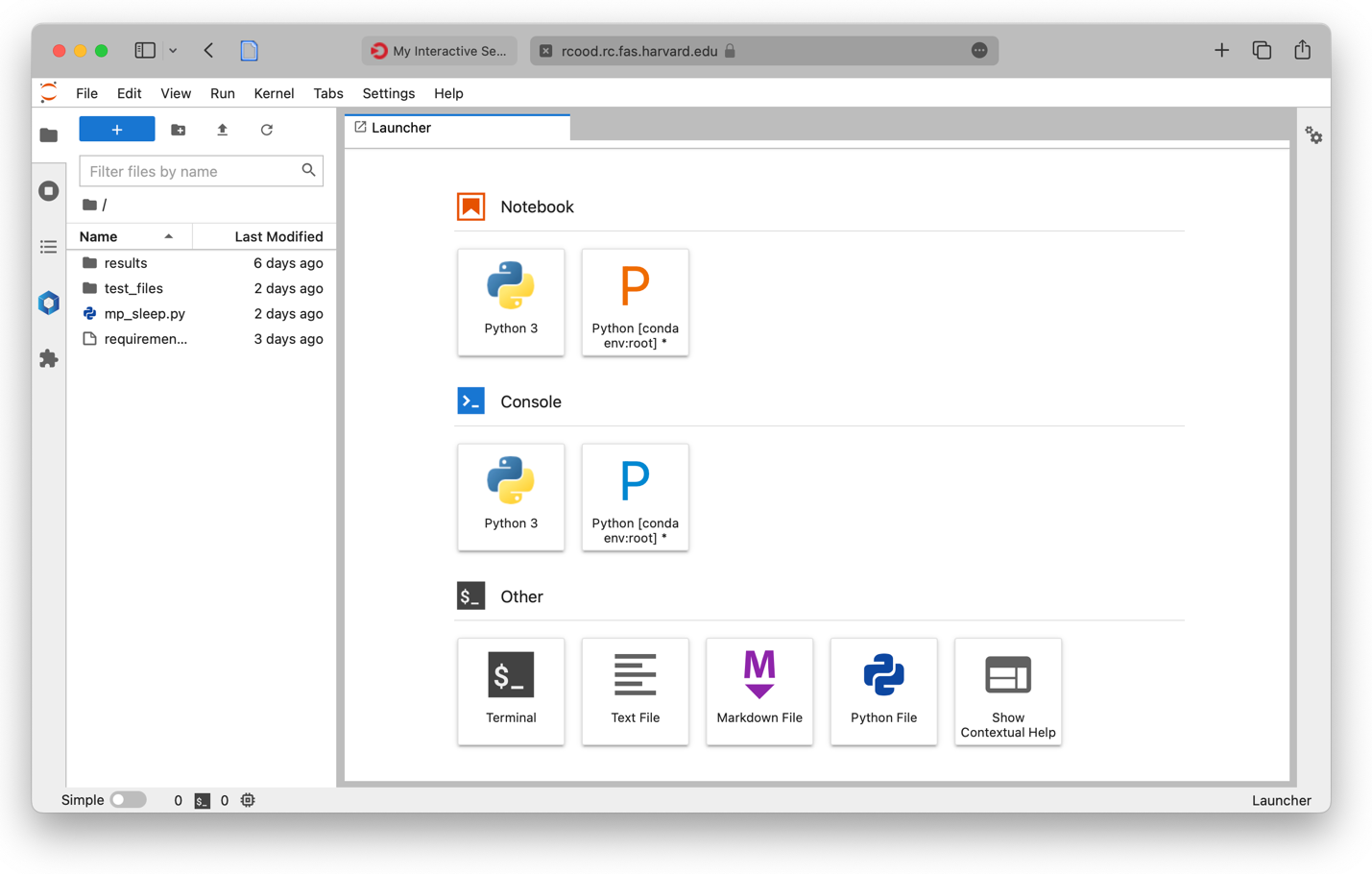
Python for Geospatial Big Data and Data Science Using the FASRC

