FH Cluster Guide

September 22, 2022

Contents

About this Course **5**

Cluster 101 **9**

1 What is a Cluster? **9**

2 Account Setup **11**

2.1 Check your HutchNet ID 11

2.2 Contacting the SciComp Team 11

3 Terminal Setup **13**

3.1 What is a terminal? 13

3.2 Windows Setup 13

3.3 Mac Setup 15

3.4 Linux Setup 16

4 Logging In **19**

4.1 What is `SSH`? 19

4.2 Windows Login 19

4.3 Mac Login 21

4.4 Linux Login 22

5 Submit your first job **23**
5.1 Download the script 23
5.2 Confirm the download 24
5.3 Inspect the script 24
5.4 Submit the script 25
5.5 Check the output 25

6 File upload and download **27**
6.1 Download Cyberduck 27
6.2 Create Connection 27
6.3 Download and edit the script 29
6.4 Upload the new script 30
6.5 Run the new script 31

7 Interactive cluster **33**

Assessment **37**

8 Cluster 101 Self Test **37**

Appendix **41**

9 Where to get help **41**
Submit a Ticket 41
Visit the SciWiki 41

10 Provide Feedback **43**

11 FAQ / Troubleshooting **45**
11.1 FAQ 45
11.2 Troubleshooting 46

About the Authors **49**

References **53**

About this Course

Our goal is to get you running on the Fred Hutch cluster quickly and efficiently with this quick-start guide. As a wise Drivers' Ed instructor once said, **you need to go slow to go fast**!

In this short course, you'll invest a bit of time now to save you time and frustration down the road. Follow along at **any time on your own schedule**. We hope that the following modules will help you take advantage of the powerful resources the Fred Hutch has to offer!

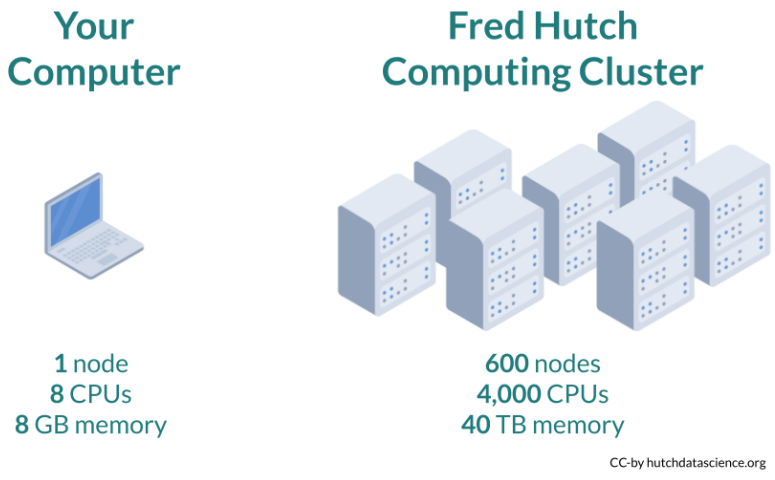
Cluster 101

Chapter 1

What is a Cluster?

A computing cluster is a set of many computers networked together. Because there are many computers working together, the network is able to handle computationally expensive tasks, like genome assemblies or advanced algorithms. Imagine you're building a house. It would take a long time by yourself! It's much better to have many builders working together.

Now that we have a team of workers, the next challenge is task management. A home construction team will need a manager to help delegate tasks. Similarly, the computing cluster uses management software to prioritize tasks, delegate workers (resources), and check on progress. The Fred Hutch cluster uses a common management and scheduling tool called Slurm.



How is the cluster different from a laptop or desktop? First, you might use an operating system like Windows or MacOS. The Fred Hutch server is a Linux

system. Second, because many people use the cluster for many tasks, there isn't a central screen and keyboard. You access the cluster remotely from your computer! We will talk more about how to connect to the cluster in a following chapter.

Computing cluster

A set of computers networked together to perform large tasks.

Chapter 2

Account Setup

You will need an account to log in to the cluster. This ensures that data stays protected.

2.1 Check your HutchNet ID

Your HutchNet ID is the standard login you receive when you start working at the Hutch or are an official affiliate. You can use it to login to most resources at the Center (Desktop Computer, Employee Self Service, VPN, Webmail) and our Scientific Computing systems.

For example:

- my email is `jsmith3@fredhutch.org`.
- my HutchNet ID is `jsmith3`.

If one of your collaborators requires access to the Fred Hutch network you can submit a non-employee action form. Non-employee is a generic administrative term for affiliates, students, contractors, etc.

2.2 Contacting the SciComp Team

To use Scientific Computing clusters at Hutch, your HutchNet ID must be associated with a PI account.

The Scientific Computing Team (SciComp) tries to set some users up ahead of time. However, not everyone is set up automatically. Please fill out this Account Setup Form and we will ensure you are set up correctly!

Errors similar to “Invalid account or account/partition” typically indicate that the account hasn’t been set up by SciComp. This is a quick fix if you use the form above.

Now, let’s set up our Terminal!

Chapter 3

Terminal Setup

The next step is getting familiar with your Terminal. This is your portal to the cluster.

3.1 What is a terminal?

The Terminal is a command line interface. In other words, the Terminal is a software application that allows you to issue commands directly to your laptop or desktop computer. The Terminal is very useful because it allows you to run commands that don't have a graphical user interface (GUI). It can also connect you to computer networks, such as the Fred Hutch cluster! The Terminal setup is different depending on your operating system. Jump to the Windows, MacOS, or Linux sections below.

