Using the Habanero HPC Research Cluster

0. Get Access

2 options for graduate students:

- You are entitled to a free account
  - Submit a request form for free HPC access: https://columbia.service-now.com/cu?id=sc_cat_item_cu&sys_id=9876ecc213bd160006c376022244b00a
- You can ask to be added as a new user to an existing HPC group (e.g.: cwc; sipa). Current HPC customers can request access to their HPC group for a new user by emailing rcs@columbia.edu (e.g.: to be added as a new user to the SIPA group, ask Doug to request that access for you)

Each of these accounts will have a corresponding directory (= storage space) on the server. The shared storage server is named "Rigel", therefore the path to all home and scratch directories begins with "/rigel"

- my home directory associated with my free account: /rigel/home/<UNI>
  - memory = 10 GB. Appropriate for smaller files: docs, source code, scripts...
  - this is the default directory you "land" into when you connect to the server.
- a group account's scratch storage /rigel/<ACCOUNT>
  - Ex: /rigel/cwc has size = 20 TB, no default User Quota.
  - DO THIS ONCE: once you are granted access to this account, create your own directory in it:

```
$ cd rigel/cwc/users
$ mkdir <UNI>
```

1. Connect to the remote server

We use the SSH (Secure Shell) to securely connect to the remote server.

- In MacOS, use the `Terminal` app
- in Windows, download a Unix-shell program

The basic command to connect to any server with SSH is: $ ssh <user>@<serverIP>

```
$ ssh <UNI>@habanero.rcs.columbia.edu
$ type my Columbia password
```

Stay in home directory or go to another (e.g. my directory in the CWC group):

```
$ cd rigel/cwc/users/<UNI>
```

2. Execute simple tasks directly in the *login* node

After logging in, you are in a *login* node.
/!\ Don’t run large computation on the login nodes!
It negatively impacts all cluster users. Running processes longer than a few seconds and/or involving more than one core is STRICTLY FORBIDDEN on the *login* node. For any extended processing, i.e. running R scripts, don’t run them on the login nodes straight from the terminal. Integrate them in a bash .sh script, and submit it to Slurm.

Ex: install R Packages locally:
Pb: Habanero doesn’t keep the R packages I install. So each time, would have to install them again. Instead: Make a directory in my own user space to install/store them.

DO ONLY ONCE:

1. Create a directory `rpackages` where I’ll store installed packages. Will become an additional library path (where R looks for packages) to the existing read-only ones, = my newly created local library:

```
$ mkdir /rigel/home/<UNI>/rpackages
```

2. EITHER: Write one R script that installs all packages in my newly created local library, add that .R script to my remote directory, run it:

```
$ Rscript <LOCAL_CODE.R>
```

3. OR: open R within Terminal. *Start R, type R commands in Terminal, quit R*

```
$ module load R
$ R
```

- type commands one by one: `install.packages(“tidyverse”, lib = “/rigel/home/<UNI>/rpackages/“)`
- or run the whole file:

```
source(“<LOCAL_CODE.R>”)
$ q()
```

DO EACH TIME:
— Add this path to .libPaths() at the top of every R file:

```
.libPaths(“/rigel/home/<UNI>/rpackages/”)
```

3. Submit job (a batch script) using Slurm

From *login* node, submit jobs to Slurm via 'sbatch' and 'srun'

1. Create a bash .sh script that outlines what cluster resources you would like to use and then lists the commands you'd like to execute.
    - cluster resources = nb of nodes needed (need just 1, we're never gonna run that many chains), time (try 3h59 min), …
    - Then list the commands to execute.
    - ex: see `*HelloWorld_R.sh*`
2. Send the .sh script into Slurm:

```
$ sbatch <MyFile.sh>
```

3. Open the created output file: (###### = nb of the job). Contains all output that was produced during runtime, i.e. stdout and stderr.

```
$ cat slurm-######.out
```

4. Quit

Close connection to habanero.rcs.columbia.edu:

```
$ exit
```