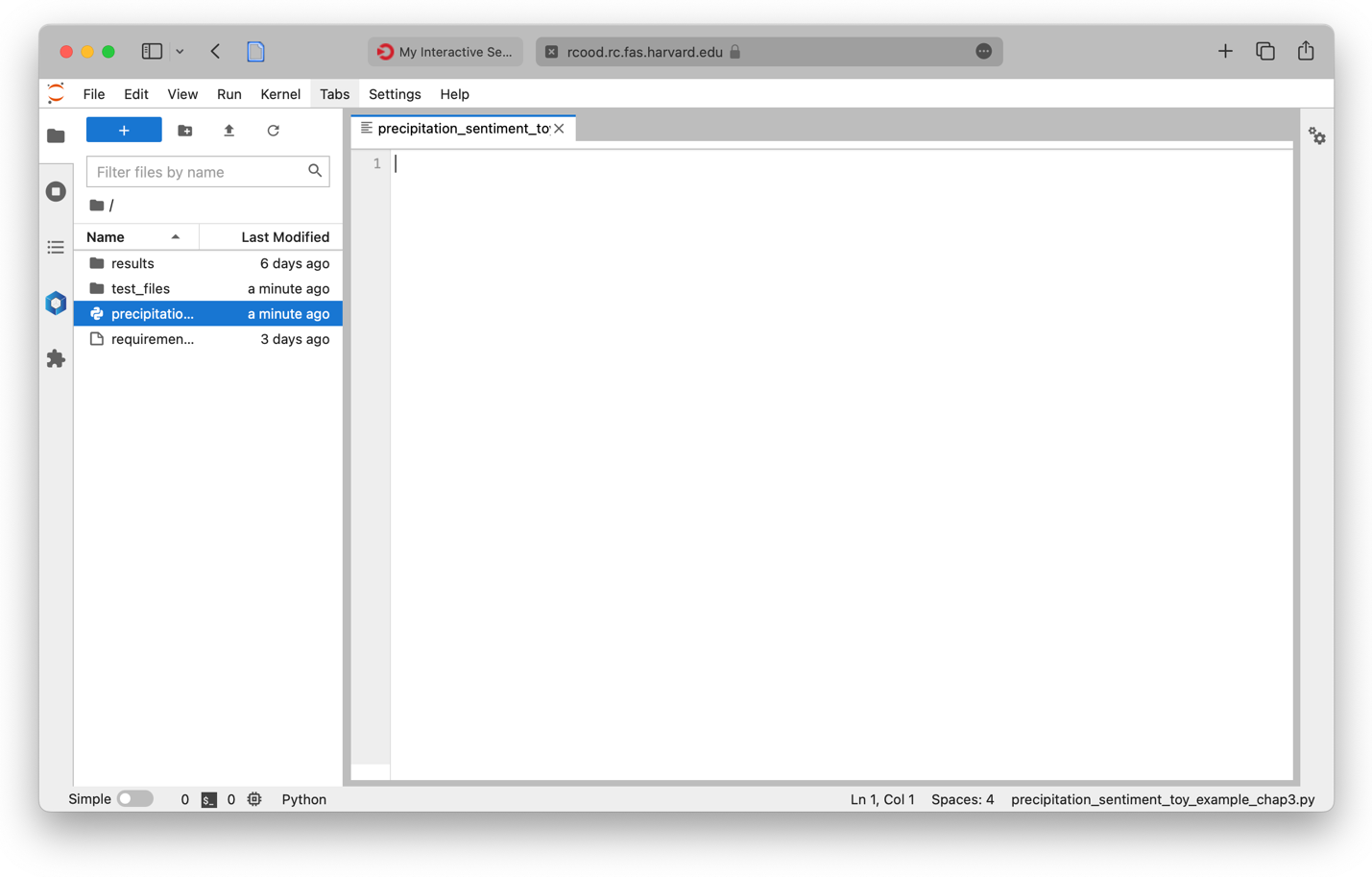
Exercise 3

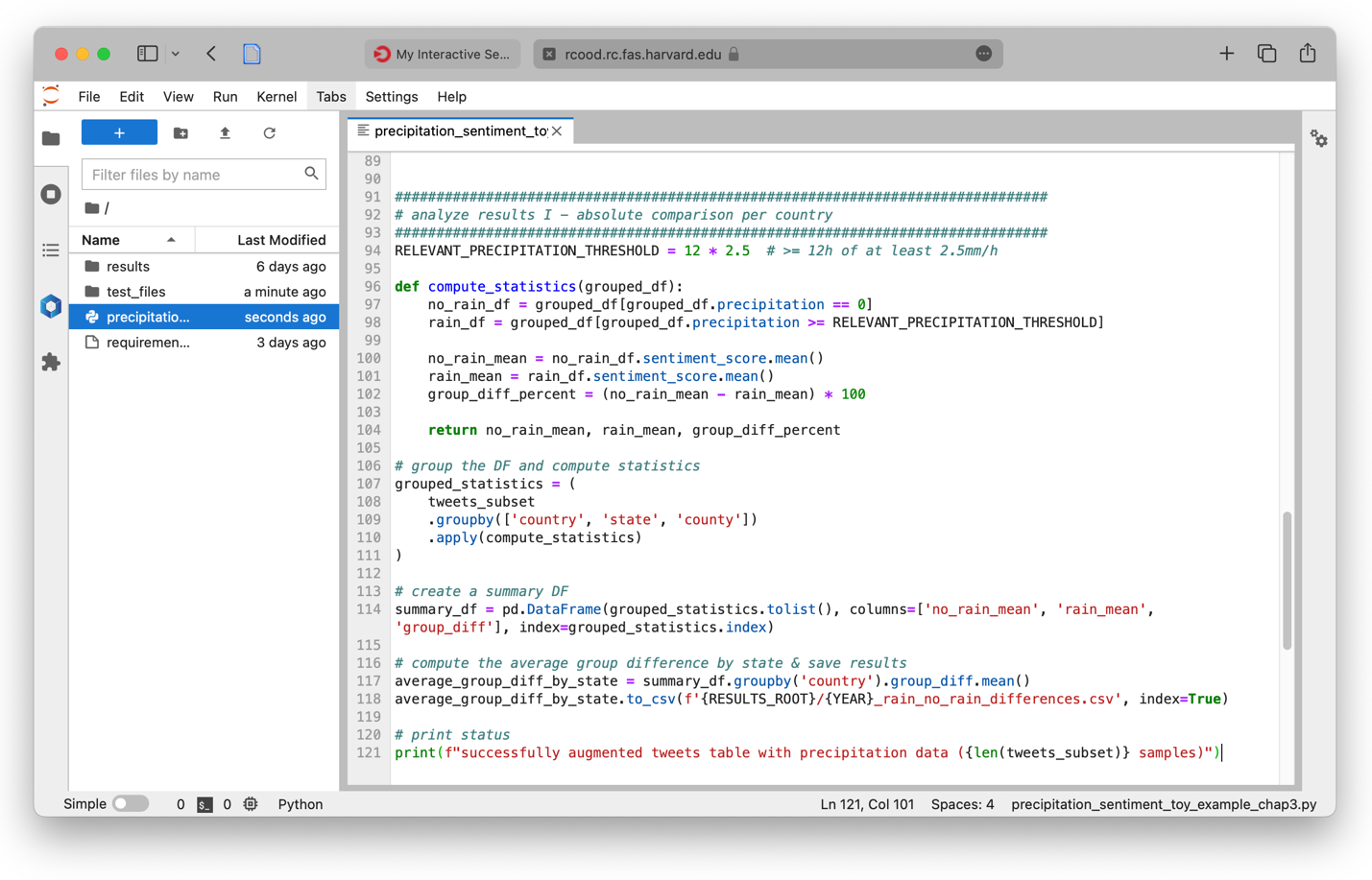
# Run your Python program on a compute-node

Start a new Jupyter session via the FASRC web interface, see Exercise 1 for help. In the “Launcher”, in section “Other”, select “Python file”:



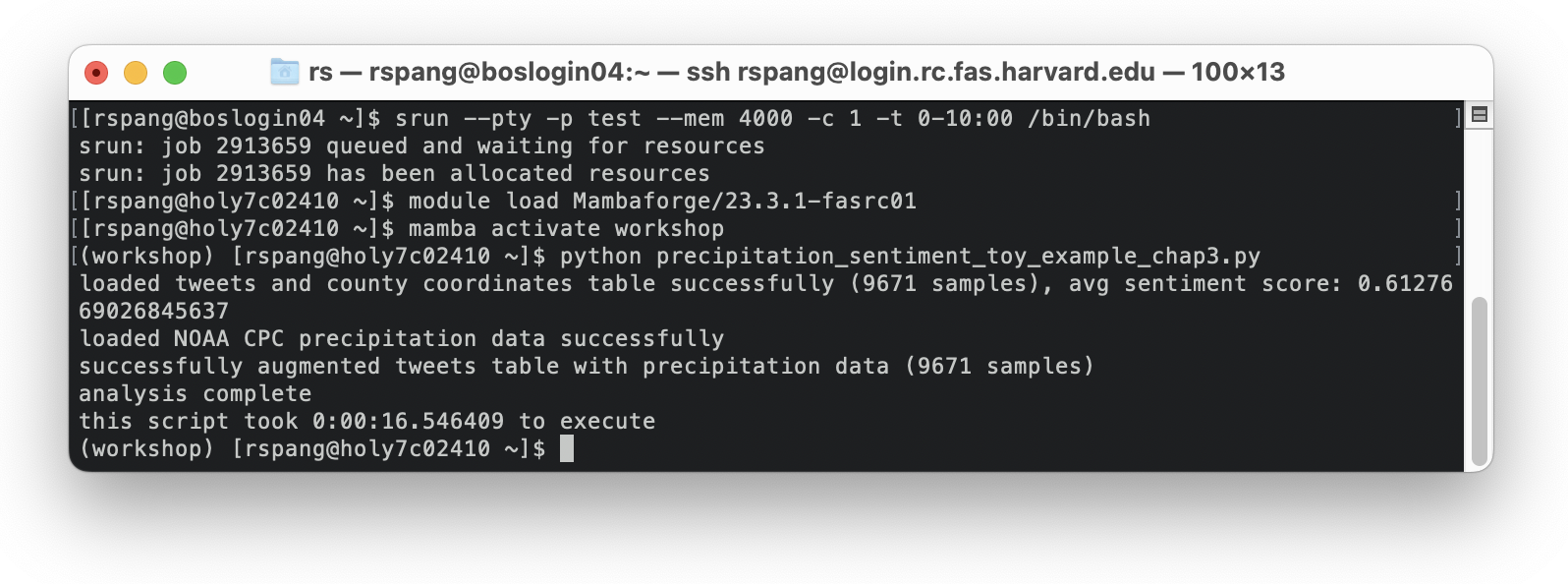
Rename the new file and copy your modified code into the window:





This is a convenient way to update your code quickly. Make sure you update the paths to the datasets: /n/holyscratch01/cga/python-workshop-gis-big-data

To run the code, use a SSH connection to a login-node and request a new interactive session. Since we now work with real data, make sure to allocate enough memory. For the example, 4GB are sufficient. srun --pty -p test --mem 4000 -c 1 -t 0-10:00 /bin/bash



## 1.2 Add time tracking

To monitor how long the execution of a segment in your code takes, you can take the time before the segment in question (or the entire code), and after. Then, you subtract the first timestamp from the second one to compute the difference. For example:

from datetime import datetime

# take start time

start\_execution\_timestamp = datetime.now()

a\_long\_running\_task() # replace this with your actual code

# take end time

end\_execution\_timestamp = datetime.now()

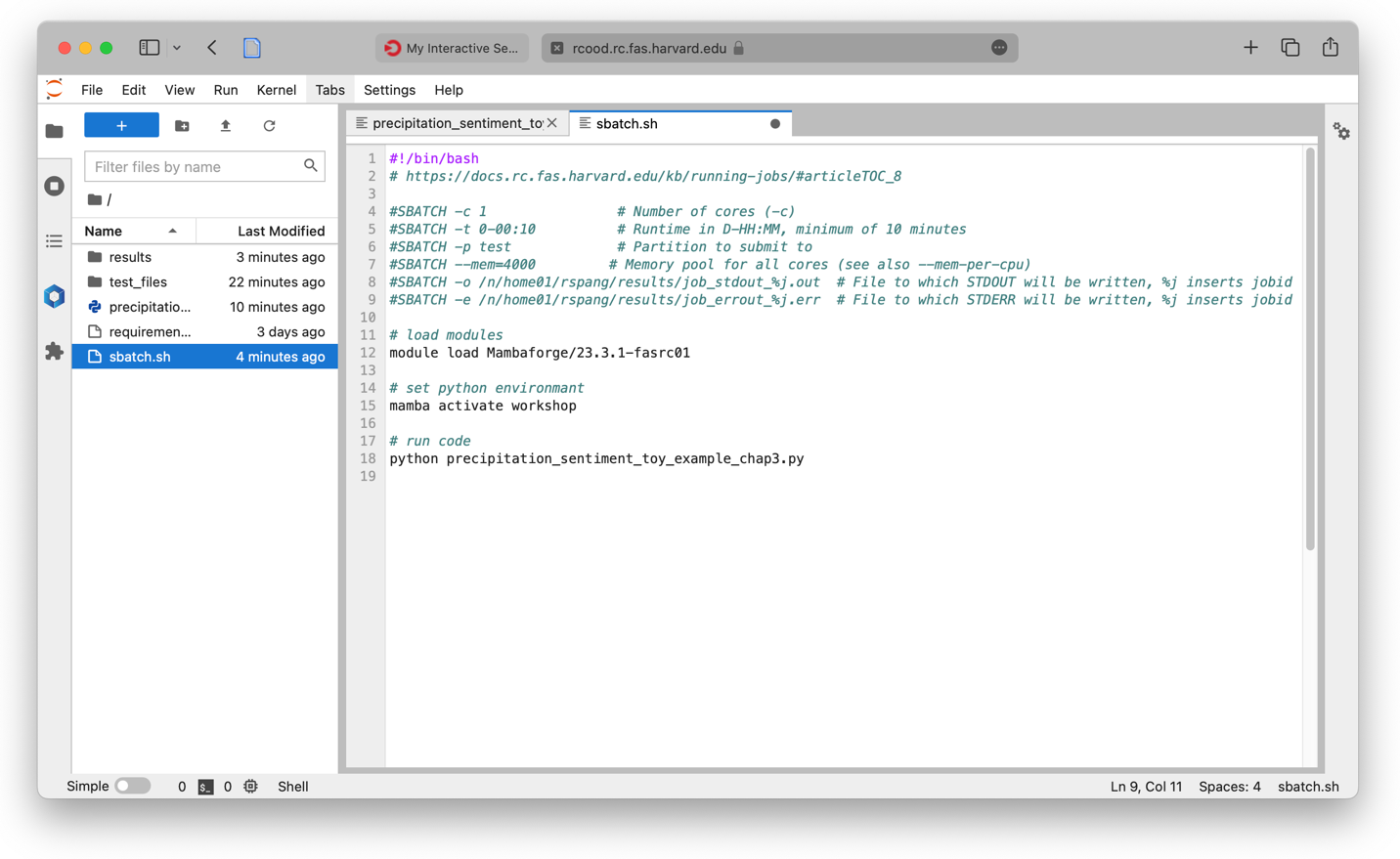
time\_diff = end\_execution\_timestamp - start\_execution\_timestamp