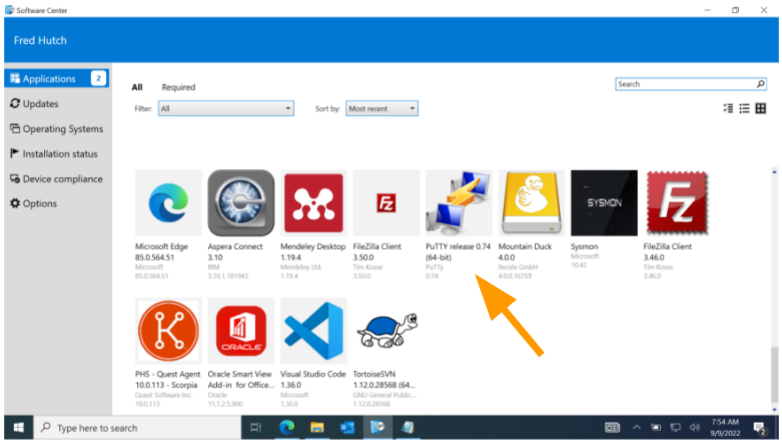
"Terminal" used to be synonymous with "computer". With the creation of operating systems like Windows and MacOS, computers became much easier to use and exploded in popularity! Your colleagues are almost always referring to the command line application when they say "Terminal".

3.2 Windows Setup

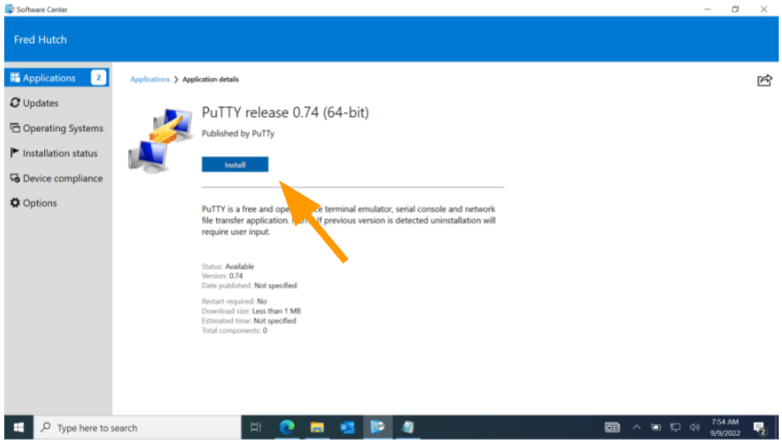
Click to view steps

You will need to install a Terminal application called PuTTY to connect to the Fred Hutch Cluster.

1. You should then see PuTTY available in the Software Center. Click "Install" and go through the Setup Wizard.



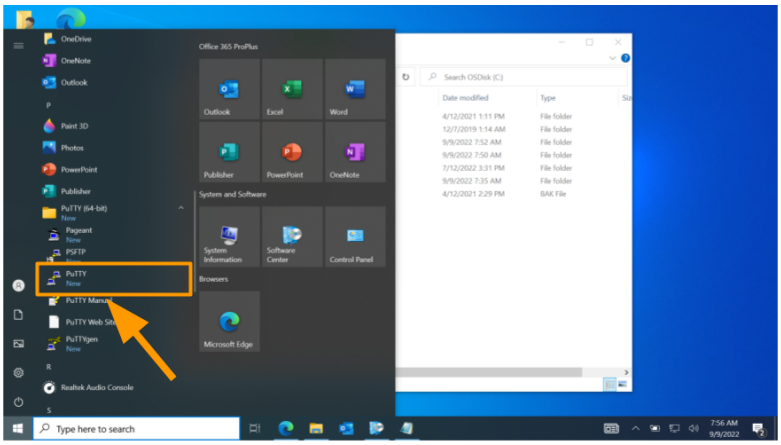
Screenshot of the Software Center application. CC-by hutchdatascience.org



Screenshot of the Software Center PuTTY installation page. CC-by hutchdatascience.org

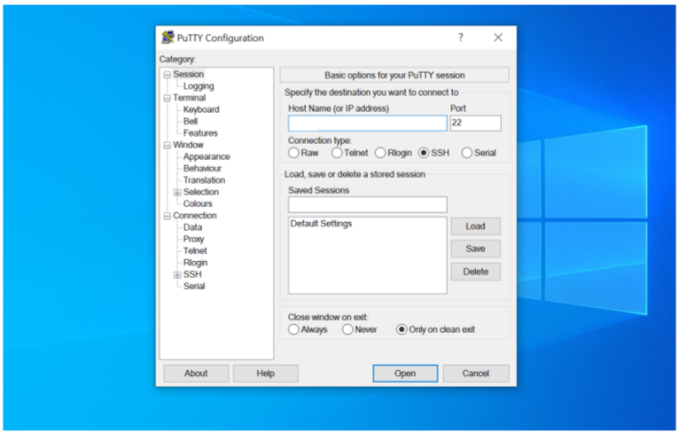
You can also install PuTTY manually if you don't see it in the Software Center.

