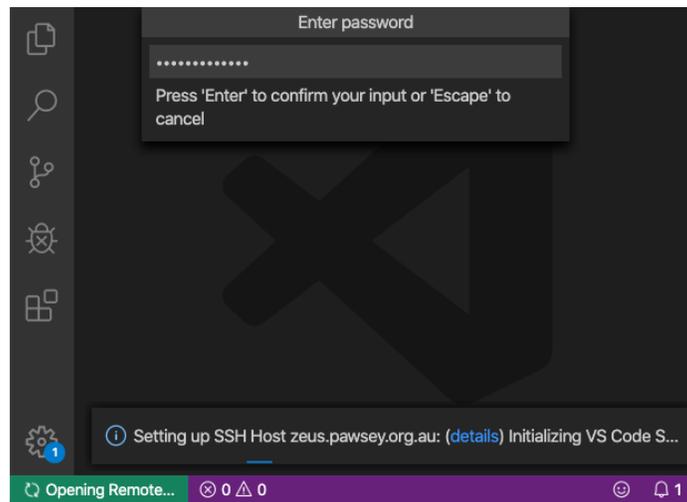


- You may also have to check the box for "Remote.SSH: Lockfiles in Tmp", under settings, in order to connect to Pawsey systems.

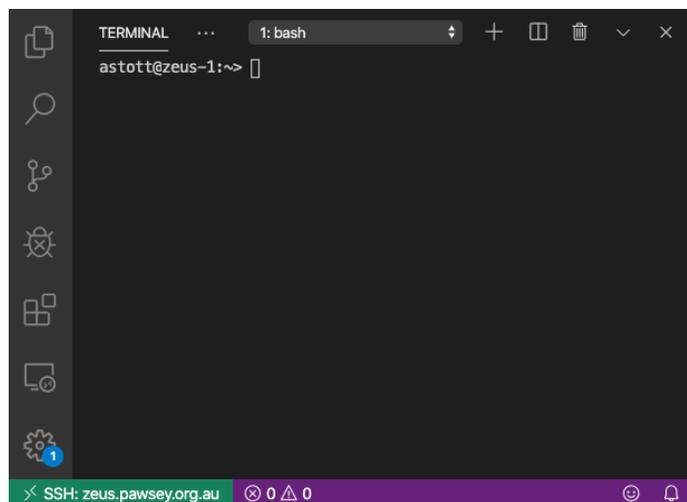


- Once the host details is saved, you can click on "Connect", a prompt appears for you to enter your Pawsey password.





- Finally a new window with the integrated terminal will open and you will see in the bottom left corner that you are connected to the remote host.



- Note: To use code editing interactively, use the command "code" to open and edit text files from your remote directories, e.g. typing the command code sample.txt will open the <sample.txt> file within Visual Studio Code.

Alternative programs to log in from Windows

MobaXterm

Another popular alternative client for Windows users is MobaXterm, which includes an integrated X server with SSH client to run any graphical applications on supercomputers.

You can download MobaXterm Home Edition from the following link: <http://mobaxterm.mobatek.net/download-home-edition.html> .

If you do not have administrative (root) rights on your Windows machine, go for the "**portable edition**".

