

How to connect to the Serpent Server

Author: Fredrik Dehlin

Date: January 28, 2021

This short text will act as a quick reference guide for you when you are connecting to KTH's Serpent Server.

Connection Details

IP: 130.237.70.233

SSH Port: 2222

Credentials: Obtained on request

Your first connection

Depending on which operating system you are using you can connect to the server using different methods. If you are using a Unix-based operating system e.g. macOS or a Linux distribution the recommended connection procedure differs from that of a Windows user's, and this tutorial will show the connection procedure for both. Furthermore, a procedure to transfer files between the server and your local system will also be explained.

Unix-based systems

The procedure to connect from a Unix-based system will be shown on a machine running Ubuntu 20.04, but the procedure is identical on a system running e.g. macOS. When connecting to the server from my Ubuntu computer I prefer to use two built in tools, **ssh** and **sftp**, where ssh is a protocol to establish an encrypted tunnel allowing for remote access and sftp is a protocol to transfer files between the server and your local machine.

ssh

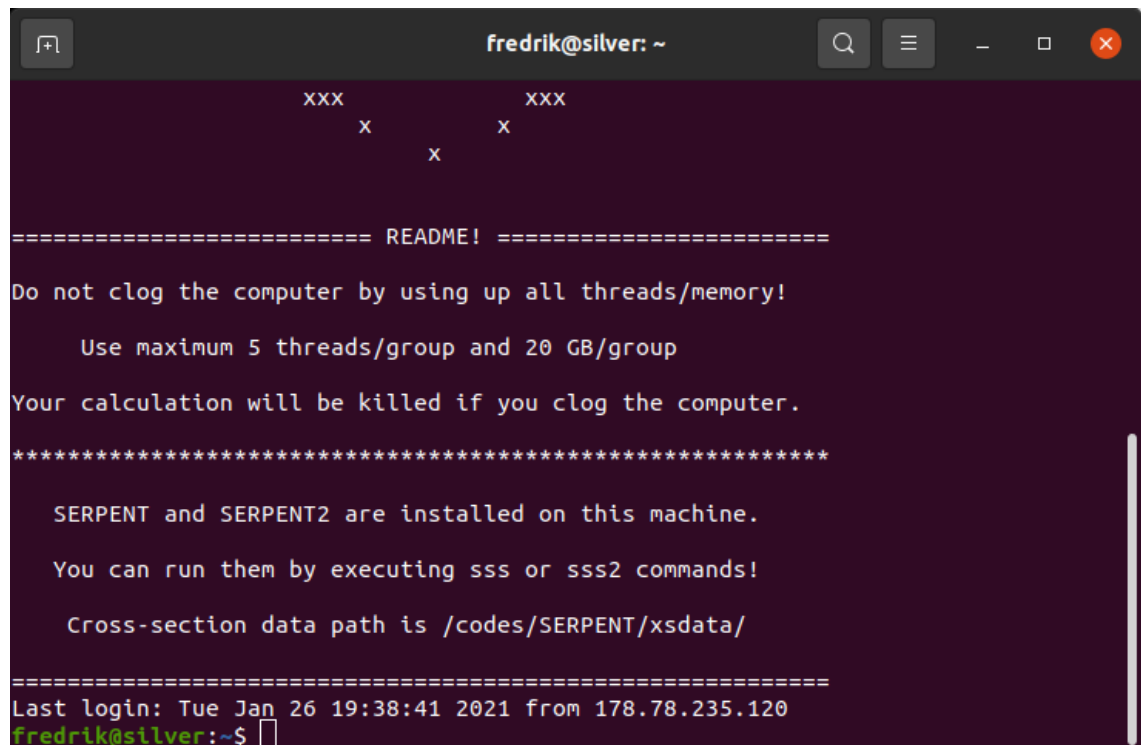
The operating system should come with ssh pre-installed, and to connect using the ssh protocol follow this procedure:

1. Open the terminal and type

```
$ ssh -p 2222 <username>@130.237.70.233
```

where instead of <username> you type your own server username.