2. PuTTY should now be available in your applications. Click on PuTTY to open.



Screenshot of the Windows application menu. CC-by hutchdatascience.org

3. You should now see the PuTTY Configuration menu.



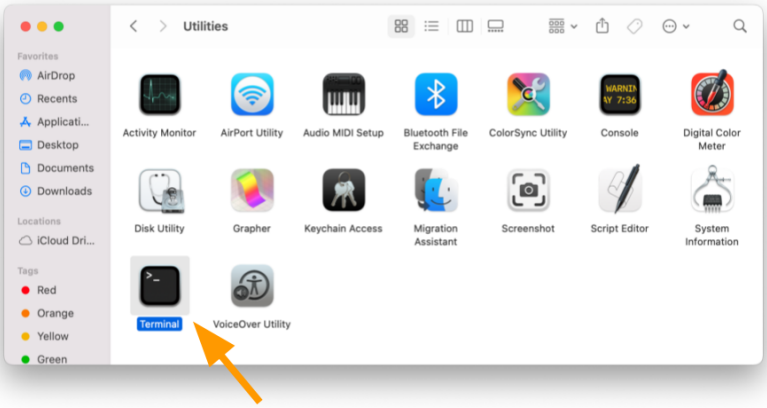
Screenshot of the PuTTY Configuration menu. CC-by hutchdatascience.org

3.3 Mac Setup

Click to view steps

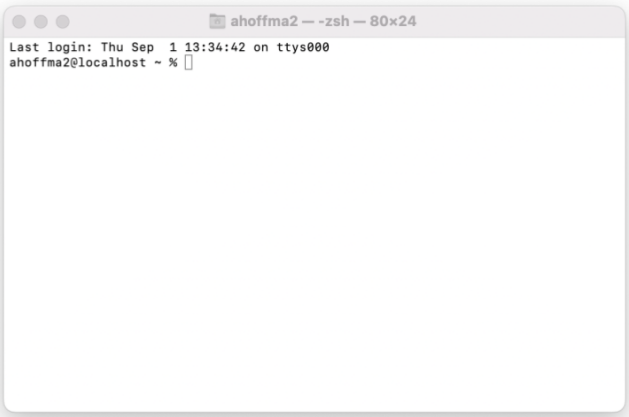
Mac machines come with a Terminal installed.

1. Go to Finder > Applications > Utilities > Terminal and double-click.



Screenshot of the Mac Utilities directory. CC-by hutchdatascience.org

2. Your Terminal should look like this:



Screenshot of the Mac Terminal. CC-by hutchdatascience.org

3.4 Linux Setup

Click to view steps

The commonly used Linux distribution, Ubuntu, already comes with a Terminal installed.

1. Press ctrl + alt + T. Your open Terminal window should look like this:

[SCREENSHOT]

1. Update the Terminal and prepare it for connecting to the cluster by running:

```
sudo apt install openssh-server
```

Enter your password and enter `Y` when prompted.

Chapter 4

Logging In

Now that you have your Terminal application ready, you want to connect to the cluster. You will do this using a method called SSH, which stands for "Secure SHell".

4.1 What is `SSH`?

SSH is a secure way to remotely connect to another computer or network of computers. In other words, SSH helps us protect your data and the data on the Fred Hutch cluster through authentication.

Hostname

The hostname is the name, or label, assigned to a computer in a network. We are connecting to hostname `rhino.fhcrc.org` or `rhino` for short.

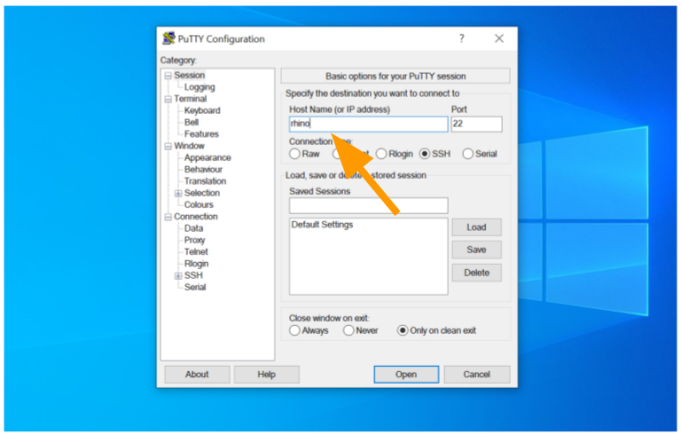
Before moving on, you will need to connect to the Fred Hutch wifi network, a networked ethernet jack, or the Fred Hutch VPN. This is the first layer of security.

The next set of steps are specific to your operating system.

4.2 Windows Login

Click to view steps

1. Go to the PuTTY Configuration menu. Under "Host Name" type **rhino** and click "Open".



Screenshot of the PuTTY Configuration menu.

CC-by hutchdatascience.org

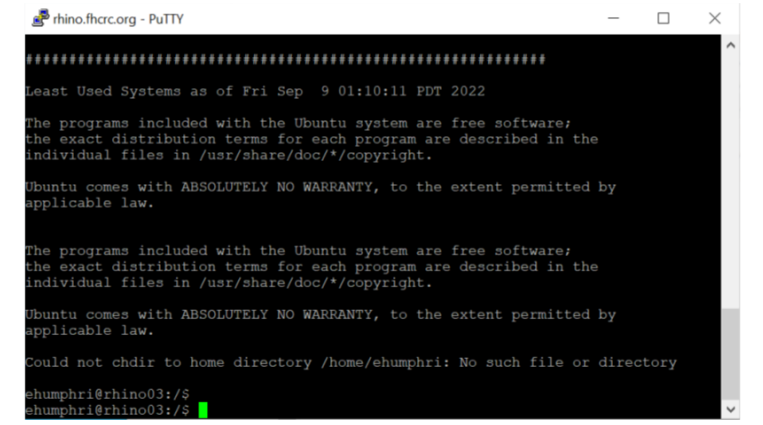
2. You will be prompted to login. Type in your HutchNetID (e.g., `jsmith3`).