# print status

print(f"this script took {time\_diff}sec to execute")

# Run your Python program on a compute-node as a job

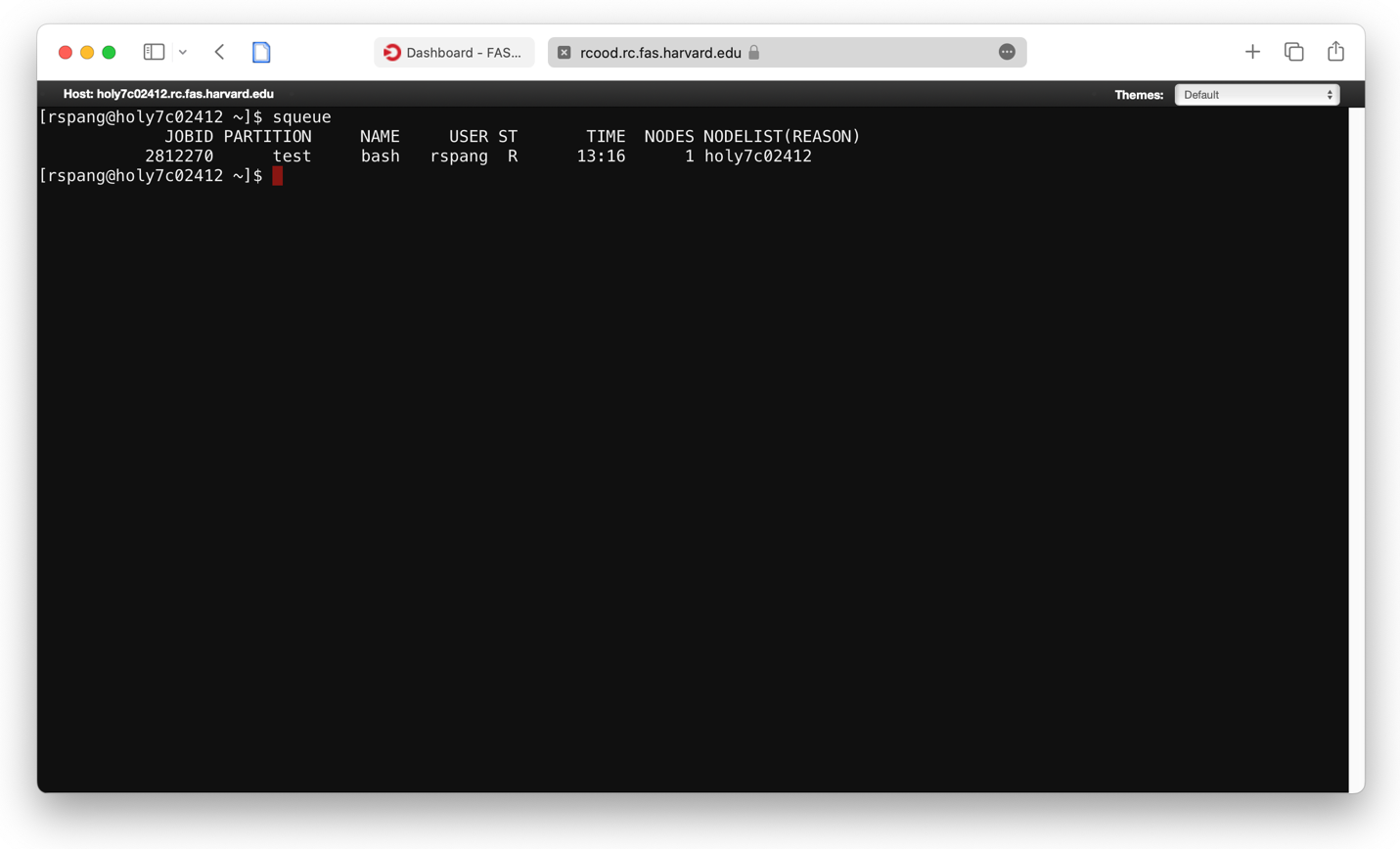
Create a sbatch script to run your Python program. This sbatch script can then be submitted as a SLURM job. The GitHub repo provides you with a template, the cheat sheet also explains the structure of a sbatch script.