2. The first time you connect to the server using ssh you'll be asked if you want to connect since The authenticity of the host can't be established. Type **yes** in the terminal window to proceed.
3. Type in your password and type enter.
4. You should now be connected to the server and be greeted by a window looking like this



```
fredrik@silver: ~  
xxx      xxx  
  x      x  
    x  
  
===== README! =====  
Do not clog the computer by using up all threads/memory!  
    Use maximum 5 threads/group and 20 GB/group  
Your calculation will be killed if you clog the computer.  
*****  
    SERPENT and SERPENT2 are installed on this machine.  
    You can run them by executing sss or sss2 commands!  
    Cross-section data path is /codes/SERPENT/xsdata/  
=====  
Last login: Tue Jan 26 19:38:41 2021 from 178.78.235.120  
fredrik@silver:~$
```

5. At this point you have completed the connection process and can start executing commands remotely.

sftp

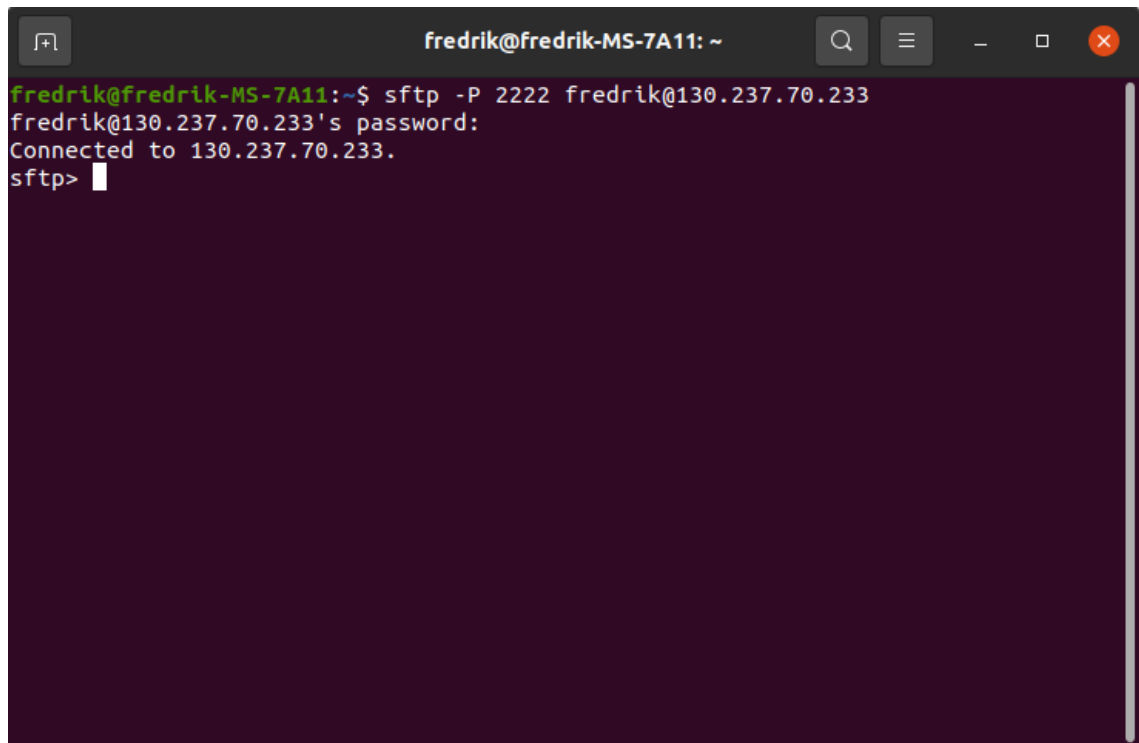
This protocol will allow you to transfer files and folders to and from the server.

1. Open a terminal window and type

```
$ sftp -P 2222 <username>@130.237.70.233
```

where instead of **<username>** you type your own server username.