Screenshot of the PuTTY login prompt for `rhino`.

CC-by hutchdatascience.org

3. Enter your password. No∗ or symbols will show up, so type it in carefully!

4. You are now logged in! There should be a login message, with your name at the bottom.



Screenshot of the Fred Hutch cluster login message. CC-by hutchdatascience.org

Congratulations! You are now logged in to the Fred Hutch cluster!

4.3 Mac Login

Click to view steps

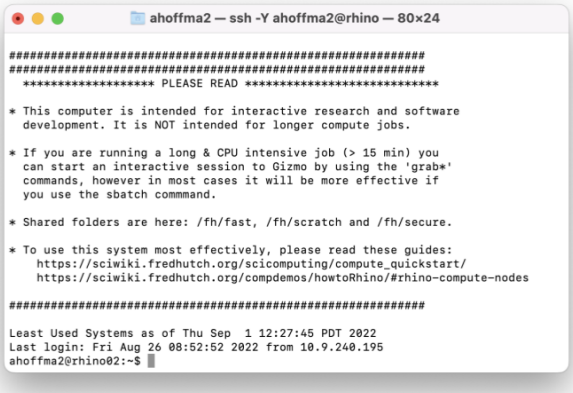
1. Type the following commands, substituting in your HutchNet ID:

   ```
   ssh -Y HutchID@rhino
   ```

2. You will see a message that looks like `The authenticity of host 'rhino (XXX.XXX.XX.XX)' can't be established.` Type in `yes` and hit enter.

3. Enter your password. No* or symbols will show up, so type it in carefully!

4. You are now logged in! There should be a login message, with your name at the bottom.



Screenshot of the Fred Hutch cluster login message. CC-by hutchdatascience.org

Congratulations! You are now logged in to the Fred Hutch cluster!

4.4 Linux Login

Click to view steps

Congratulations! You are now logged in to the Fred Hutch cluster!

Chapter 5

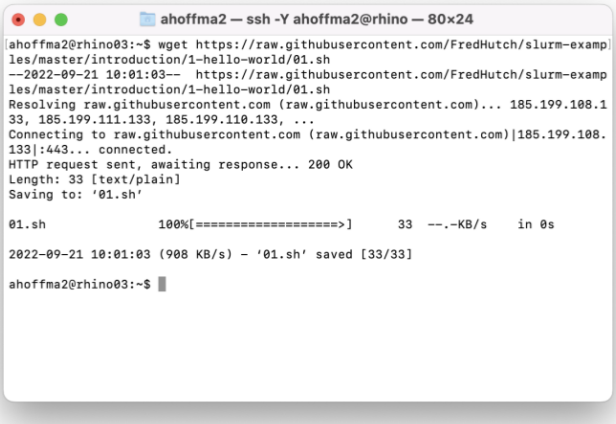
Submit your first job

The Fred Hutch cluster uses Slurm to organize and prioritize jobs. Instead of running commands directly on rhino, you will submit a script (a snippet of directions) to tell Slurm what to do.

5.1 Download the script

We can use the `wget` command to download a script from GitHub. This means we don't have to write the script from scratch. Copy and paste the following into the terminal, and hit return:

```
wget https://raw.githubusercontent.com/FredHutch/slurm-examples/master/introduction/1-hello-world
```



Screenshot of wget command output.

CC-by hutchdatascience.org

5.2 Confirm the download

Let's confirm that we can see the file we just downloaded. We can use the `ls` (list files) command for this. Type `ls` and hit return. You should see the file `01.sh` in your home directory. The `.sh` ending means this is a script meant to run from the command line.

```
ls
```

Screenshot of ls command output.

CC-by hutchdatascience.org

5.3 Inspect the script

Let's next inspect the script. The `cat` command, followed by a file name, lists the entire contents of a specific file.

```
cat 01.sh
```

Screenshot of cat command output, the contents of the script.

CC-by hutchdatascience.org

- The first line of the script, `#!/bin/bash`, indicates that this is a command line or "bash" script.
- The second line is empty, and the third line, `echo "Hello, World"` means that the computer will "echo", or print out, "Hello, World".

5.4 Submit the script

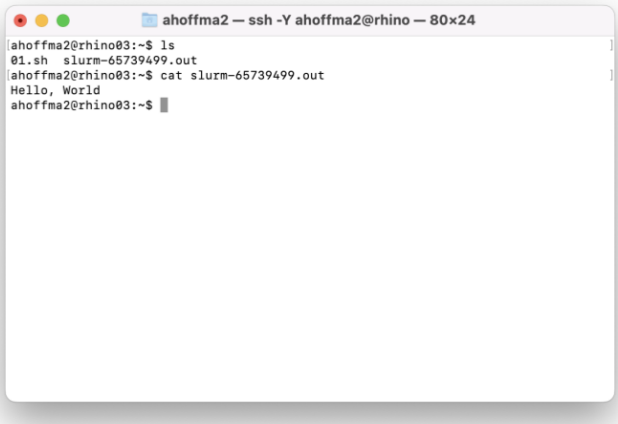
We use the `sbatch` command to submit a script and start running a job on the cluster. Copy the following and hit return. You should see a message like "`Submitted batch job 12345678`". Your number will vary because this is a unique job identifier.

```
sbatch 01.sh
```

5.5 Check the output

Type `ls` again. You should now see a log file like `slurm-12345678.out` listed alongside your script `01.sh`. Let's use `cat` to inspect the output in the log file. We should see our message has been printed!

```
cat slurm-<your-number-here>.out
```



Screenshot of cat command output, the contents of the log file.

