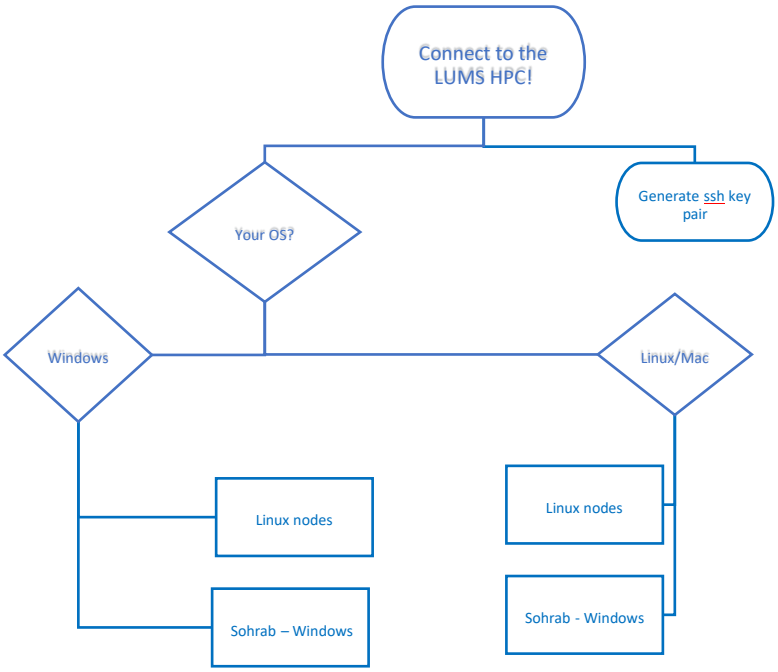
2021

How to connect to the HPC Computing resources

How to connect to the HPC - Linux cluster/nodes



Before start using HPC, you need to know about a few essential things you must be familiar with:

- Connecting to Linux Computing resources
- Connecting to Windows Computing resources

1. To connect to the HPC, we need to have the SSH client to get access to compute nodes. This gives us the command - line option. The software on the client-side depends upon the host operating system. We will use PuTTY for Windows, while for Linux and Mac OS, we can SSH directly from the terminal.
2. We need to know how to transfer files back and forth from local machine to HPC.
3. Optional: If you intend to use program with a Graphical User Interface (GUI), you need to have an X-server on the client-side and X - forwarding enabled in the remote machine (HPC nodes)
4. It is often the case when several versions of a program or libraries are installed, to use the appropriate one according to your needs, LUMS HPC (currently only in the cheetah cluster) supports modules, you can select and load the modules you need.

We will cover the connectivity to the HPC in this document only.

1 Connecting from Windows to Linux nodes

- Get Putty - a Free SSH client

PuTTY utilities are freely available from
https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html

Instead of installation PuTTY package, which contains several components, you can download PuTTY and PuTTYgen executable binary files only. This will be particularly useful if you don't have the installation privilege on the client machine.

From the PuTTY package, we will use two components:

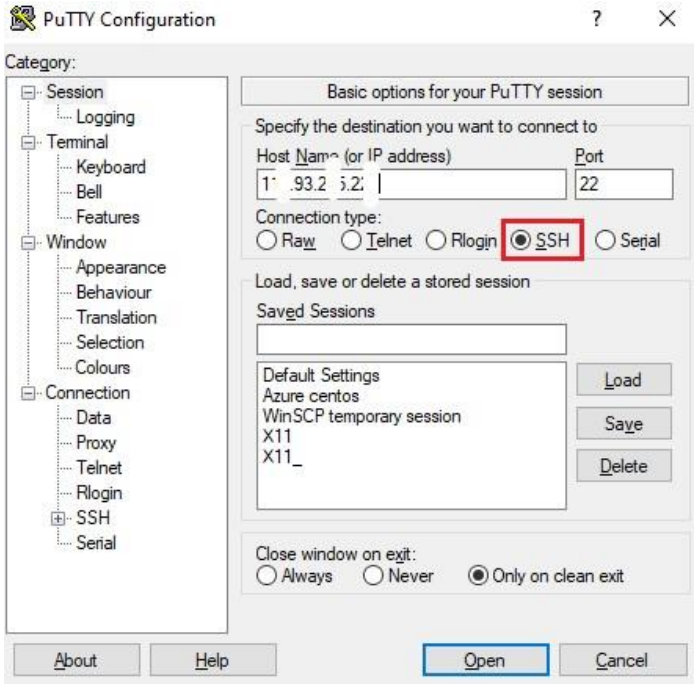
PuTTY: The SSH client itself

PuTTYgen: an RSA and DSA key generation utility

1.1 Login without using key - pair

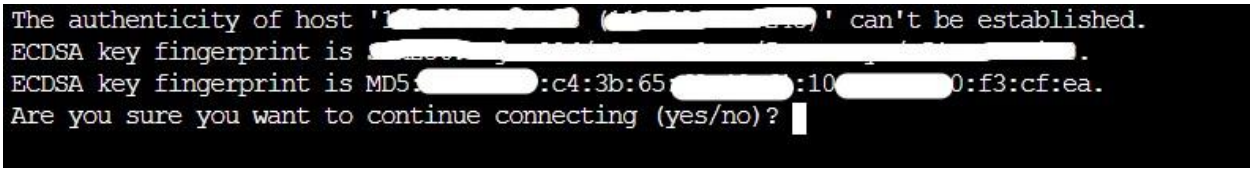
After obtaining our account details, you can log in to the HPC using the provided user name and password you opted.

- Open the PuTTY app or just start the PuTTY executable "**putty.exe**". The program window will pop-up.
- Under the category, Session, in the field **HostName**, enter the IP address or name of the node/cluster, as shown in the Figure below. You can click **open** to proceed.



If you click **Open** at the bottom of the dialog box now, then it will ask for the username and password. However, for the first-time connection, the following message (in a box) will be shown before it asks for username and password:

- Click **yes** (or type **yes** and hit **Enter** if you are accessing from the terminal).

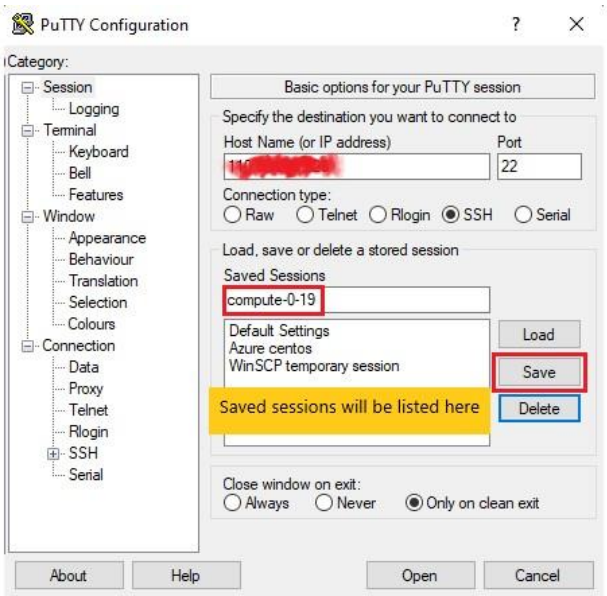


Note: If this message appears again in the future login (after you pressed yes), then either the host key might have changed, or someone's device might be pretending to be server! Which is dangerous! Be careful.

In the PuTTy configuration, the **default ssh port is 22**, which is mostly the case unless the admin has set something else (for the security reasons). We will use port 22 for all our servers.

1.2 Few cool things to do in PuTTY! (Optional but useful)

- Instead of typing the IP and username each time you try to log in, you can provide and save these setting to use in the future, as shown in the next picture.
- Username can be provided in the **Data** subsection under the category **Connection** in PuTTY.
- Type any name for the session and it will be listed in the save session list.



2 Use SSH efficiently

When you are working with multiple machines at the same time, you might often require switching between them via SSH. You need to provide the login credentials each time you try to access the clusters or nodes. SSH has a feature that makes our life exceedingly easier and that is: **key-based login**.