2. Type in your password and click enter. You should now see a window similar to this

A terminal window titled 'fredrik@fredrik-MS-7A11: ~' with standard window controls. The terminal shows the execution of 'sftp -P 2222 fredrik@130.237.70.233'. It prompts for a password, shows 'Connected to 130.237.70.233.', and then shows the 'sftp>' prompt with a cursor.

```
fredrik@fredrik-MS-7A11:~$ sftp -P 2222 fredrik@130.237.70.233
fredrik@130.237.70.233's password:
Connected to 130.237.70.233.
sftp> █
```

3. The following set of commands are useful when using sftp:
 - `ls` - list the contents in a remote directory
 - `lls` - list the contents in a local directory
 - `cd` - change remote directory
 - `lcd` - change local directory
 - `get` - retrieve a file from the server to your local computer
 - `put` - send a file from your local computer to the server
 - `mkdir` - create a new remote directory
4. Lets assume that I have a folder containing a set of Serpent input files I want to transfer to the server. They are located in my **Examples** folder with the following path: `~/Documents/SUNRISE/Gen-IV_Teaching/Example`. In the sftp terminal window I navigate to that location with

```
sftp> lcd /home/fredrik/Documents/SUNRISE/Gen-IV_Teaching/Example
```

With `lls` I list the local folder content and notice that I have two files, `serpent_input` and `serpent_input2`, in that folder.

5. To transfer `serpent_input` to the server I type

```
sftp> put serpent_input
```

and the file is transferred to my remote working directory. But what if I want to transfer both files? I will show two different methods.

6. The first method is to type

```
sftp> put ./*
```

which will upload all content of your local folder to your remote folder, but it is also possible to transfer a folder with all of its contents.

7. Begin by move up one location in the local directory tree by typing `lcd ..` and now you can transfer the entire folder **Example** with all of its content by typing

```
sftp> put -r Example
```

where the flag `-r` tells sftp to upload **Example** with all of its content.

8. At this point you return to the ssh terminal window and execute `Serpent2`, or any other command, on the files you've just uploaded.
9. Finally, the process explained to retrieve a file or a folder with all of its content uses the same syntax as described in point 5, 6 and 7 with the difference that `put` is replaced by `get`.

Windows systems

The following section will demonstrate one possible method to connect to the server when you are using a Windows computer.

1. A recommended approach when using a Windows system is to download the SSH Client PuTTY and the SFTP client WinSCP.

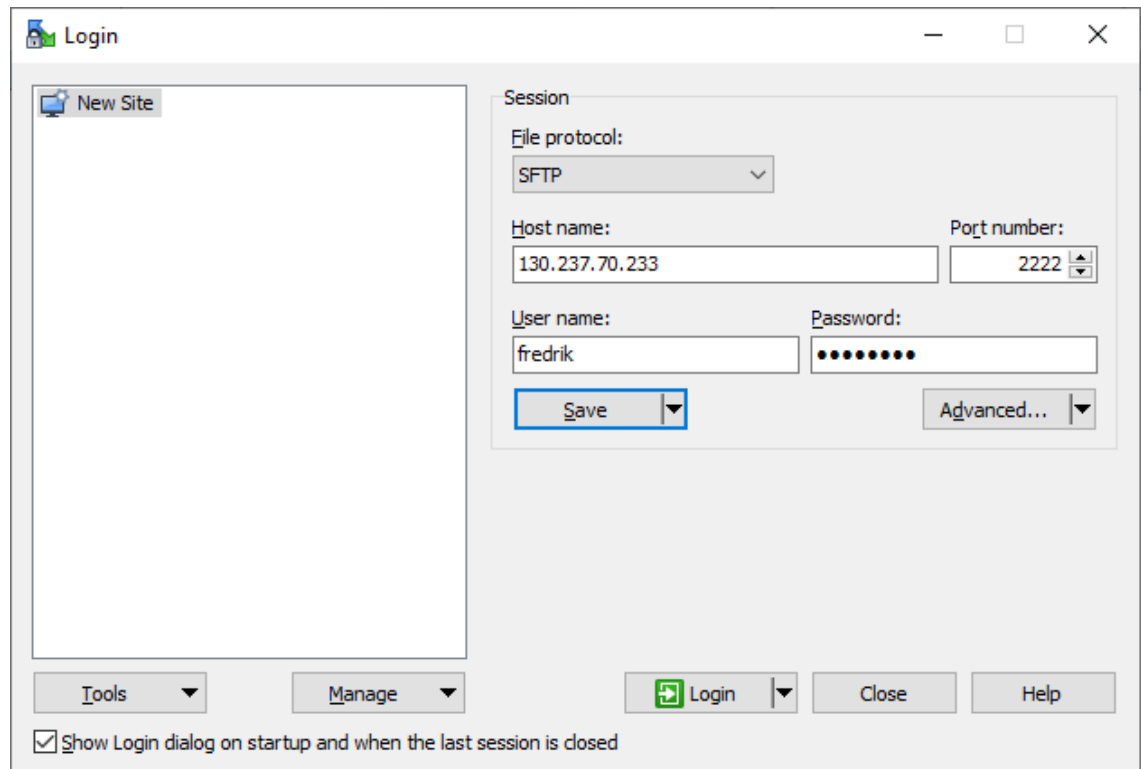
Links:

PuTTY: <https://www.putty.org/>

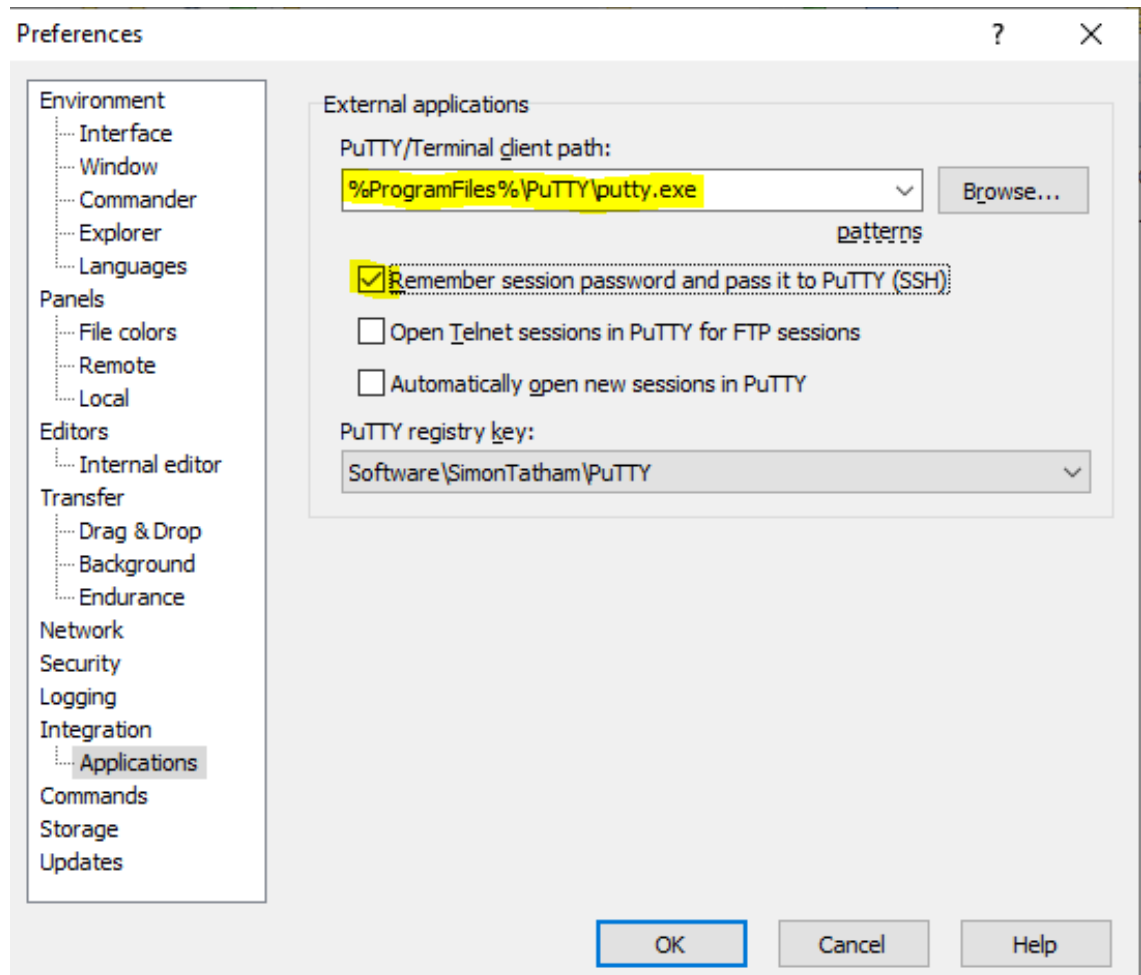
WinSCP: <https://winscp.net/eng/index.php>