CC-by hutchdatascience.org

`ls`

This command lists the files in the current directory.

`cat` *filename*

This command prints the contents of a specific file .

`sbatch` *filename.sh*

This command submits a job to the cluster with instructions specified in .sh

Chapter 6

File upload and download

Exchanging files with the cluster is very important. You can imagine scenarios where:

- You want to download log files or output files
- You want to upload a custom `.sh` script file that you wrote on your laptop
- You want to upload other files

In this course, upload and download of files is performed using Cyberduck. Cyberduck is a tool that lets us connect to the cluster securely, browse files, and transfer files securely.

6.1 Download Cyberduck

Download the latest version of Cyberduck here.

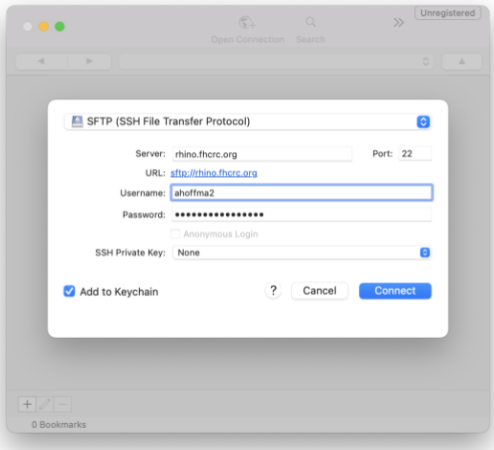
Note that the version of Cyberduck in the Software Center or Self Service might not be current, causing compatibility issues with some operating systems.

6.2 Create Connection

Launch Cyberduck and click on "Open Connection".

- From the dropdown menu, select "SFTP (SSH File Transfer Protocol)"
- For Server, type "rhino.fhcrc.org"
- Fill in your HutchNetID for Username and fill in your password

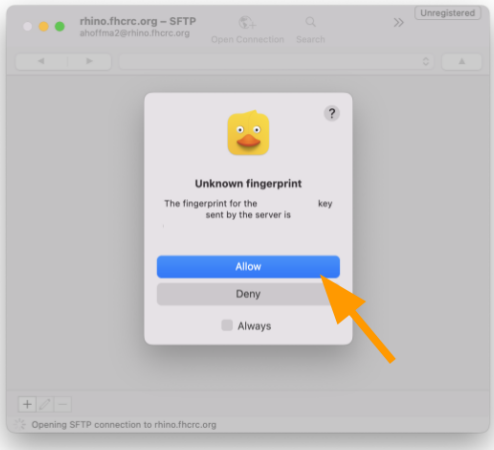
Click “Connect”



Screenshot of the Cyberduck “Open Connection” configuration.

CC-by hutchdatascience.org

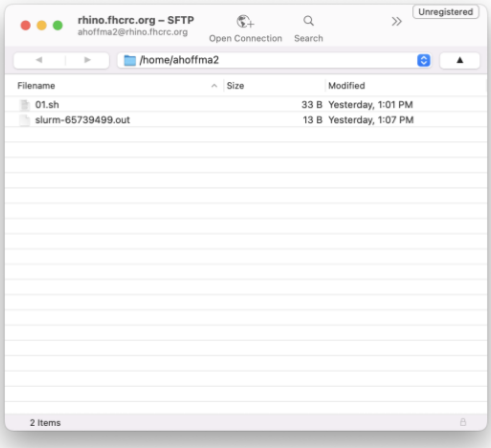
Click “Allow”. You can also check the box to indicate “Always”.



Screenshot of server fingerprint prompt.

CC-by hutchdatascience.org

You should see your script file “`01.sh`” and the log file.

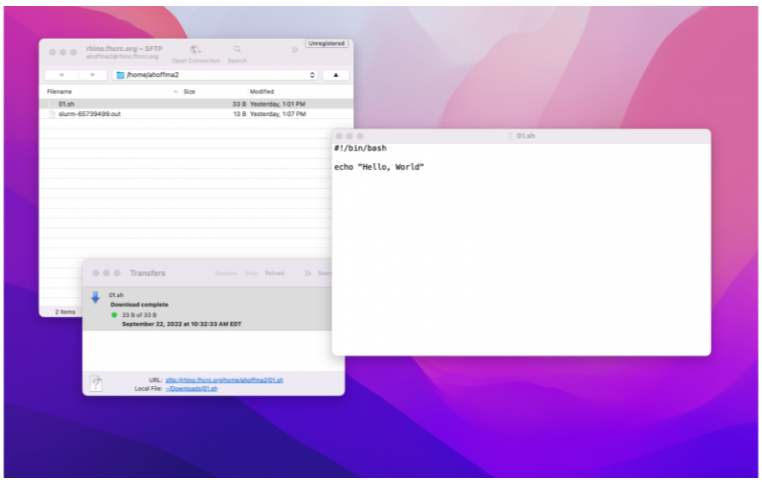


Screenshot of files on the cluster, viewed through Cyberduck.

