

RSPH Cluster/Command Line Intro

EPI Journal Club

Mar 22, 2024

Hannah Waddel

https://github.com/hbwddl/Epi_Journal_Club_Cluster/

Audience

- R user
- Problem situations:
 - Big dataset that won't fit in your computer (“Error: vector memory exhausted (limit reached?)”)
 - You have code that needs to run for hours/days and you'll need to leave your computer running for that entire time

Goals

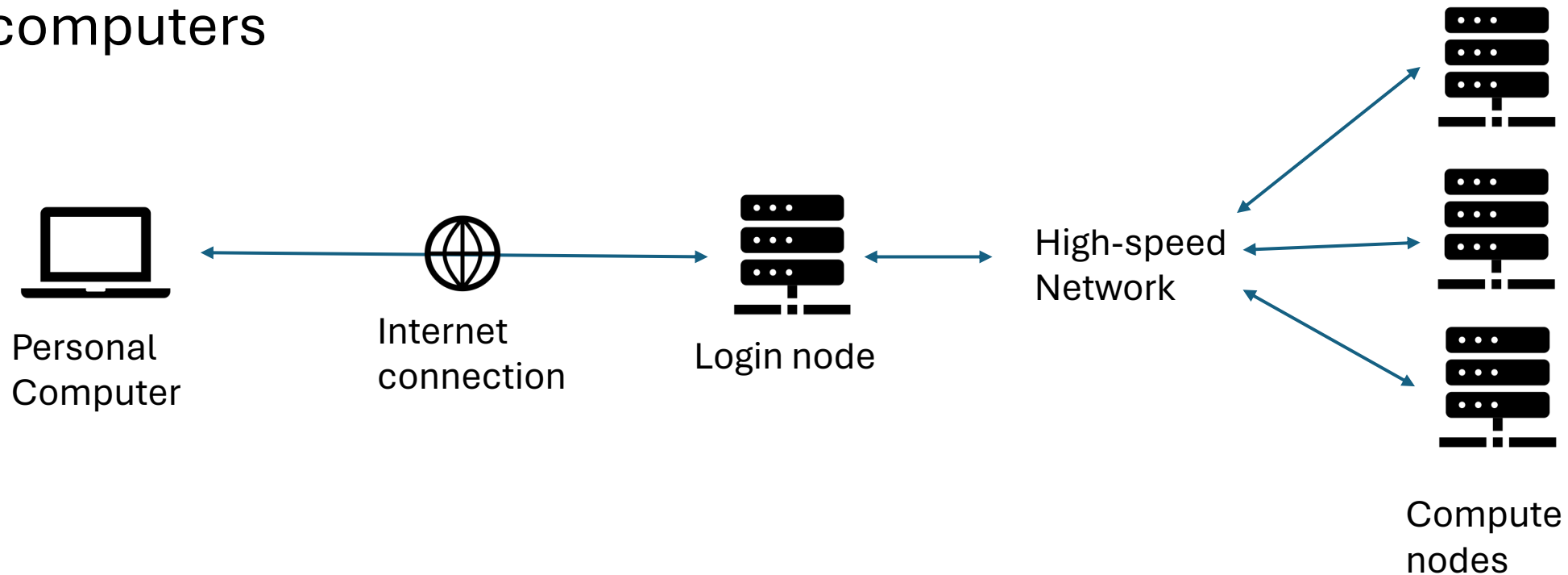
If you have an R script which you can run using the `source()` function, you will be able to move that script into the cluster, start it as a job in the cluster, and retrieve its results.

What I won't cover/further topics

- Parallel programming/packages
- Jupyter notebooks

What is a cluster?

- HPC (high performance computing) cluster
- Collection of nodes (computers) connected together into a network to enable computing power beyond the limits of personal computers



Accessing the cluster

- Connect to Emory VPN (Virtual Private Network)
 - Instructions at vpn.emory.edu
- VPN is a secure 'tunnel' to Emory's network
- Cluster account
 - Separate from Emory's account
 - Sponsored by RSPH faculty

Connect to the VPN

Accessing the cluster

- Interact with cluster through *terminal*
 - Text-based way to control a computer
- Mac: Terminal
- Windows: Powershell

Log in to the cluster

- `ssh hbwadde@clogin01.sph.emory.edu`
- Replace hbwadde with your netid (7 letter username)
- Enter password

Success!



```
hbwadde — hbwadde@clogin01:~ —  
Last login: Wed Mar 20 21:16:53 on ttys002  
hbwadde@BIOC02GC51HQ05P ~ % ssh hbwadde@clogin01.sph.emory.edu  
  
Welcome to the  
  
HPC  
  
High Performance Computing (HPC) Cluster  
  
*** AUTHORIZED USE ONLY ***  
  
-->>> DO NOT RUN APPLICATIONS ON THE LOGIN NODE <<----  
Please submit ALL computations, including small  
interactive ones, to compute nodes.  
  
Last login: Wed Mar 20 21:20:21 2024 from 10.110.72.4  
[hbwadde@clogin01 ~]$
```