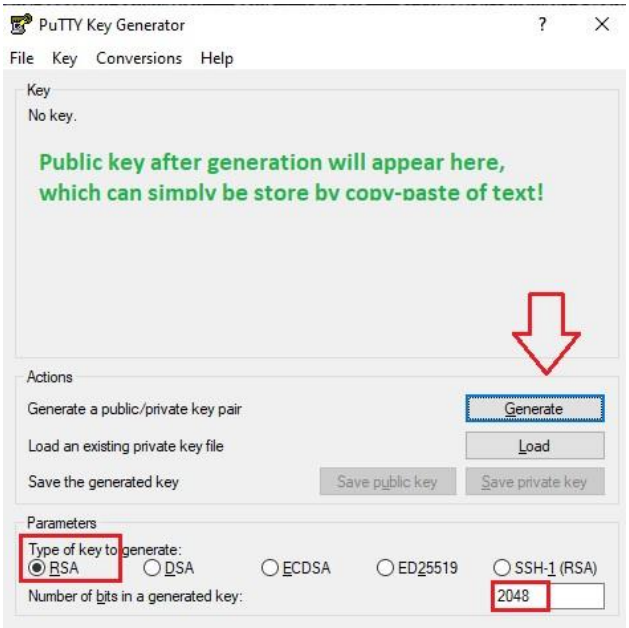
Key-based login is a method where two files replace your password: **a public key and a private key**. These two keys are attached and can only be used unitedly. We can think of the public key as a lock to which the key to open the lock is the private key. You can share your public key (lock) with anyone, and no one will be able to open it as long as you keep the key secure.

Anyone who gets your private key would be able to portray you even without your knowledge. It is wise to use separate key pairs for each machine you want to connect from. This allows you to remove access in case you lost your machine.

2.1 Generating public and private Keys

To generate the private-public key pair, we will use the program PuTTYgen. The window of the program is shown in Fig below.

We will go with the default parameters ("RSA" key & 2048 bits in the key) to generate the key



Click on the Generate, you need to **keep moving the mouse over the PuTTYgen window**! Actually, this creates some random data to be used for key pair generation. Once the pair is generated, the public key will become visible in the field.

It is good practice to fill a suitable **Key comment**, which might prove helpful in the key's identification.

Next, (although it is optional) it is stressed to have a passphrase set! This will increase the security and won't allow the unauthorized use of a private key (to open the lock). You should only avoid it in case you are sure that your PC is not shared, and your disk drive is encrypted (which is mostly not the case!).

We recommend using the name " id_rsa.pub" for the public key, and " id_rsa.ppk" for the private key.

3 Uploading public key to the server

You need to upload the public key to the server in the following directory:

/home/your_username/.ssh

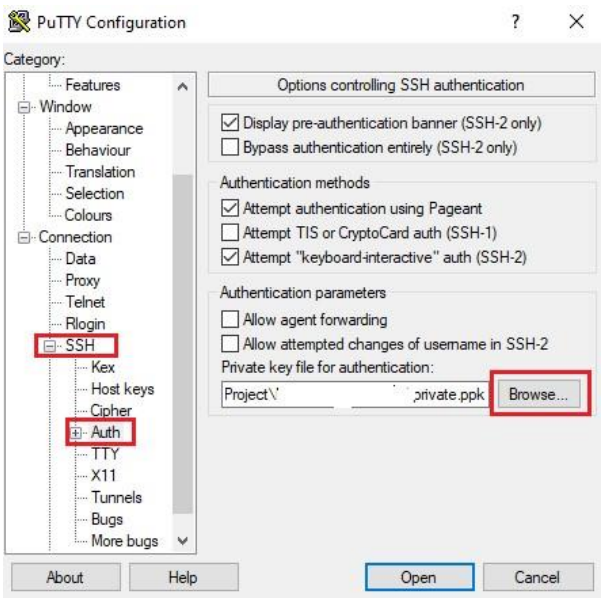
You can do it in the way you want. Two popular methods are:

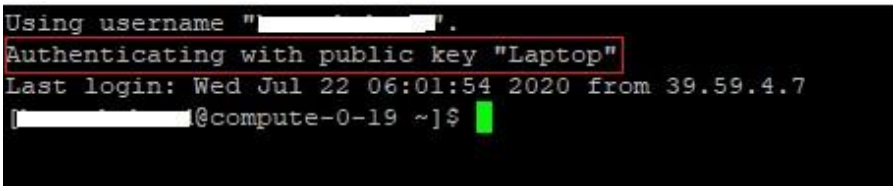
1. Create new file using editor, then paste the key and save ○ Go to the designated directory and make a new file name "authorized_keys" using nano or vi editors. (see Tutorial … on using nano & vi editors)
    - You need to give the full read and write permissions for your username only, to do that, issue the following command (from inside .ssh directory):
    - $ chmod 600 authorized_keys
2. The second way to upload the public key file is to use WinSCP software (from windows). To know how to use WinSCP, see Tutorial ().

Note: Don't save the public key by "Save public key"! It will probably save the key in the wrong format!

3.1 Making connection: Connecting public and private keys

- Now, it is time to point the private key to the uploaded key! Open the PuTTY again.
- You will either have a save session already or you can make a new one.
- The only thing you need to do is to go to the **ssh > Auth > Browse…** and then selecting your save "Private key". That's it.
- Go back to the Session and save it. Now you will be able to connect the server with just one click!





Congratulations, you are on the HPC infrastructure now!

Tip: To see at which directory you have landed, you can check the directory, "Print the Working Directory" (pwd): $ pwd and name of the host by:

$ hostname

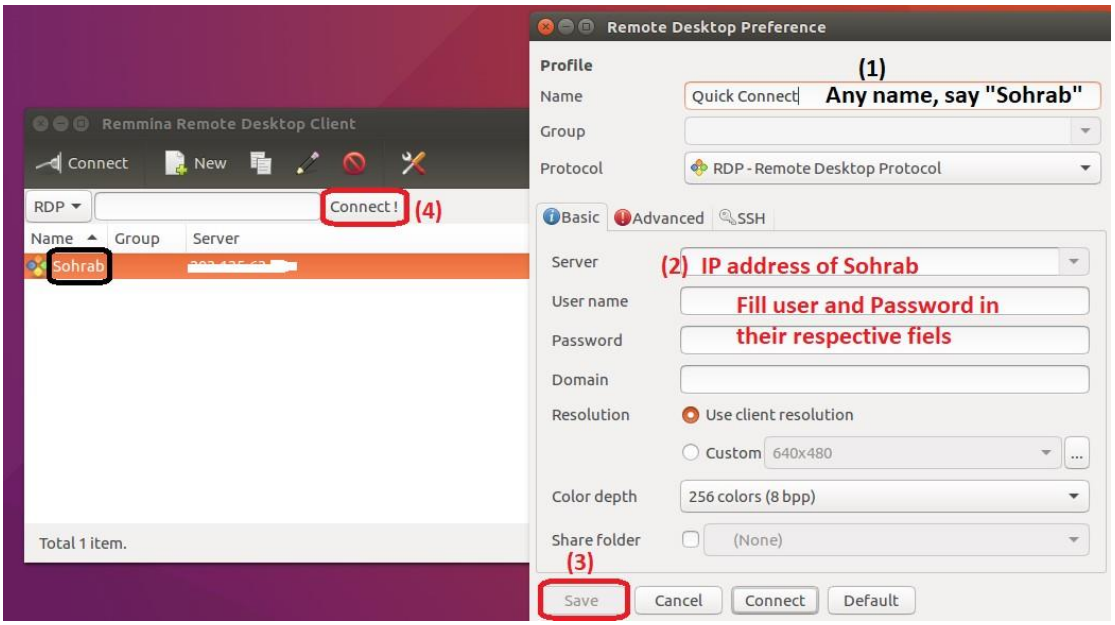
4 Connecting from Linux/Mac to Linux nodes

Connecting from a Linux or Mac machine to the Linux based HPC nodes is the straightforward task.

