Tinney HPC Cluster Introduction Work Book

November 2025.

Help and support resources:

1. https://www.tchpc.tcd.ie/support/gettingstarted
2. https://www.tchpc.tcd.ie/docs/ - Research IT's documentation page
3. https://www.tchpc.tcd.ie/docs/tinney/ - Tinney specific documentation
4. https://www.tchpc.tcd.ie/support/contact

Logging In

- Login to your assigned cluster via SSH with your assigned account. Instructions: https://www.tchpc.tcd.ie/docs/cluster-usage/#logging-into-clusters-from-the-trinity-network

  ```
  ssh -X -l username tinney.tchpc.tcd.ie
  ```

- Available from the College network, (including VPN & Wireless), via SSH/SCP/SFTP

Basic Linux

Linux Commands Cheat Sheet https://www.linuxtrainingacademy.com/linux-commands-cheat-sheet/

command	what it does
`ls`	list current directory contents
`ls -la`	show more details for `ls`
`pwd`	show the current working directory
`cp file1 file2`	copy file1 to file2
`cp -r dir1 dir2`	copy directory dir1 to dir2
`mv file1 file2`	rename/move file1 to file2
`mkdir directory`	create the directory called directory
`nano file1`	open the file1 file with the nano text editor
`cat file1`	display the contents of file1, must be a text file
`less file`	display the contents of file1, must be a text file, allows for scrolling the file

Transferring Files

Instructions on how to transfer files at https://www.tchpc.tcd.ie/docs/cluster-usage/#transferring-files.

You may need a client like WinSCP, https://winscp.net/eng/download.php, or similar.

Using Software

Lots of software is installed and can be installed but it may not immediately be available to you. You will have to take steps to make it available to you.

The `modules` system is used to control access to software. It modifies your environment, e.g. adds locations to your `$PATH` variable to look for additional places were software is installed to.

List of `module` commands

command	what it does
`module av`	see what software is available
`modgrep matlab`	search for gcc in modules, works for other patterns
`module load matlab`	load a module
`module list`	see what modules you have loaded
`module rm matlab`	unload a module
`module purge`	remove all modules

Working with the resource queue

To get access to the computational resources in the cluster you must use the queuing system. Slurm is the resource manager used, thus why all its commands start with `s`.

Getting info on the queue - `squeue`

command	what it does
`squeue`	show the queue status
`squeue --user username`	show a particular users jobs in the queue only
`squeue --start`	show when jobs are estimated to start
`squeue --start --user username`	show when user smcgrat's jobs are expected to start
`squeue --user username -l`	Display long output about my jobs in the queue

Getting info on resources available - `sinfo`, e.g. what nodes are down in the queue.

command	what it does
`sinfo`	Display queue/partition names, run times and available nodes
`sinfo -Nel`	more detailed view of resources available in each node

Get info on a job - `scontrol`

```
$ scontrol show jobid 10809
```

Cancel a running or pending job:

```
$ scancel 10809
```

You will need to know the jobid number and have permissions to cancel the job. I.e. you can only cancel your own jobs.

Submitting Jobs

Interactive Jobs

Interactive jobs are were you request the resources in the cluster but then do the work yourself on those nodes, instead of having the scheduler do it.

Interactive allocations, where you log directly into the node:

```
$ salloc -N 1
$ module load apps matlab
$ matlab -nodesktop
```

The `$SLURM_NODELIST` variable will list the compute node(s) you have been allocated.

Then do your work on the node. If you need to use graphical tools remember to connect with X Windows forwarding, https://www.tchpc.tcd.ie/docs/cluster-usage/#graphical-user-interface-gui-application-usage-in-an-interactive-job, enabled.

Batch Jobs

Batch jobs are ones you submit to be run by the resource manager.

Batch jobs are submitted with e.g.: `sbatch submit-cpie.sh` where the contents of `submit-cpie.sh` could be as follows. Note, you can call your submission file, `submit-cpie.sh`, whatever you want.

E.g. - Matlab script.

This matlab script, `multiply_args.m`, will calculate the product of 2 numbers provided to it as arguments to the script. (Copilot used to create the script).

```
% multiply_args.m
% Usage: matlab -batch "multiply_args(arg1,arg2)"

function multiply_args(a, b)
    % Convert input arguments to numeric values
    num1 = str2double(a);
    num2 = str2double(b);

    % Check for valid numbers
    if isnan(num1) || isnan(num2)
        fprintf('Error: Both arguments must be numeric.\n');
        return;
    end

    % Compute product
    product = num1 * num2;

    % Display result
    fprintf('The product of %g and %g is %g.\n', num1, num2, product);
end
```

A submission script, called `submit-matlab.sh`, will run the matlab calculation in batch mode on the cluster.

```
#!/bin/bash
#SBATCH -n 8 # 8 cores
#SBATCH -p compute
#SBATCH --reservation=intro
#SBATCH -t 1-00:00:00
#SBATCH -J "Job name"

# load modules, you will need to modify this for your needs
module load matlab

# run your work, you will need to modify this for your needs
matlab -batch "multiply_args('5','7')"
```

This e.g.

- requests 8 CPU cores. This is mandatory.
- in the `compute` partition. This is optional.
- In the `intro` reservation. This is only needed for the day of the introductory course, won't usually be needed.
- For 1 day with the `-t` flag. This is optional. It can be set to any value less than the the maximum time jobs can run for. See the `TIMELIMIT` column

from the `sinfo` command.
- Gives it a job name. Optional.
- Loads some modules.
- Runs a matlab script in batch mode.

Submit to the queue with: `sbatch submit-matlab.sh`

E.g. - Bash calculations.

The `submit-hello-world.sh` file.

```
#!/bin/bash
#SBATCH -N 1
#SBATCH -p compute
#SBATCH --reservation=intro
#SBATCH -t 00:10:00

echo "Starting to Run job now"
echo ""

# load some modules
module load apps

# run your code
counter=0
maximum=24
while [ $counter -lt $maximum ]; do
        echo "Hello world number $counter"
        ((counter++))
done

echo ""
echo "Job complete"
```

To run it: `sbatch submit-hello-world.sh`

E.g. - Script to install packages in `R`.

An example script is available on Tinney at `/home/support/rl8/apps/r-env/setup.sh`.
It does the following it install some `R` packages.

```
R -e "install.packages(’devtools’, repos=’http://cran.rstudio.com/’, lib=’$lib’, dependencie
R -e "install.packages(’Rcpp’, repos=’http://cran.rstudio.com/’, lib=’$lib’)"
R -e "install.packages(’RcppArmadillo’, repos=’http://cran.rstudio.com/’, lib=’$lib’)"
```

It is available in the modules system. To load it and run it do the following.

```
module load r-env
setup.sh # Note, this will take several minutes.
```

Other `R` packages can be installed through the same method. From the `R` interrupter the usual installation method, e.g. `install.packages("PackageName")` will work. Packages will be installed into your home directory so available cluster wide.

The `Rscript` utility is also available to run `R` scripts.

`staskfarm` **Note**. To get `staskfarm` to work ensure to include the following `#SBATCH` directives. Don't bother using `#SBATCH -N ...`, use `-n ...` instead.

```
#SBATCH -n 24
#SBATCH --ntasks=24
```