Troubleshooting

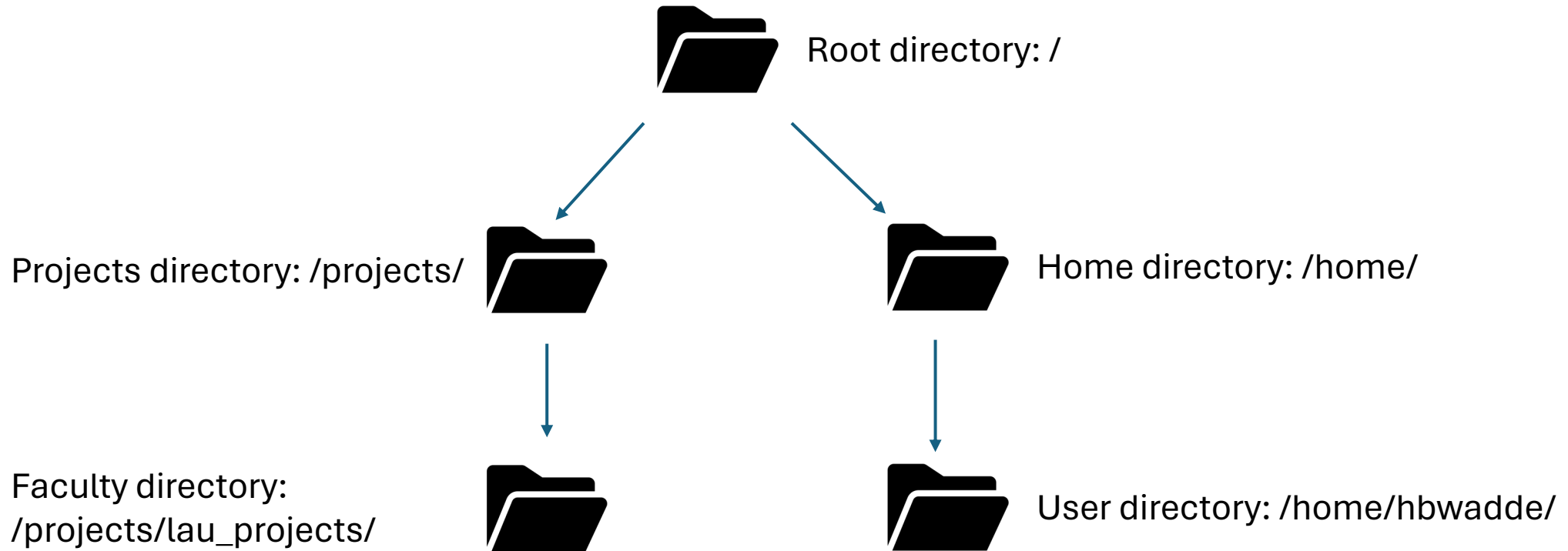
- Does nothing: Are you connected to the VPN?
- “ssh: Could not resolve hostname clogin01.sph.emory.edu: nodename nor servname provided, or not known”
 - Did you correctly spell address clogin01.sph.emory.edu
- “Permission denied, please try again.”
 - Username or password incorrect
- “Command not found”
 - Typo in the ssh command

In the cluster

- `pwd` : Print Working Directory
- `ls` : List files
- Up arrow: get previous command

```
hbwadde — hbwadde@clogin01:~ — ssh hbwadde@clogin01.sph.emory.edu — 62x20
[hwadde@clogin01 ~]$ pwd
/home/hbwadde
[hwadde@clogin01 ~]$ ls
BIOS-job-examples  meta_sim  rjags_4-4.tar.gz
bio.txt            Mortgage_Discrimination
JAGS-4.3.0.tar.gz  R
[hwadde@clogin01 ~]$
```

Cluster files/filepath



Home directory



Root directory: /

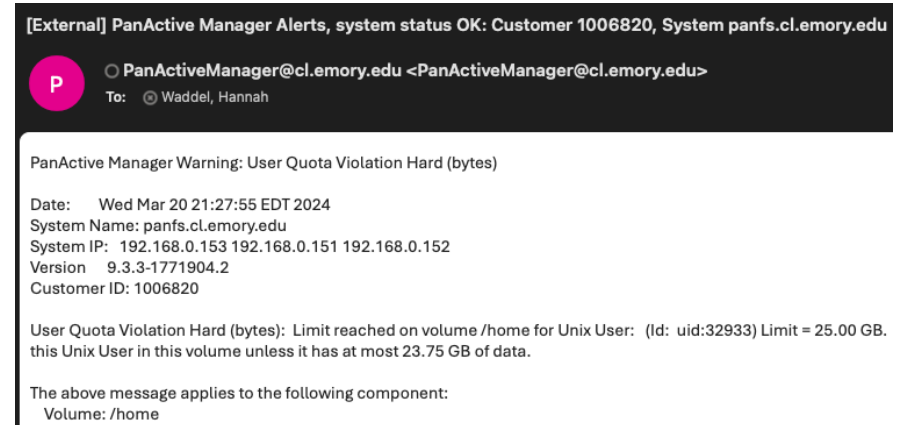


Home directory: /home/



User directory: /home/hbwadde/

- "Default" directory when you log in
- Only accessible to you
- Each user gets 25GB of space