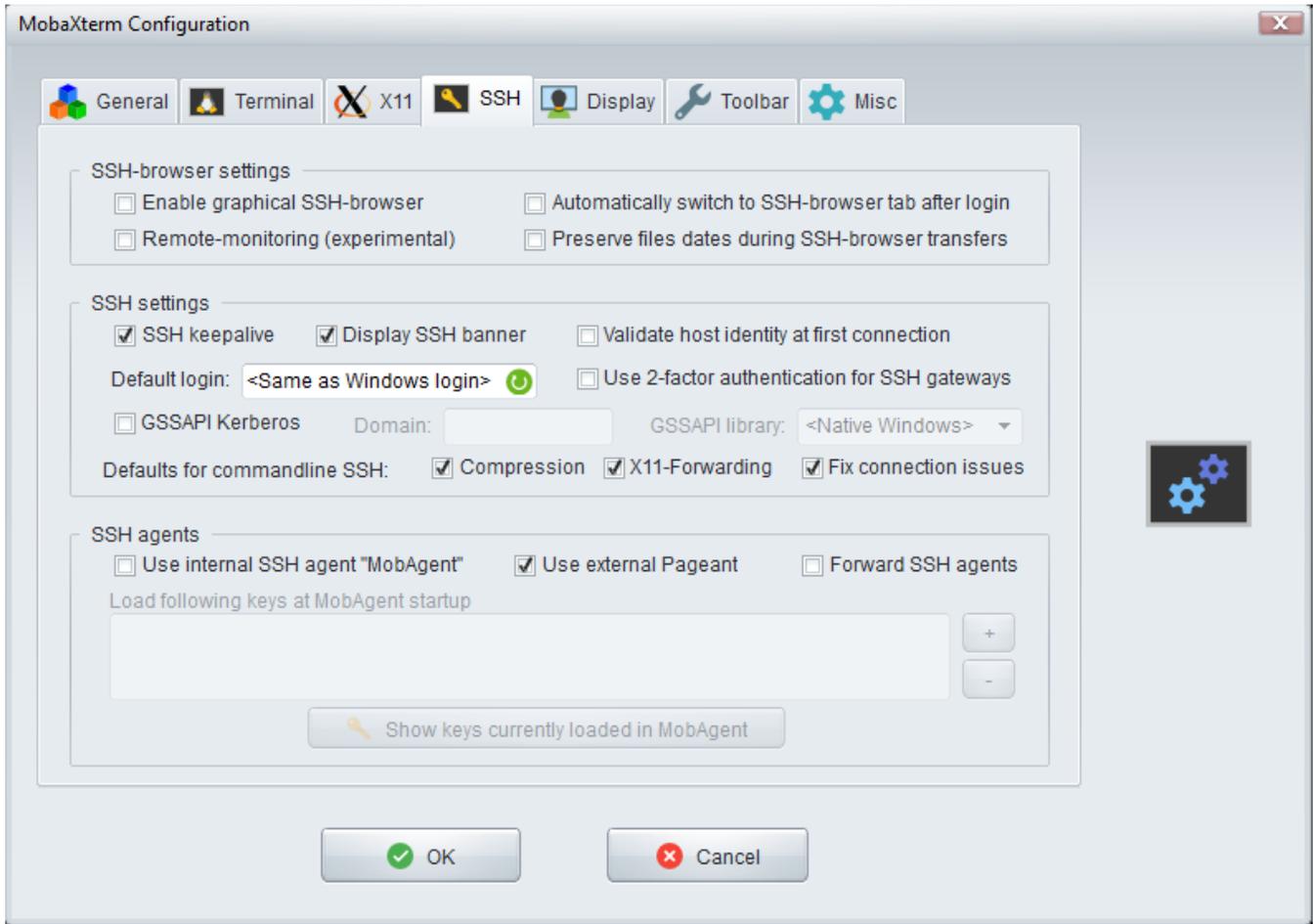
If you do have admin rights, you can eventually choose the "installer edition" (recommended). Double-click the downloaded Microsoft Installer file (.msi), and the Windows wizard will automatically guide you through the installation process. Also make sure to check whether Windows Firewall hasn't blocked any features of this program after installation.