Once created, the job can be submitted (on a login-node): sbatch sbatch.sh

# Monitor jobs

The SLURM command squeue returns a list of all currently running jobs. This will return (at least) the one interactive session you used to start the python program.



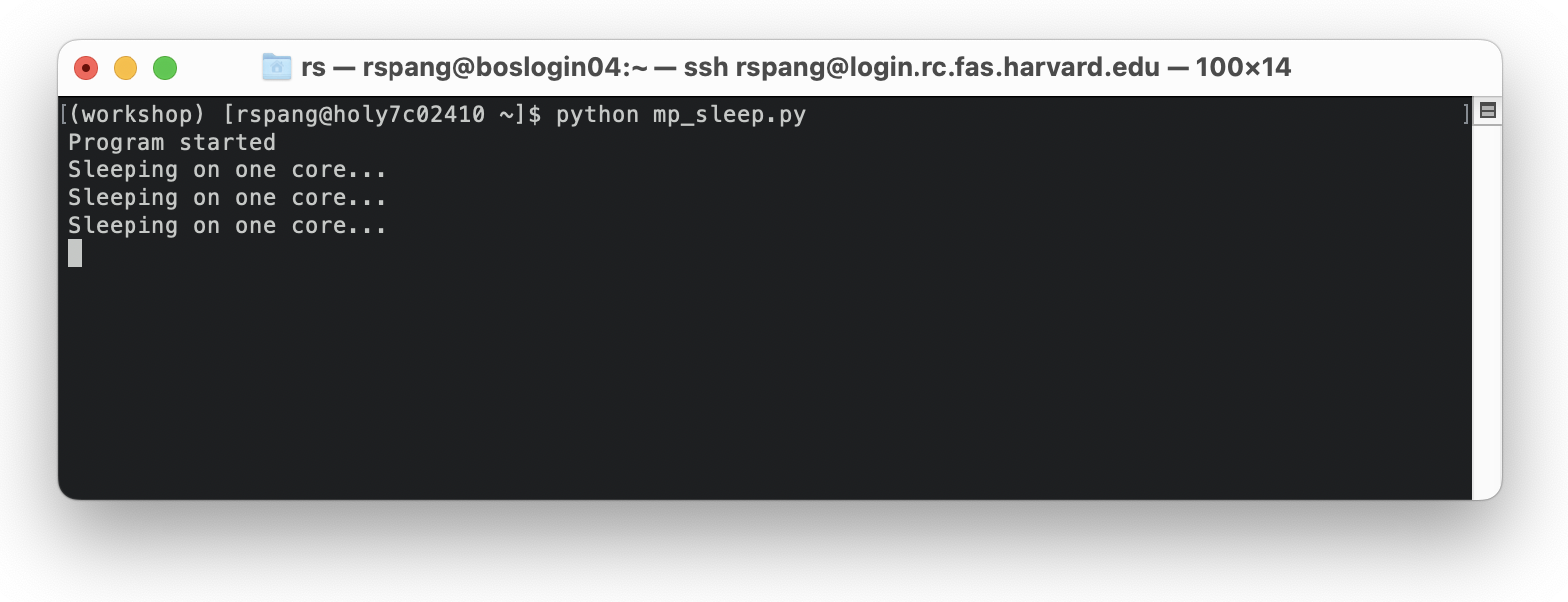
## 3.1 Monitor the execution of interactive shells