Your job fails and you get a warning email if you exceed 25GB

Projects directory

Root directory: /



Projects directory: /projects/

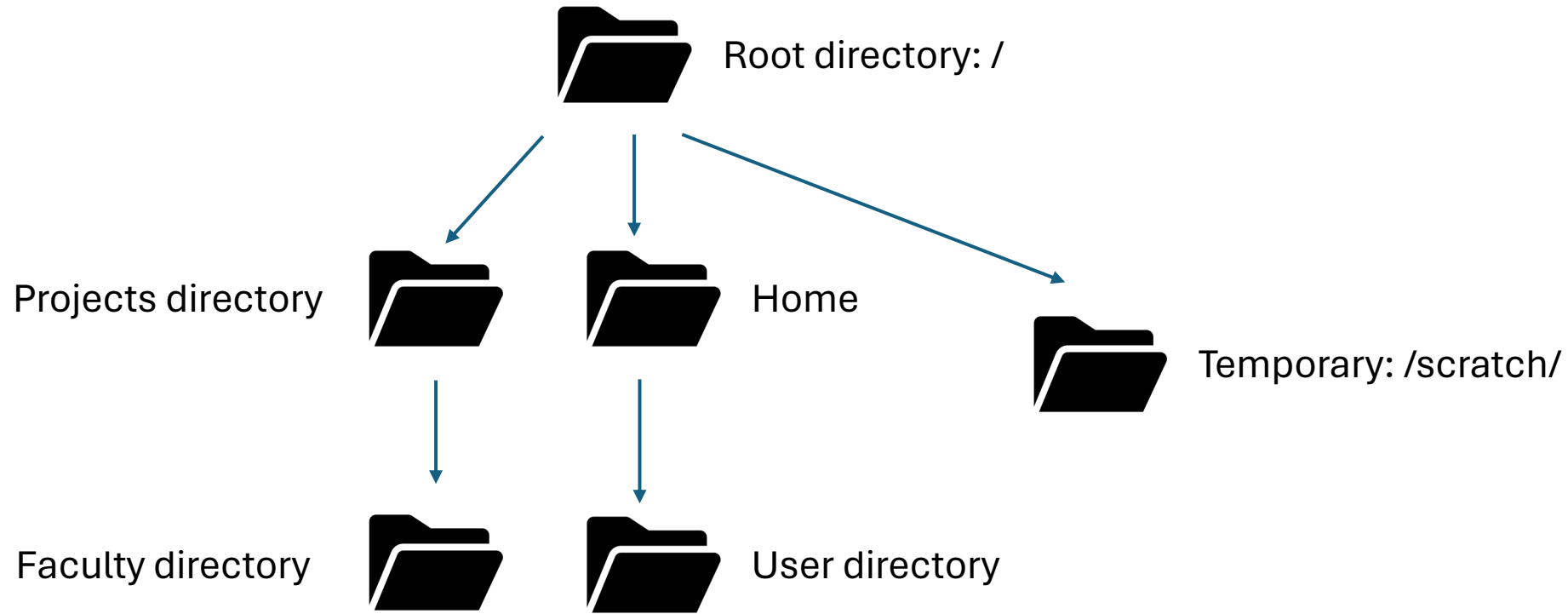


Faculty directory:
/projects/lau_projects/



- Receive access to this through a faculty member
- Each faculty member has 1 TB of storage, may have more

Cluster files/filepath



Scratch Directory



Root directory: /



Temporary: /scratch/

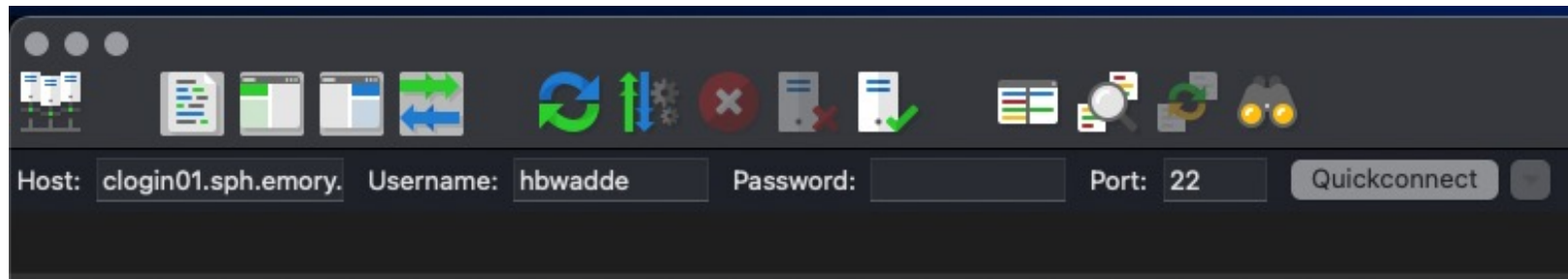
- If you need more space than 25GB, you can work in the /scratch/ directory
- Temporary directory: your files will be cleared after two weeks
 - Email Keven Haynes if you need more time