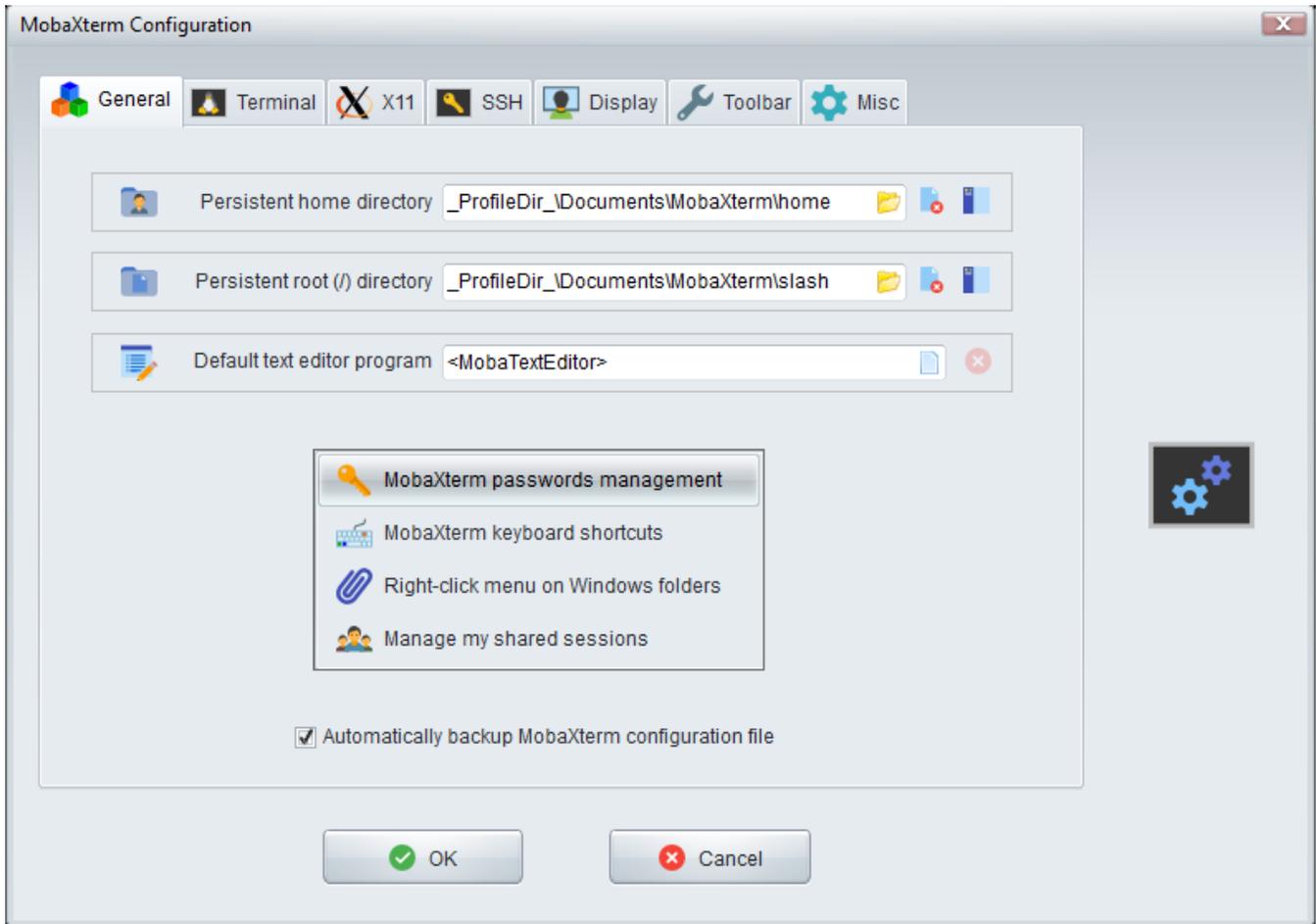
Recommended settings in MobaXterm

In the SSH Settings, we recommend:

- Turn off the graphical SSH-browser (sftp client), which otherwise creates an unnecessary connection to the remote host. You should be using hpc-data.pawsey.org.au to transfer data with Pawsey systems, and there are better graphical interfaces such as Filezilla. See [Transferring files](#) for more information.
- Leave X11-Forwarding enabled. Then you don't need to "ssh -X".

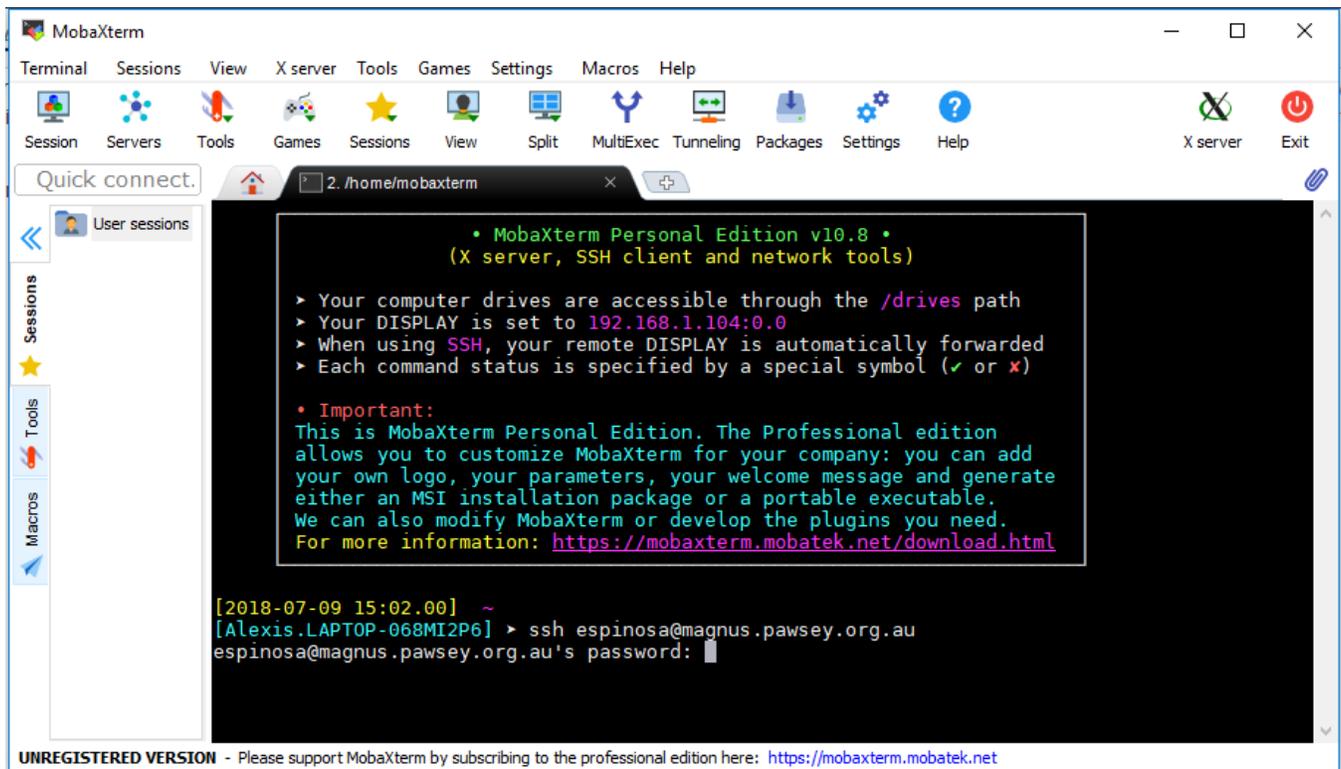
In the General Settings of MobaXterm, change `_ProfileDir_\Documents\MobaXterm` to `_AppDataDir_\MobaXterm` as the base for the directories where MobaXterm keeps its settings etc. This avoids filling up the Documents library with small files, which will pollute Windows Search.