An interactive shell connection runs in your terminal; while a program is running, you cannot use the same terminal for different purposes. However, you can start a second terminal and start a second SSH connection. While this will allow you to run SLURM monitoring tools, it is not guaranteed that you will be connected to the same node if you request a second (interactive) session with SLURM.

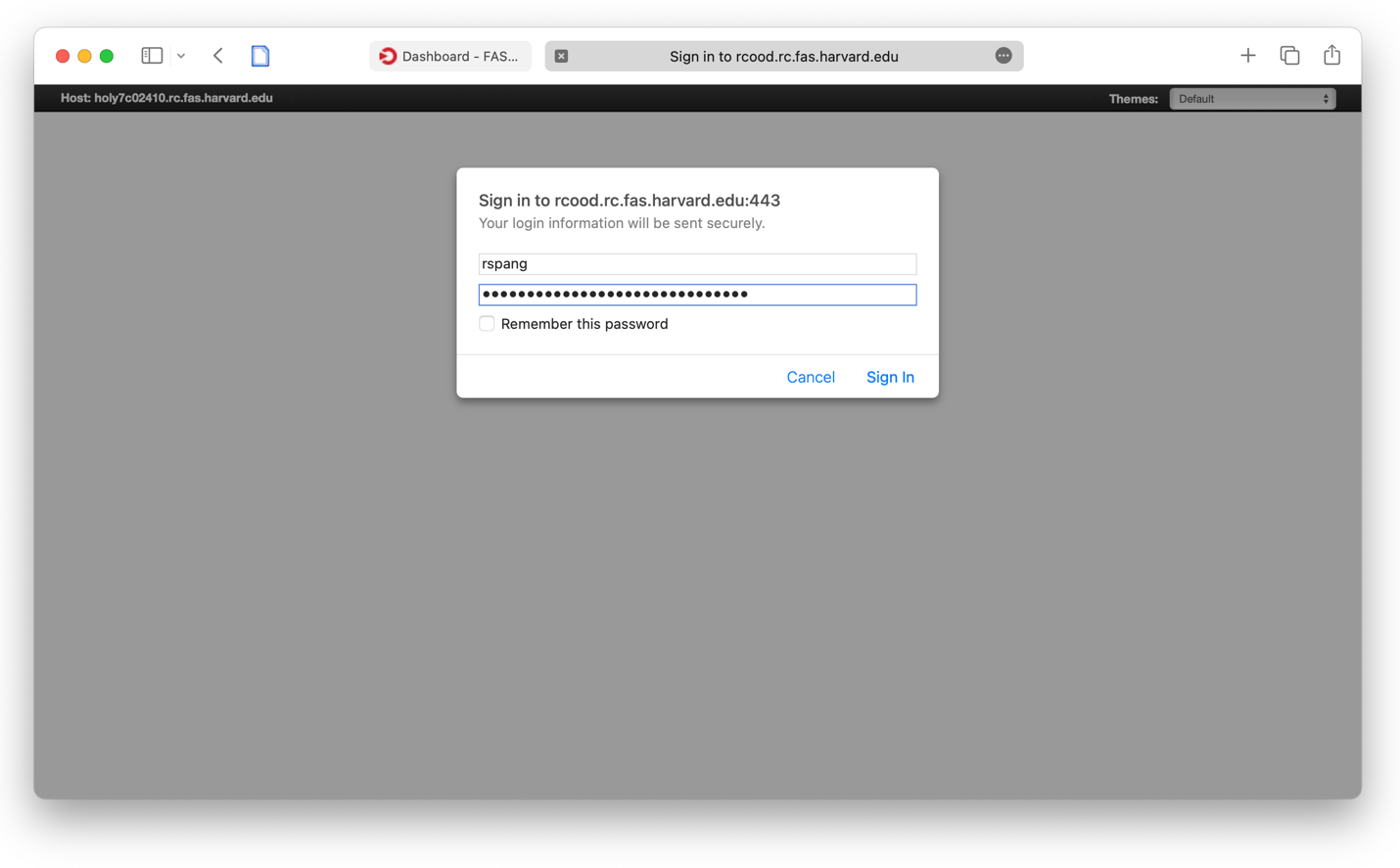
However, there is a web-tool that allows you to connect to the same instance from your browser! Open the following link