Moving files to and from the cluster

- Filezilla: FTP (File Transfer Protocol) program
 - Gives you a graphical/click and drag interface for the cluster
 - NOTE: This software is not supported by IT—if you email Keven about Filezilla he will not be able to help you.
-
- Files can also be transferred with the scp command
 - From the computer with the file you want to move, run:
`scp my_file.txt hbwadde@clogin01.sph.emory.edu:/home/hbwadde`

Filezilla: Connect to the cluster

- Make sure you are connected to the VPN
- Fill out the 'Quickconnect' toolbar
- Host: clogin01.sph.emory.edu
- Username: Your netid
- Password: Your cluster password
- Port: 22



Exercise: Move our example files to the cluster

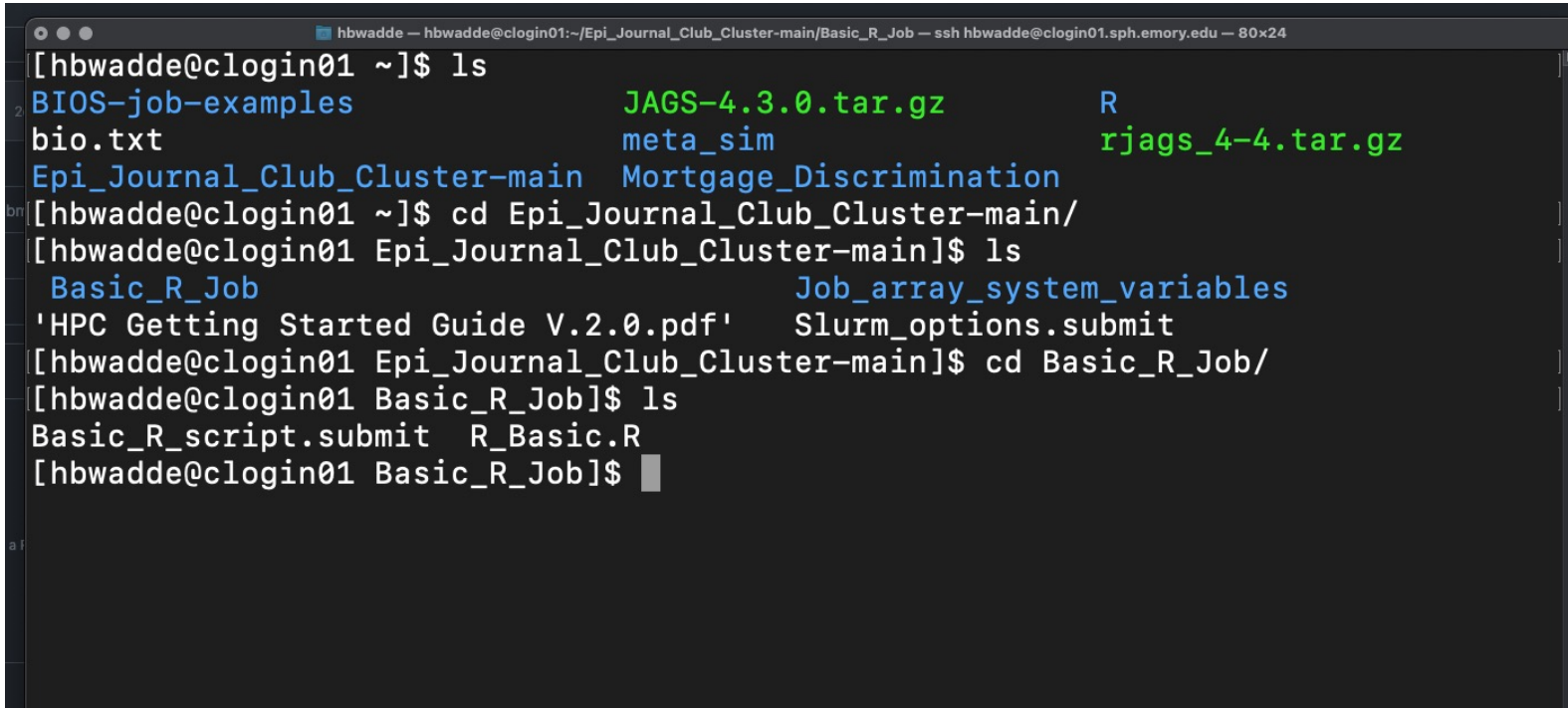
- https://github.com/hbwddl/Epi_Journal_Club_Cluster/
- Click-and-drag in Filezilla like it's finder/file explorer

