

A lot of information regarding the cluster has already been compiled by Andy [here](#). You do not need to know all of that. Below, I have written down a few notes which might be useful for our joint project!

Login into the cluster

You just need to use ssh to log in into the *submission node* (called macomp001)
In the terminal of your pc type

ssh your_college_username@macomp001.ma.ic.ac.uk

For example, my college username is nz3618 so I just type
ssh nz3618@macomp001.ma.ic.ac.uk

It will then ask you to prompt your college password which you will have to do unfortunately every time you log in. If it's difficult to remember, you can copy and paste it on the terminal.

Connecting from outside the college network:

If you are connecting from a network which is not the Imperial one (e.g. you're on a Cycladic island), you need to set up a VPN to log into the cluster. Imperial uses a VPN service called Tunnelblick. Instructions on how to set this up are [here](#). It seems they are now asking people wishing to use the VPN service to write to ICT Service Desk. I did this some years ago and thus I still have access to it. If you do not have a VPN yet, maybe you need to do this. Let me know if this is the case and I can help you with this.

Once logged in the cluster

You will be inside your home folder and you can use the terminal as if you were using it on your PC! I am assuming that you have some knowledge on basic shell commands (as in "**cd ../**", "**mkdir ...**"). If not, I can send you some notes about this too!

You can use this terminal exactly like you would on your PC. For example, if you wanted to run a Julia script "ciao.jl" you could simply type on the terminal

julia ciao.jl

And it would run! However, there is no point in using the cluster this way, because you are using the login node to do some calculations which are much better performed by "outsourcing" them to compute nodes within the cluster. Through the queueing system, we can ask the cluster to run some jobs for us and save the resulting data in our home folder.

By the way, sometimes running scripts in the login node is prohibited because it would slow down every other user joining the cluster. In the Maths cluster this is not the case, unless Andy notices it and kills your job :) However, you could run some very quick scripts (I usually run scripts to analyse the data that I have saved) or for de-bugging purposes!

Before submitting jobs to the cluster

It is essential that you follow a few instructions before submitting jobs otherwise you might not see the results of the jobs you submit!

Since jobs will be running on nodes different from the login node, we need to make sure that transfers among nodes will run without us having to prompt the password each time!

- 1) In the cluster, go to your home directory. If you do not know where it is just type in the terminal

cd ~ (~ is your home directory!)

- 2) Create a SSH key pair by typing in the terminal (from your home directory)

ssh-keygen

It will ask you to prompt a passphrase. **Do not put anything, just hit return every time it asks you for a passphrase.**

This will create a SSH key pair in your `.ssh` folder accessible from your home directory. (Just a quick note, folders starting with a dot are hidden, so if you just type the usual command to see what's inside a folder, which is "**ls**", you won't see them. You need to actually start typing "**ls .**" and then press the tab key and let the terminal show you all the hidden folders)

- 3) Go to the `.ssh` folder and copy the public SSH key in a file called `authorized_keys` within the `.ssh` folder. In order to do so you need to type

cd .ssh (you are moving to the `.ssh` folder)

cp id_rsa.pub authorized_keys

- 4) Go back to your home folder and type the command

update-ssh-known-hosts

Basically this script logs into every node (with the SSH key you have just created) and records it. Wait until the script is complete and then you are ready to submit jobs to the cluster!

Setting up Julia to work in the cluster

Julia is supported on the cluster, you can verify this just by typing

julia

In your terminal on the cluster. You will see that there is an installation of Julia (quite old actually, but it will suffice for our purposes). If you want to get out of the Julia environment just type **exit()**

Each user of the cluster has a folder named `.julia` (hidden, once again) where all the libraries that are needed get "stored"! You should be able to see it by going to your home folder and typing

ls . (And pressing tab)

If you don't (as it was in my case), don't worry! We just need to download first a few libraries that we need! Libraries in Julia are managed using Pkg. Whenever you want to download a library you will type in Julia

```
using Pkg
```

```
Pkg.add("MyLibrary")
```

Which will download and install library "MyLibrary".

So, let's download libraries. Type in the terminal

```
julia
```

And when you are in the Julia environment, type

```
using Pkg
```

```
Pkg.status()
```

This should set the .julia folder that I have just mentioned before.

Now you can add all the libraries you want. For our project (so far) we need libraries Graphs, LinearAlgebra, StatsBase, DataFrames, CSV and JLD2. Try to install them with the **Pkg.add()** command!

If the libraries have been correctly installed and pre-compiled, if you type

```
using Graphs ( I took the Graphs library as an example)
```

Should work smoothly and no errors should arise!

Downloading and Uploading files on the cluster

We need to find a way to send local files (such as scripts to be run on the cluster) to the cluster and vice-versa. We can do this using the scp command, which in its basic form is

```
scp ./ciao.jl your_college_name@macomp001.ma.ic.ac.uk:~
```

The scp command will send the script "ciao.jl" (which is in the folder where we are typing the command from) to the home folder in the cluster (remember ~ is the home folder of the cluster). It will ask you for your college password exactly as when you log in into the cluster.

Of course we can download stuff from the cluster in the same way

```
scp your_college_name@macomp001.ma.ic.ac.uk:~/our_project/ciao.jl ./
```

The previous command will download the script ciao.jl which is in a folder called our_project (folder which is itself in the home folder on the cluster) and will save it in your local folder from where you are typing the command.

If you are downloading or uploading a folder, you need to make sure to use the **-r** flag in the scp command, e.g.

```
scp -r your_college_name@macomp001.ma.ic.ac.uk:~/our_project ./
```

The above command will download the whole folder our_project from the cluster to your local PC.