CC-by hutchdatascience.org

6.3 Download and edit the script

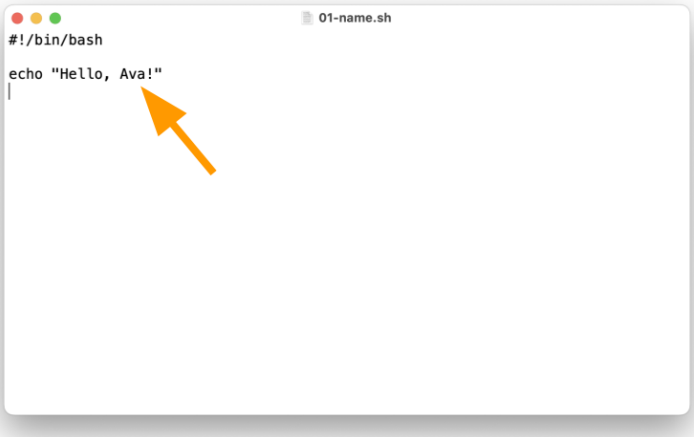
- Right click on "`01.sh`" and select "Download"
- You will see a "Transfers" prompt open, and the `01.sh` file should now appear in your Downloads folder
- Open the `01.sh` file



Screenshot of files, Transfers window, and the downloaded script.

CC-by hutchdatascience.org

Edit the message to include your name and save the file. Rename the file `01-name.sh`

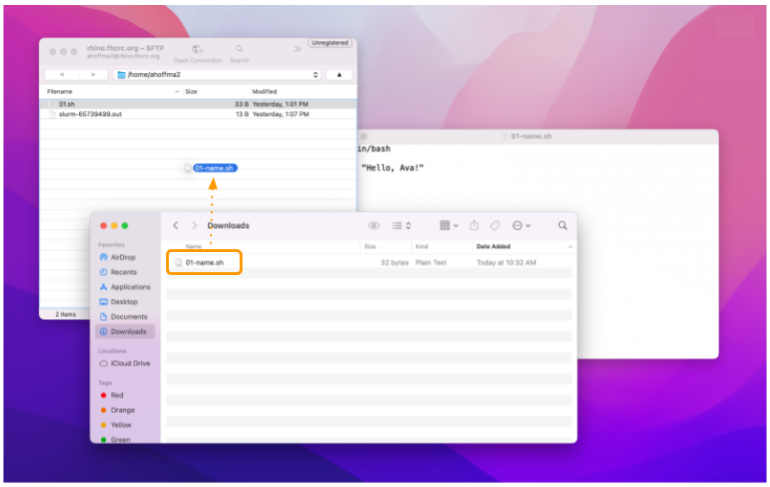


Screenshot of the script, edited with a name.

CC-by hutchdatascience.org

6.4 Upload the new script

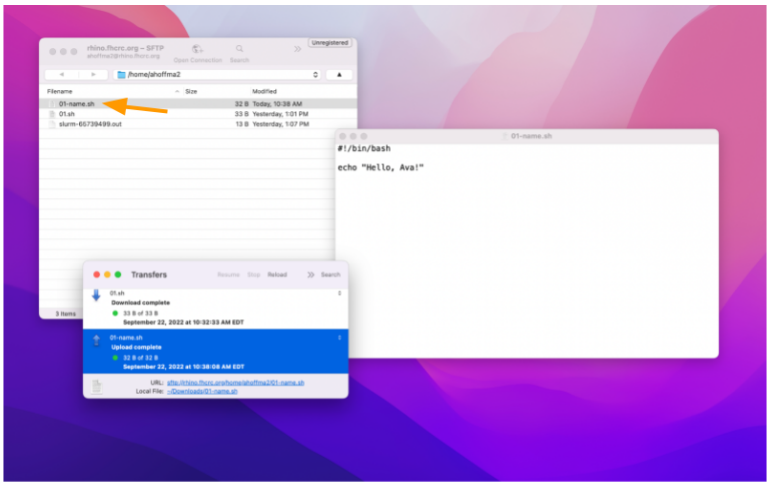
From your Downloads folder, simply drag the file to Cyberduck.



Screenshot of Downloads folder with edited script being dragged over to files via Cyberduck.

CC-by hutchdatascience.org

You should now see the new script among your cluster files.



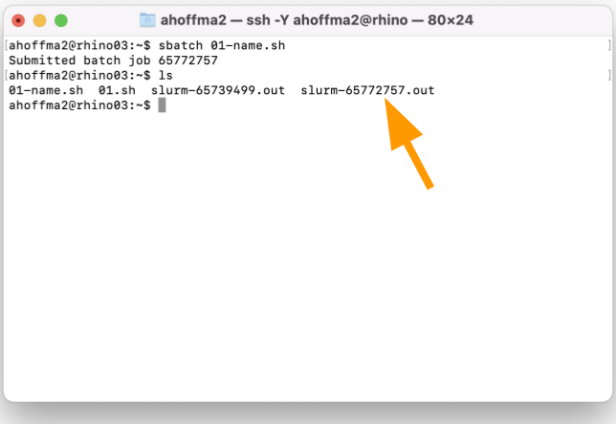
Screenshot of edited script among cluster files visible on Cyberduck.