Navigating the cluster

- `cd` : change directory
 - Needs an argument of which folder to change to
- `cd /projects/` : change to /projects/ directory
- `cd ..` : Change directory one step up
- `cd ~` : Change to home/default directory (/home/hbwadde/)

Navigating the cluster

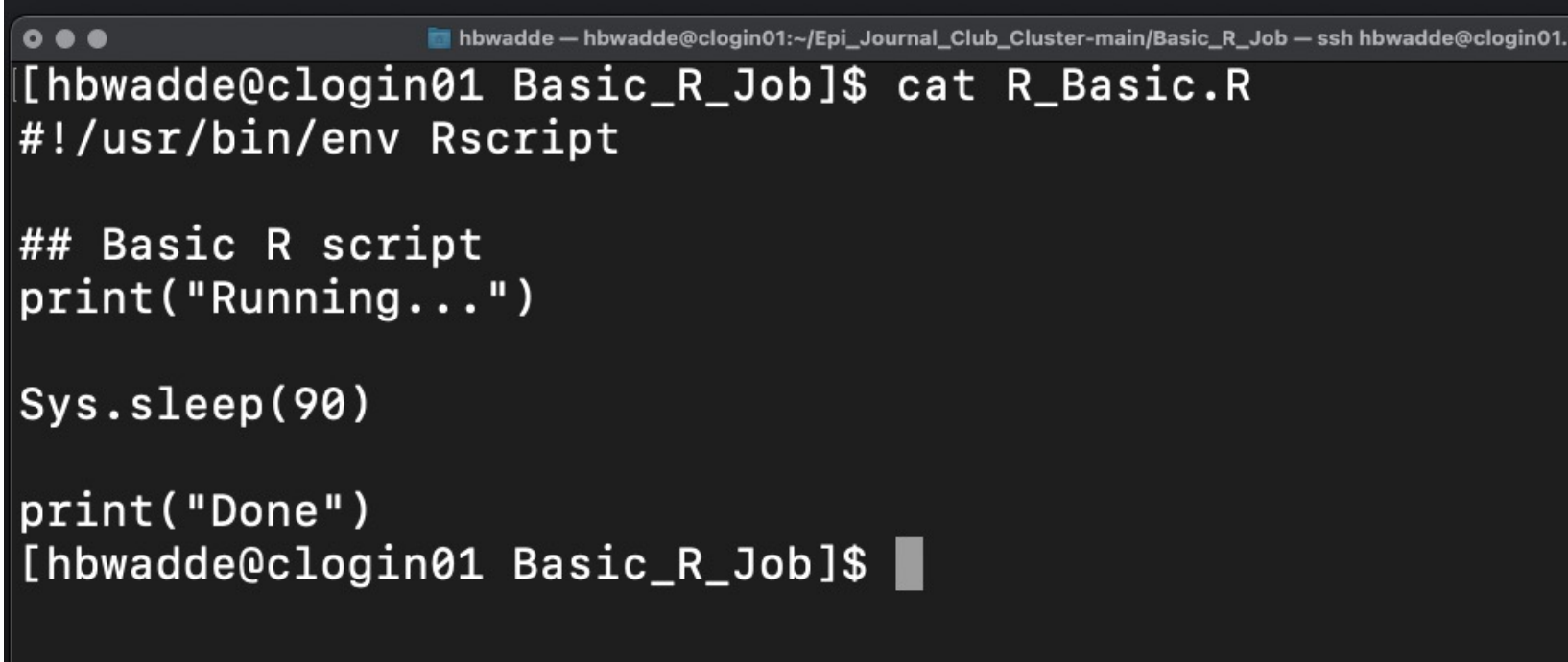
- The terminal will tab-complete for you if you type out a portion of a filename, folder name, or command
- Exercise: Move to your Basic_R_Job folder

A terminal window with a dark background and light-colored text. The window title bar shows the user 'hbwadde' and the current directory path. The terminal shows a series of commands and their outputs, demonstrating how to navigate through a file system using tab completion. The commands include 'ls' to list files, 'cd' to change directories, and another 'ls' to verify the current location. The files listed include various tar.gz archives, PDF guides, and R script files.

```
hbwadde — hbwadde@clogin01:~/Epi_Journal_Club_Cluster-main/Basic_R_Job — ssh hbwadde@clogin01.sph.emory.edu — 80x24
[hbwadde@clogin01 ~]$ ls
BIOS-job-examples      JAGS-4.3.0.tar.gz      R
bio.txt                meta_sim                rjags_4-4.tar.gz
Epi_Journal_Club_Cluster-main  Mortgage_Discrimination
[hbwadde@clogin01 ~]$ cd Epi_Journal_Club_Cluster-main/
[hbwadde@clogin01 Epi_Journal_Club_Cluster-main]$ ls
Basic_R_Job            Job_array_system_variables
'HPC Getting Started Guide V.2.0.pdf'  Slurm_options.submit
[hbwadde@clogin01 Epi_Journal_Club_Cluster-main]$ cd Basic_R_Job/
[hbwadde@clogin01 Basic_R_Job]$ ls
Basic_R_script.submit  R_Basic.R
[hbwadde@clogin01 Basic_R_Job]$
```