Basic connection (using the "local terminal" option)

Start MobaXterm using, for example, the icon added to the Start menu during the installation process. Then click on "Start local terminal" option that should appear exactly in the middle of the MobaXterm window. From your local terminal you can use ssh command (as explained in the topics above), for example:



For small remote GUI applications, use `ssh -X` if X11 forwarding is turned off in the MobaXterm settings. However, we recommend using the Pawsey [Remote Visualisation](#) for large GUI applications such as ParaView or VisIt.

MobaXterm local terminal

This is a practice we recommend...

To be able to start a local terminal (since version 8), you need to download and install the CygUtils.plugin package from <http://mobaxterm.mobatek.net/plugins.html>. Move the downloaded file into the same directory as the MobaXterm executable, which is probably "C:\Program Files (x86)\Mobatek\MobaXterm Personal Edition", and then restart MobaXterm.

Common Problems

"Server unexpectedly closed network connection" is usually due to the system denying you access as a result of too many failed connection attempts from your IP address. This is a security feature we have implemented to mitigate against brute force password attacks, and the system thinks that your multiple failed logins (perhaps you typed your password wrong a few times) is such an attack. When this happens, use the [Helpdesk Service](#) and provide your username and when you could not log in, and we can work out what IP address you tried and unblock it.

"fork: retry: Resource temporarily unavailable" is due to you having too many processes running on the login nodes. Pawsey has limited the number of processes for each username because the login nodes are in high demand from many users, and process limits reduce the chance of the login nodes becoming overloaded and unusable. You cannot get this limitation lifted for you, but there are steps to take to minimize its impact on you:

Transfer data via the datamover nodes, hpc-data.pawsey.org.au. These are different servers and any processes spawned by scp/winscp/filezilla/etc will not affect you on the login nodes.

Sometimes when you logout processes can get left behind. If you can't log in to kill them, contact the [Helpdesk Service](#). You can kill processes via the kill command, and work out the process number to kill by using ps ux.

Try to limit yourself to two or three login sessions. E.g. one for editing files and another for submitting jobs.

VirtualBox

VirtualBox (<https://www.virtualbox.org>) is amazing software developed by Innotek GmbH, which is part of the Oracle corporation. VirtualBox can virtualize a "guest" computer inside another computer and allows users to install a large number of "guest" operating systems, including several Linux family flavours (Ubuntu or openSUSE, for example).

After installing Linux in a virtual box, you will not only be able to perform ssh connections easily, but also make use of a whole linux environment similar to that on our systems. Within a Linux virtual machine, you can install the same tools intended to be used in Pawsey supercomputers. This will allow testing and post-processing within the local Linux environment, and also provide a great opportunity for improving Linux and scripting skills (which are extremely useful for the best use of Pawsey systems).

Putty

A free client, called PuTTY, is available for all recent versions of Windows. It can be downloaded and installed following the instructions on the PuTTY website, provided you have the appropriate permissions for your computer. Otherwise, you may need to ask a system administrator to complete the installation for you.

<http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>

Cygwin

Cygwin provides a Linux environment within Windows and can also be used to login into Pawsey systems. It is a port of many applications to Windows, including ssh and X, and is available from the following link: <http://www.cygwin.com>

Logging in with graphical interaction

Our systems do not provide a graphical environment of the operating system, as for Windows, Apple or Ubuntu desktop computers. Therefore, most of the interaction with Pawsey systems is performed by executing commands within the terminal opened with ssh. Nevertheless, some of the tools in Pawsey systems do have Graphical User Interfaces (GUIs). To allow these tools to control your local display and be able to open up a new GUI (such as for a debugger), use the X forwarding option "-X" of SSH, as explained in the separate page: [Use graphical \(GUI\) programs with X over SSH](#).