The software mentioned above is solely a recommendation from my part, and you are free to use another program if you prefer it over my recommendations.

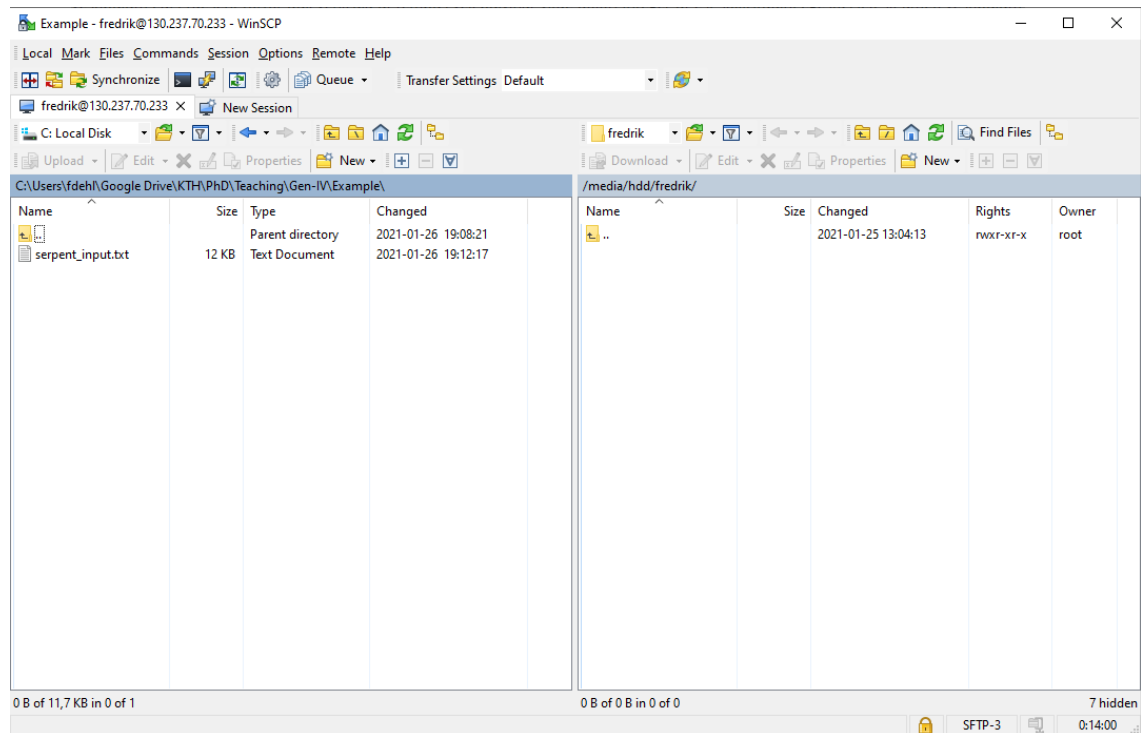
2. Once the above software is installed it is time to launch WinSCP and you'll be met by a login window where you'll enter the server's **IP** under host name, **SSH Port** under port and your own **Credentials** under user name and password. Don't forget to save to speed up successive connections! Your window should now look like this:



3. To simplify the SSH connection head to Options > Preferences > Integration > Applications and tick the box next to Remember session password and pass it to PuTTY (SSH). Also verify that the correct path to putty.exe is shown under PuTTY/terminal client path:



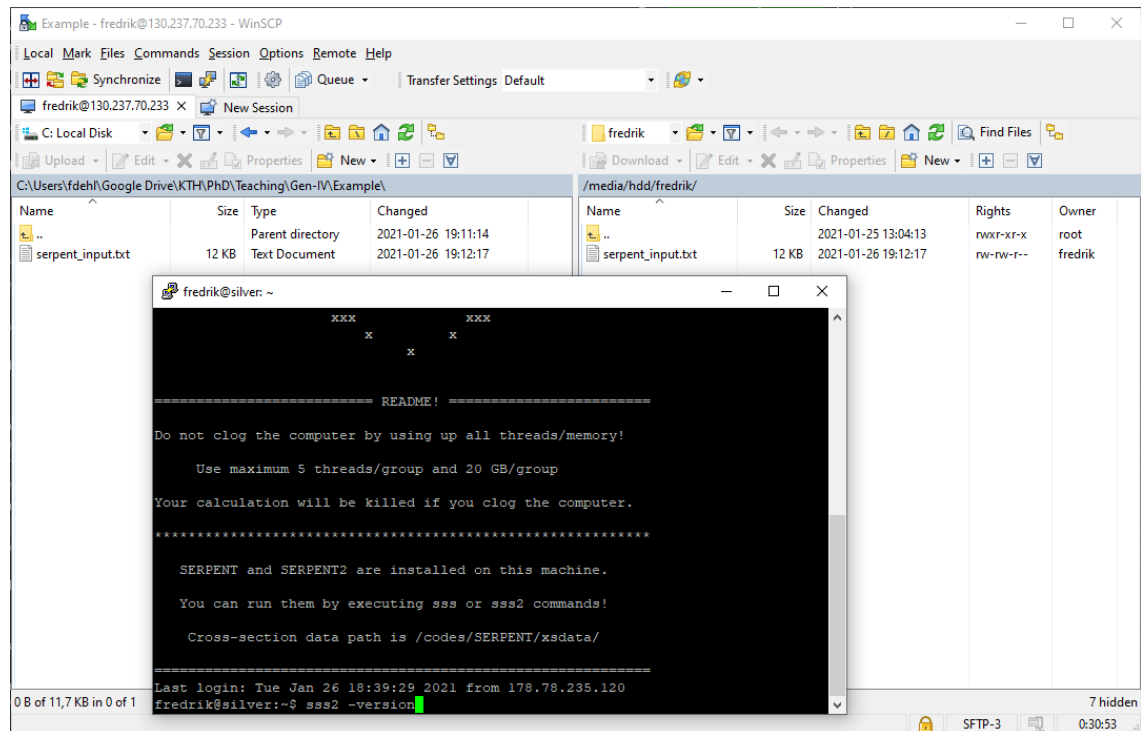
4. The main WinSCP window will look similar to the one shown below. The left box shows the contents in your local file path and there you navigate to the folder where you've saved your Serpent input file. In my case `C:\Users\fdehl\Google Drive\KTH\PhD\Teaching\Gen-IV\Example`. Moreover, the right box shows the contents in your remote file path, i.e. your working directory on the server.



5. At this point you can use the GUI to drag and drop files between the two locations.
6. When it is time to run your Serpent2 simulation you need a terminal window which can be obtained by opening PuTTY via either `Commands > Open in PuTTY` or the default keyboard short-cut `Ctrl + P`.

7. From the PuTTY terminal window you can execute commands on the server, e.g. check the installed version of Serpent2 with

```
$ sss2 -version
```



After an update to Windows 10 in April 2018, Microsoft has decided to include a SSH-client by default meaning that the procedure explained in the Unix-based system section now also applies to Windows. If you prefer to use a terminal based approach instead of the GUI approach shown above, you can use either CMD or PowerShell and follow the Unix-based system connection methodology.

IMPORTANT! Keep in mind that Serpent2 requires that the input file is written in a Unix format. You cannot execute a plain text file written in Windows (DOS-format) without conversion. One method to perform this conversion is by using the tool `dos2unix` which is installed on the server, e.g. if the input file `serpent_input` is written in Windows you convert it to Unix by running:

```
$ dos2unix serpent_input
```