Viewing files

- `cat filename` : prints out the contents of a file
- `head filename` : prints out the first 10 lines of a file
- `tail filename` : prints out the last 10 lines of a file

A terminal window with a dark background and light text. The window title bar shows 'hbwadde — hbwadde@clogin01:~/Epi_Journal_Club_Cluster-main/Basic_R_Job — ssh hbwadde@clogin01'. The terminal content shows the command '[hbwadde@clogin01 Basic_R_Job]\$ cat R_Basic.R' followed by the output of the file: '#!/usr/bin/env Rscript', '## Basic R script', 'print("Running...")', 'Sys.sleep(90)', 'print("Done")', and the prompt '[hbwadde@clogin01 Basic_R_Job]\$' with a cursor.

```
hbwadde — hbwadde@clogin01:~/Epi_Journal_Club_Cluster-main/Basic_R_Job — ssh hbwadde@clogin01
[hbwadde@clogin01 Basic_R_Job]$ cat R_Basic.R
#!/usr/bin/env Rscript

## Basic R script
print("Running...")

Sys.sleep(90)

print("Done")
[hbwadde@clogin01 Basic_R_Job]$
```

SLURM: Submitting cluster jobs

- The login node is for logging in and small file movement tasks only
 - Not powerful enough to run software
- SLURM (Simple Linux Utility for Resource Management)
 - The cluster runs the Linux operating system (every cluster does), specifically the Rocky Linux operating system
- SLURM is the workload manager which allocates the compute nodes to different users for computing ‘jobs’
- SLURM runs your code on the compute nodes
- Restaurant host analogy (Large party takes a while to seat, small party gets seated quickly—same with large jobs and small jobs)

Cluster organization

- The compute nodes in the cluster, which actually do the heavy lifting, are divided into partitions
- These are based on how long you are able to run a job on them
 - month-long-cpu
 - week-long-cpu
 - day-long-cpu
 - short-cpu (30 minutes)
- Faculty members may give you access to their own partitions

SLURM Commands

- Slurm has its own set of terminal commands which you use to submit and control jobs
- Jobs are designed and submitted with a 'batch file'
- A batch file contains instructions for how your cluster job should be run

Batch files

```
hbwadde — hbwadde@clogin01:~/Epi_Journal_Club_Cluster-main/Basic_R_Job — ssh hbwadde@clogin01.sph.emo
[hbwadde@clogin01 Basic_R_Job]$ cat Basic_R_script.submit
#!/bin/bash
#SBATCH --partition=short-cpu
#SBATCH --job-name=R_basic
#SBATCH --error=R_basic.%J.err
#SBATCH --output=R_basic.%J.out

module load R

Rscript R_basic.R
[hbwadde@clogin01 Basic_R_Job]$
```

Submitting a batch file: sbatch command

- sbatch batchfile

```
[hbwadde@clogin01 Basic_R_Job]$ sbatch Basic_R_script.submit  
Submitted batch job 18326121  
[hbwadde@clogin01 Basic_R_Job]$
```

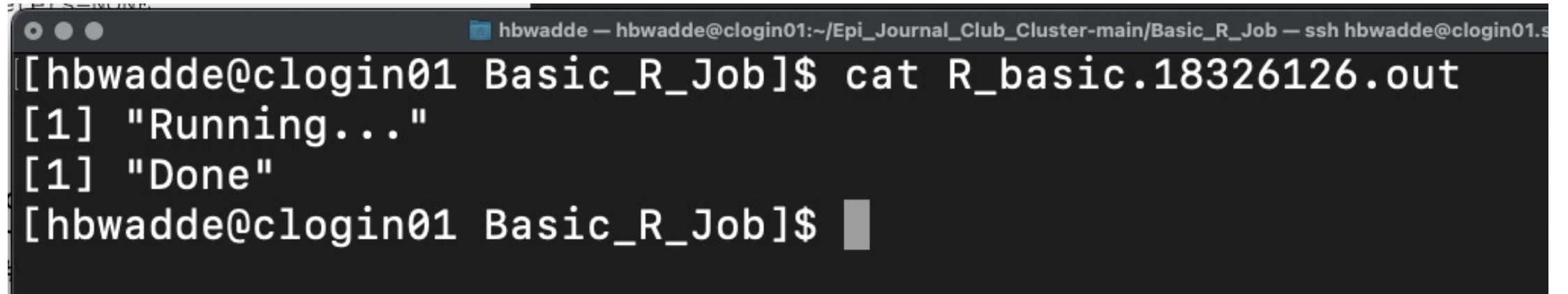
Check status of jobs/partitions: squeue

- squeue : shows every job on the cluster
- squeue -u hbwadde
 - -u is called a 'flag'
 - Tells squeue "show me jobs submitted by user hbwadde"

```
etars=NONE
hbwadde — hbwadde@clogin01:~/Epi_Journal_Club_Cluster-main/Basic_R_Job — ssh hbwadde@clogin01.sph.emory.edu — 85x24

[hbwadde@clogin01 Basic_R_Job]$ sbatch Basic_R_script.submit
Submitted batch job 18326124
[hbwadde@clogin01 Basic_R_Job]$ squeue -u hbwadde
      JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
      18326124 short-cpu   R_basic   hbwadde  R        0:03        1 node22
[hbwadde@clogin01 Basic_R_Job]$
```