CC-by hutchdatascience.org

6.5 Run the new script

Return to your Terminal. Submit a job with your new script by running the following. When you type `ls` you should see a new log file!

```
sbatch 01-name.sh
```



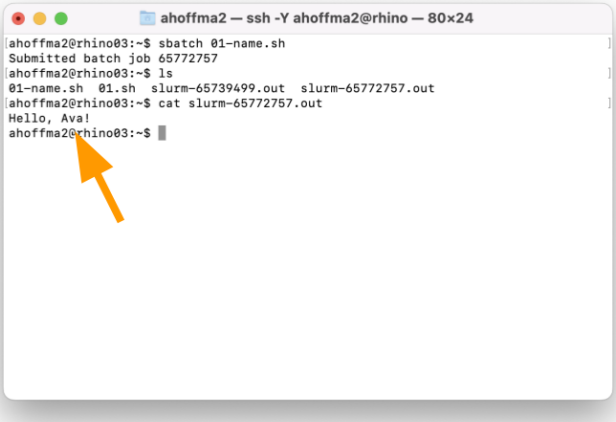
Screenshot showing new log file present on the cluster.

CC-by hutchdatascience.org

The job numbers included in log file names generally increase in number. The greater the number, the more recently the job was run.

Use the `cat` command to inspect the log. The message should show the new text that you added!

```
cat slurm-<your-number-here>.out
```



Screenshot of edited script message output using `cat` command.

CC-by hutchdatascience.org

Chapter 7

Interactive cluster

Assessment

Chapter 8

Cluster 101 Self Test

Appendix

Chapter 9

Where to get help

We want to help! Here are some ways you can get help for your work on the cluster.

Submit a Ticket

Submitting a good ticket helps the SciComp Team address your needs quickly and efficiently. We suggest you submit the following in a ticket:

1.
2.
3.

Visit the SciWiki

The SciWiki Scientific Computing page is full of useful tips and guides.

Chapter 10

Provide Feedback

Please submit an issue at our GitHub repository. You can also click the edit button on the top of the page in question.

Chapter 11

FAQ / Troubleshooting

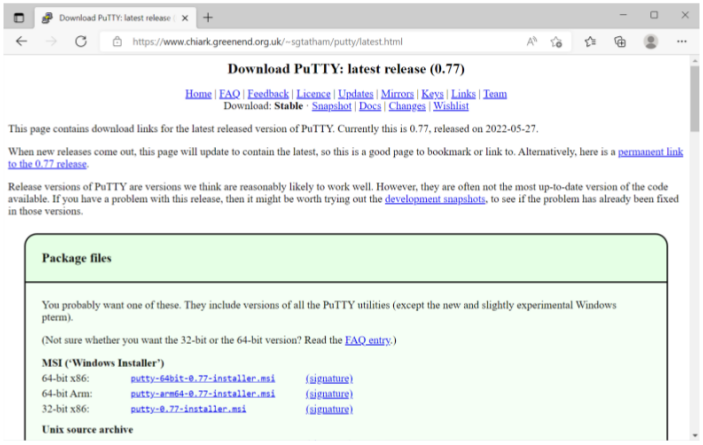
11.1 FAQ

Here are some questions you might have.

11.1.1 How can I manually install PuTTY?

Click to view steps

1. Click here to install the latest version of PuTTY. You will choose the 64-bit x86 installation with few exceptions.



Screenshot of the PuTTY download page. CC-by hutchdatascience.org

2. Click through to install via the Setup Wizard.

11.2 Troubleshooting

Here are some issues you might encounter.

`ssh: Could not resolve hostname rhino: nodename nor servname provided, or not known`

Click to view steps

This error means that your computer is having trouble connecting to rhino. Ensure one of the following is true:

1. You are connected to the Fred Hutch wifi network on campus.
2. You are connected to the Fred Hutch VPN
3. You are plugged into an ethernet cable on campus that taps into the Fred Hutch network. Note that not all ethernet wall jacks have this capability, so try another jack if you are having trouble. Please email the IT helpdesk and include your office number and the number on the jack if you find a jack that isn't working.

`ssh: connect to host rhino port 22: Undefined error: 0`

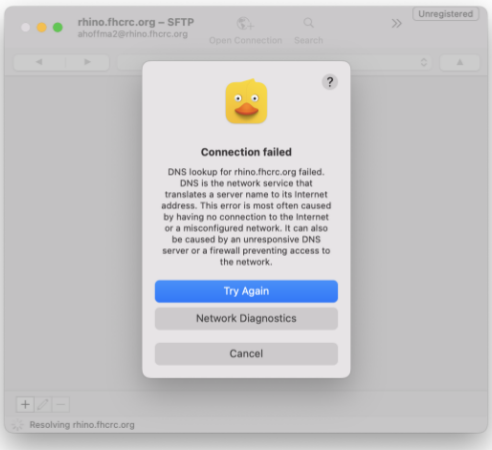
Click to view steps

This likely indicates a disruption to your internet connection and/or VPN. Ensure you are connected to the internet and connected to the Fred Hutch network on campus or the VPN.

`Connection failed` message in Cyberduck

Click to view steps

This likely indicates a disruption to your internet connection and/or VPN. Ensure you are connected to the internet and connected to the Fred Hutch network on campus or the VPN.



Screenshot of the Cyberduck "Connection failed" message.

CC-by hutchdatascience.org

About the Authors

These credits are based on our course contributors table guidelines.