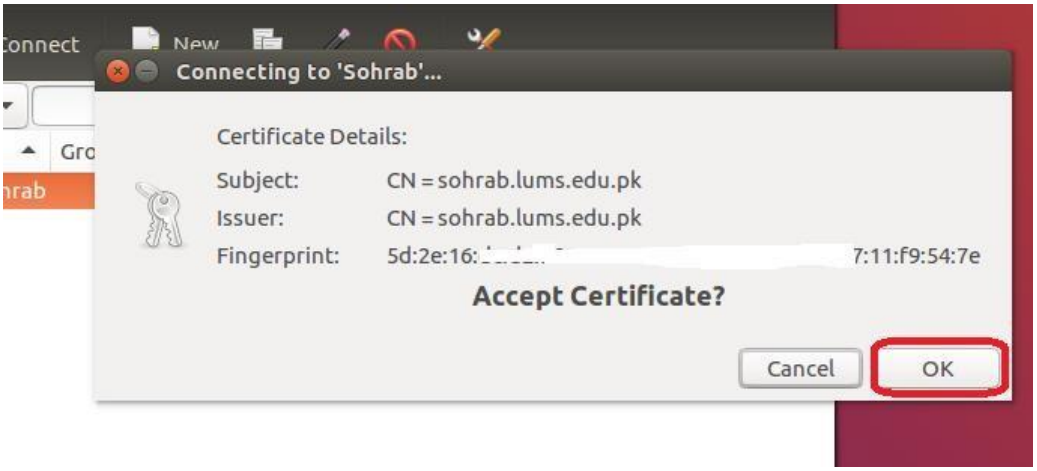
- Open the terminal in Linux
- Type ssh (IP here)
- Type yes (better to do this only if you have confirmed the server you are connecting to!)
- Provide your username and password!
  To have the public and private key, follow the same instructions as given in Section …

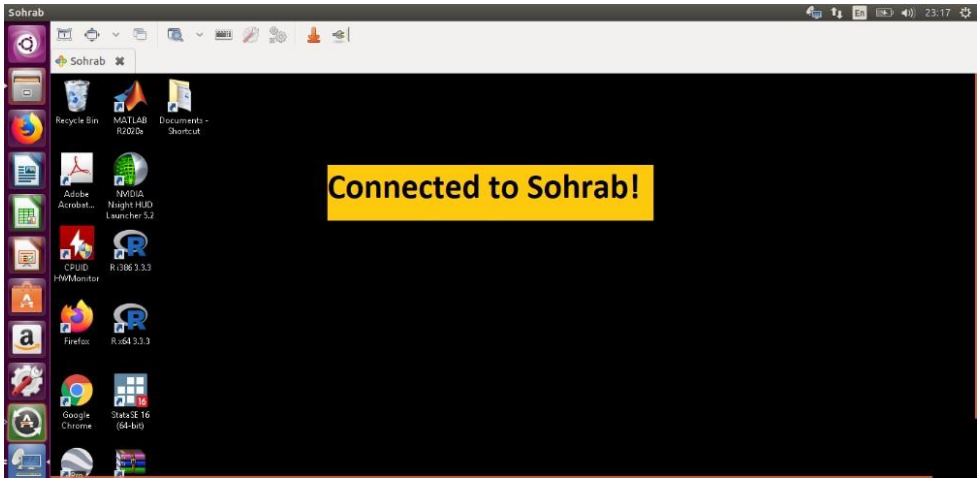
5 Connecting from Linux to Sohrab (Windows based)

- You can install different RDP tools in Linux environment! One of them is "**Remmina Remote Desktop**".It can be installed just like any other Linux package.



- Accept the certificate!





6 Connecting from Windows to Sohrab (Windows based)

This one is the easiest! You just need the built-in remote desktop connection app of windows and type the IP of Sohrab system there. You will be asked to provide username and password assigned to you by the HPC admin.

7 References:

1. Public and private keys: https://www.youtube.com/watch?v=PFZz_xi8Hy0

Acknowledgement:

The Scientific Pedagogical group consisted of

Faculty/Staff Partner

- Nouman Zubair

Students

- Haseeb Ahmed
- Muhammad Adnan
- Muhammad Matloob Altaf