Check the results

A terminal window with a dark background and light text. The title bar at the top shows 'hbwadde — hbwadde@clogin01:~/Epi_Journal_Club_Cluster-main/Basic_R_Job — ssh hbwadde@clogin01.s'. The terminal content shows a command being executed and its output.

```
[hbwadde@clogin01 Basic_R_Job]$ cat R_basic.18326126.out  
[1] "Running..."  
[1] "Done"  
[hbwadde@clogin01 Basic_R_Job]$
```

Stop a job: scancel

- `scancel jobid`

```
[hbwadde@clogin01 Basic_R_Job]$ sbatch Basic_R_script.submit  
Submitted batch job 18326133  
[hbwadde@clogin01 Basic_R_Job]$ scancel 18326133
```

- `scancel -u hbwadde` : cancels all jobs by user hbwadde

SLURM Arrays

- If you have to submit multiple jobs that do the same thing, use an array

```
hbwadde — hbwadde@clogin01:~/Epi_Journal_Club_Cluster-main/Job_array_system_variables — ssh hbwadde@clogin01.sph.emory.edu — 85x24
[hbwadde@node24 Job_array_system_variables]$ cat Job_array_system_variables.submit
#!/bin/bash
#SBATCH --partition=short-cpu
#SBATCH --job-name=R_basic
#SBATCH --error=R_basic.%J.err
#SBATCH --output=R_basic.%J.out
#SBATCH --array=1-4

module load R

Rscript R_script_array_system_variables.R
[hbwadde@node24 Job_array_system_variables]$
```

SLURM Arrays

```
hbwadde — hbwadde@clogin01:~/Epi_Journal_Club_Cluster-main/Job_array_system_variables — ssh hbwadde@clogin01.sph.emory.edu — 85x24
[hbwadde@node24 Job_array_system_variables]$ sbatch Job_array_system_variables.submit

Submitted batch job 18326140
[hbwadde@node24 Job_array_system_variables]$ squeue -u hbwadde
```

JOBID	PARTITION	NAME	USER	ST	TIME	NODES	NODELIST(REASON)
18326135	interacti	bash	hbwadde	R	8:20	1	node24
18326140_1	short-cpu	R_basic	hbwadde	R	0:05	1	node23
18326140_2	short-cpu	R_basic	hbwadde	R	0:05	1	node23
18326140_3	short-cpu	R_basic	hbwadde	R	0:05	1	node23
18326140_4	short-cpu	R_basic	hbwadde	R	0:05	1	node23

```
[hbwadde@node24 Job_array_system_variables]$
```


System variables

- When you are submitting an array job, there are values from Slurm that you may want to get
 - Get the number in the array to set random seed, for example
- These can be accessed from within R with the `Sys.getenv()` command
- I've provided an example/template

R script with system variables (template)

```
[hbwadde@node24 Job_array_system_variables]$ cat R_script_array_system_variables.R
#!/usr/bin/env Rscript

## R script which uses SLURM variables
job <- as.numeric(Sys.getenv("SLURM_JOB_ID")) # Gets the job ID
task <- as.numeric(Sys.getenv("SLURM_ARRAY_TASK_ID")) # Gets the task number/array number
submit_directory <- Sys.getenv("SLURM_SUBMIT_DIR") # Gets the directory you submitted your job from
user <- Sys.getenv("LOGNAME") # Gets your username

print(paste0("Task ",task," under job ID ",job, " submitted from folder ",submit_directory," by ",user))

Sys.sleep(90)

print("Done")
[hbwadde@node24 Job_array_system_variables]$
```