Credits	Names
Pedagogy	
Lead Content Instructor(s)	FirstName LastName
Lecturer(s) (include chapter name/link in parentheses if only for specific chapters) - make new line if more than one chapter involved	Delivered the course in some way - video or audio
Content Author(s) (include chapter name/link in parentheses if only for specific chapters) - make new line if more than one chapter involved	If any other authors besides lead instructor
Content Contributor(s) (include section name/link in parentheses) - make new line if more than one section involved	Wrote less than a chapter
Content Editor(s)/Reviewer(s)	Checked your content
Content Director(s)	Helped guide the content direction
Content Consultants (include chapter name/link in parentheses or word "General") - make new line if more than one chapter involved	Gave high level advice on content
Acknowledgments	Gave small assistance to content but not to the level of consulting
Production	
Content Publisher(s)	Helped with publishing platform
Content Publishing Reviewer(s)	Reviewed overall content and aesthetics on publishing platform

Credits	Names
Technical	
Course Publishing Engineer(s)	Helped with the code for the technical aspects related to the specific course generation
Template Publishing Engineers	Candace Savonen, Carrie Wright
Publishing Maintenance Engineer	Candace Savonen
Technical Publishing Stylists	Carrie Wright, Candace Savonen
Package Developers (ottrpal) Candace Savonen, John Muschelli, Carrie Wright	
Art and Design	
Illustrator(s)	Created graphics for the course
Figure Artist(s)	Created figures/plots for course
Videographer(s)	Filmed videos
Videography Editor(s)	Edited film
Audiographer(s)	Recorded audio
Audiography Editor(s)	Edited audio recordings
Funding	
Funder(s)	Institution/individual who funded course including grant number
Funding Staff	Staff members who help with funding

```
## - Session info ---------------------------------------------------------------
##  setting  value
##  version  R version 4.0.2 (2020-06-22)
##  os       Ubuntu 20.04.3 LTS
##  system   x86_64, linux-gnu
##  ui       X11
##  language (EN)
##  collate  en_US.UTF-8
##  ctype    en_US.UTF-8
##  tz       Etc/UTC
##  date     2022-09-22
##
## - Packages -------------------------------------------------------------------
##  package     * version    date       lib source
##  assertthat    0.2.1      2019-03-21 [1] RSPM (R 4.0.3)
##  bookdown      0.24       2022-02-15 [1] Github (rstudio/bookdown@88bc4ea)
```

```
##  callr         3.4.4      2020-09-07 [1] RSPM (R 4.0.2)
##  cli           2.0.2      2020-02-28 [1] RSPM (R 4.0.0)
##  crayon        1.3.4      2017-09-16 [1] RSPM (R 4.0.0)
##  desc          1.2.0      2018-05-01 [1] RSPM (R 4.0.3)
##  devtools      2.3.2      2020-09-18 [1] RSPM (R 4.0.3)
##  digest        0.6.25     2020-02-23 [1] RSPM (R 4.0.0)
##  ellipsis      0.3.1      2020-05-15 [1] RSPM (R 4.0.3)
##  evaluate      0.14       2019-05-28 [1] RSPM (R 4.0.3)
##  fansi         0.4.1      2020-01-08 [1] RSPM (R 4.0.0)
##  fs            1.5.0      2020-07-31 [1] RSPM (R 4.0.3)
##  glue          1.6.1      2022-01-22 [1] CRAN (R 4.0.2)
##  htmltools     0.5.0      2020-06-16 [1] RSPM (R 4.0.1)
##  knitr         1.33       2022-02-15 [1] Github (yihui/knitr@a1052d1)
##  lifecycle     1.0.0      2021-02-15 [1] CRAN (R 4.0.2)
##  magrittr      2.0.2      2022-01-26 [1] CRAN (R 4.0.2)
##  memoise       1.1.0      2017-04-21 [1] RSPM (R 4.0.0)
##  pkgbuild      1.1.0      2020-07-13 [1] RSPM (R 4.0.2)
##  pkgload       1.1.0      2020-05-29 [1] RSPM (R 4.0.3)
##  prettyunits   1.1.1      2020-01-24 [1] RSPM (R 4.0.3)
##  processx      3.4.4      2020-09-03 [1] RSPM (R 4.0.2)
##  ps            1.3.4      2020-08-11 [1] RSPM (R 4.0.2)
##  purrr         0.3.4      2020-04-17 [1] RSPM (R 4.0.3)
##  R6            2.4.1      2019-11-12 [1] RSPM (R 4.0.0)
##  remotes       2.2.0      2020-07-21 [1] RSPM (R 4.0.3)
##  rlang         0.4.10     2022-02-15 [1] Github (r-lib/rlang@f0c9be5)
##  rmarkdown     2.10       2022-02-15 [1] Github (rstudio/rmarkdown@02d3c25)
##  rprojroot     2.0.2      2020-11-15 [1] CRAN (R 4.0.2)
##  sessioninfo   1.1.1      2018-11-05 [1] RSPM (R 4.0.3)
##  stringi       1.5.3      2020-09-09 [1] RSPM (R 4.0.3)
##  stringr       1.4.0      2019-02-10 [1] RSPM (R 4.0.3)
##  testthat      3.0.1      2022-02-15 [1] Github (R-lib/testthat@e99155a)
##  usethis       2.1.5.9000 2022-02-15 [1] Github (r-lib/usethis@57b109a)
##  withr         2.3.0      2020-09-22 [1] RSPM (R 4.0.2)
##  xfun          0.32       2022-08-10 [1] CRAN (R 4.0.2)
##  yaml          2.2.1      2020-02-01 [1] RSPM (R 4.0.3)
##
## [1] /usr/local/lib/R/site-library
## [2] /usr/local/lib/R/library
```

References