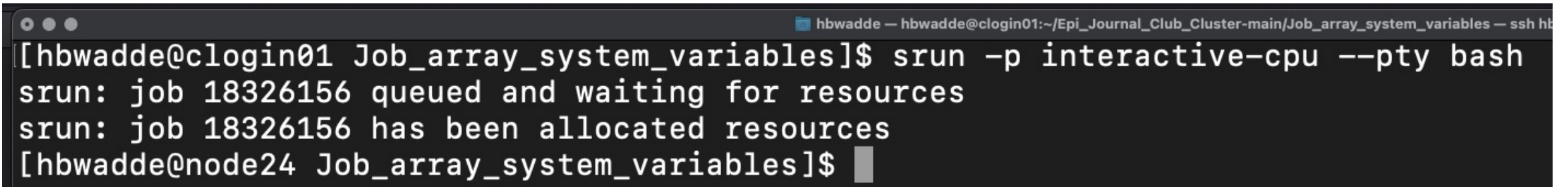
Array job output

- * in the terminal is a 'wildcard match'. It says 'match any file that starts with this'

```
[hbwadde@node24 Job_array_system_variables]$ cat R_system_variables.1832615*.out
[1] "Task 4 under job ID 18326150 submitted from folder /home/hbwadde/Epi_Journal_Club_Cluster-main/Job_array_system_variables by hbwadde"
[1] "Task 1 under job ID 18326151 submitted from folder /home/hbwadde/Epi_Journal_Club_Cluster-main/Job_array_system_variables by hbwadde"
[1] "Task 2 under job ID 18326152 submitted from folder /home/hbwadde/Epi_Journal_Club_Cluster-main/Job_array_system_variables by hbwadde"
[1] "Task 3 under job ID 18326153 submitted from folder /home/hbwadde/Epi_Journal_Club_Cluster-main/Job_array_system_variables by hbwadde"
[hbwadde@node24 Job_array_system_variables]$
```

Interactive terminals

- Not submitting a job, but too much load for the login node
 - Can use this to install R packages for later use
- Interactive terminal
- `srun -p interactive-cpu --pty bash`

A terminal window with a dark background and light text. The title bar at the top shows window control buttons and the text 'hbwadde — hbwadde@clogin01:~/Epi_Journal_Club_Cluster-main/Job_array_system_variables — ssh hbwadde@clogin01'. The terminal content shows a user at a login node submitting a job to start an interactive terminal. The output shows the job being queued and then allocated on a different node.

```
[hbwadde@clogin01 Job_array_system_variables]$ srun -p interactive-cpu --pty bash
srun: job 18326156 queued and waiting for resources
srun: job 18326156 has been allocated resources
[hbwadde@node24 Job_array_system_variables]$
```

Modules

- Software on the cluster is stored in ‘modules’ that must be loaded when you use it
- R, SAS, MATLAB, Python, etc etc
- To see all available software/versions:
 - `module spider`
 - When you’re done, quit by typing `q`
- Load module with `module load` command
 - `module load R` : loads R

Installing R packages

- `module load R`
- `R`
- `install.packages("dplyr")`
- Select mirror by typing number (0 or 71 works great)

```
[hbwadde@node24 Job_array_system_variables]$ R
```

```
R version 4.2.2 (2022-10-31) -- "Innocent and Trusting"  
Copyright (C) 2022 The R Foundation for Statistical Computing  
Platform: x86_64-pc-linux-gnu (64-bit)
```

```
R is free software and comes with ABSOLUTELY NO WARRANTY.  
You are welcome to redistribute it under certain conditions.  
Type 'license()' or 'licence()' for distribution details.
```

```
  Natural language support but running in an English locale
```

```
R is a collaborative project with many contributors.  
Type 'contributors()' for more information and  
'citation()' on how to cite R or R packages in publications.
```

```
Type 'demo()' for some demos, 'help()' for on-line help, or  
'help.start()' for an HTML browser interface to help.  
Type 'q()' to quit R.
```

```
> install.packages("dplyr")  
Installing package into '/home/hbwadde/R/x86_64-pc-linux-gnu-library/4.2'  
(as 'lib' is unspecified)  
--- Please select a CRAN mirror for use in this session ---  
Secure CRAN mirrors
```

- 1: 0-Cloud [https]
- 2: Australia (Canberra) [https]
- 3: Australia (Melbourne 1) [https]
- 4: Australia (Melbourne 2) [https]

57: Spain (A Coruña) [https]
58: Spain (Madrid) [https]
59: Sweden (Umeå) [https]
60: Switzerland (Zurich 1) [https]
61: Taiwan (Taipei) [https]
62: Turkey (Denizli) [https]
63: Turkey (Istanbul) [https]
64: UK (Bristol) [https]
65: UK (London 1) [https]
66: USA (IA) [https]
67: USA (MI) [https]
68: USA (MO) [https]
69: USA (OH) [https]
70: USA (OR) [https]
71: USA (TN) [https]
72: United Arab Emirates [https]
73: Uruguay [https]
74: (other mirrors)

Selection: 71


```
The downloaded source packages are in
      '/tmp/RtmpxqrX8a/downloaded_packages'
[> library(dplyr)

Attaching package: 'dplyr'

The following objects are masked from 'package:stats':
  filter, lag

The following objects are masked from 'package:base':
  intersect, setdiff, setequal, union

[> q()
[Save workspace image? [y/n/c]: n
[hbwadde@node24 Job_array_system_variables]$
```

Additional commands

- `rm filename` : removes a file *WARNING: PERMANENT*
- `rm -rf folder` : removes a folder
- `mkdir foldername` : creates a folder/directory
- `touch filename` : creates a file
- `exit` : logout

Further helpful topics

- <https://www.linuxcommand.org/tlcl.php> : Linux command line, free online book
- <https://linuxize.com/post/how-to-create-bash-aliases/> : Bash aliases, they let you have a 'nickname' for commands you use often