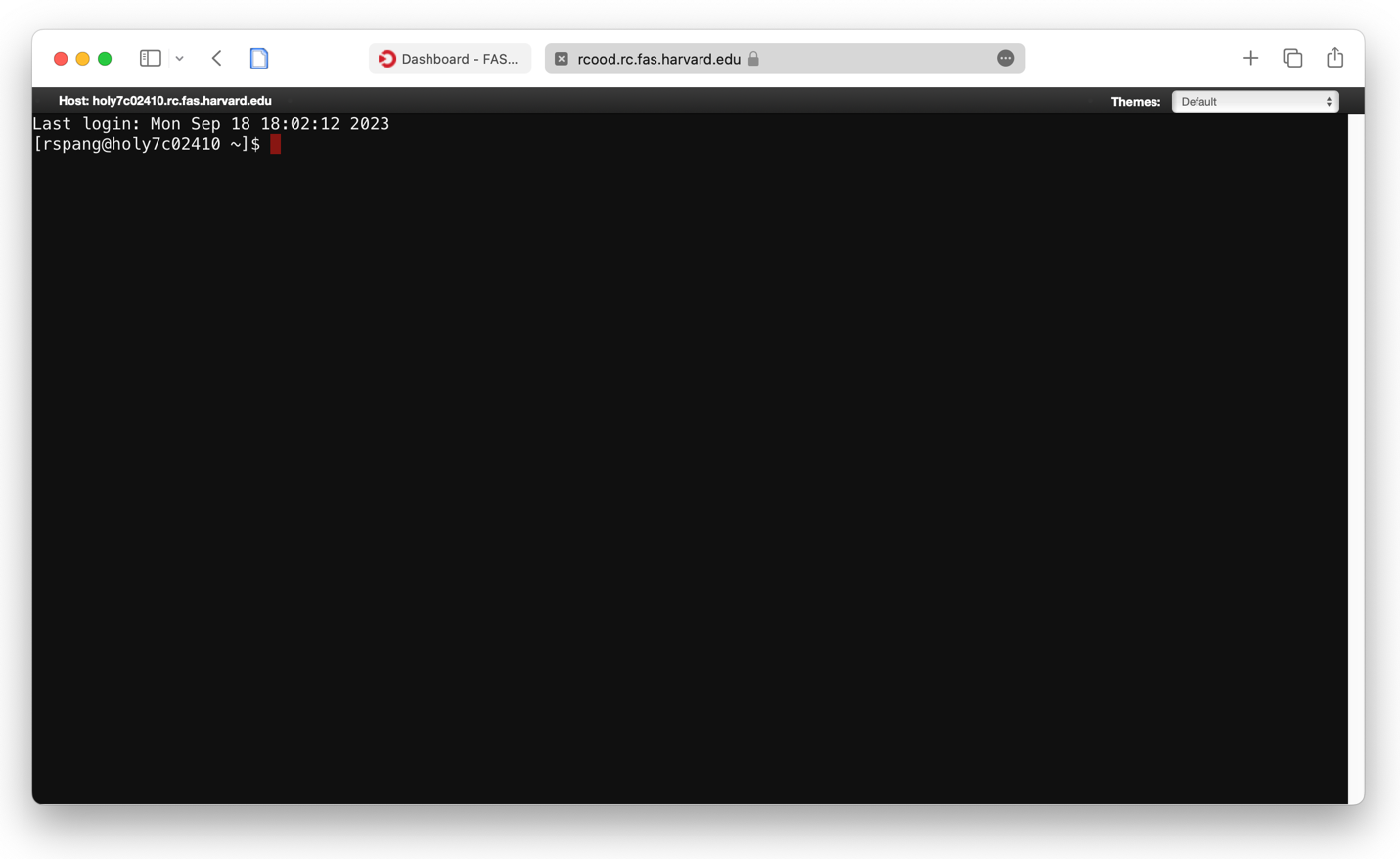
https://rcood.rc.fas.harvard.edu/pun/sys/shell/ssh/COMPUTE\_NODE.rc.fas.harvard.edu

in a browser; replace “COMPUTE\_NODE” with the node-ID you are connected to.



In this example, the node-ID is be “holy7c02410”.

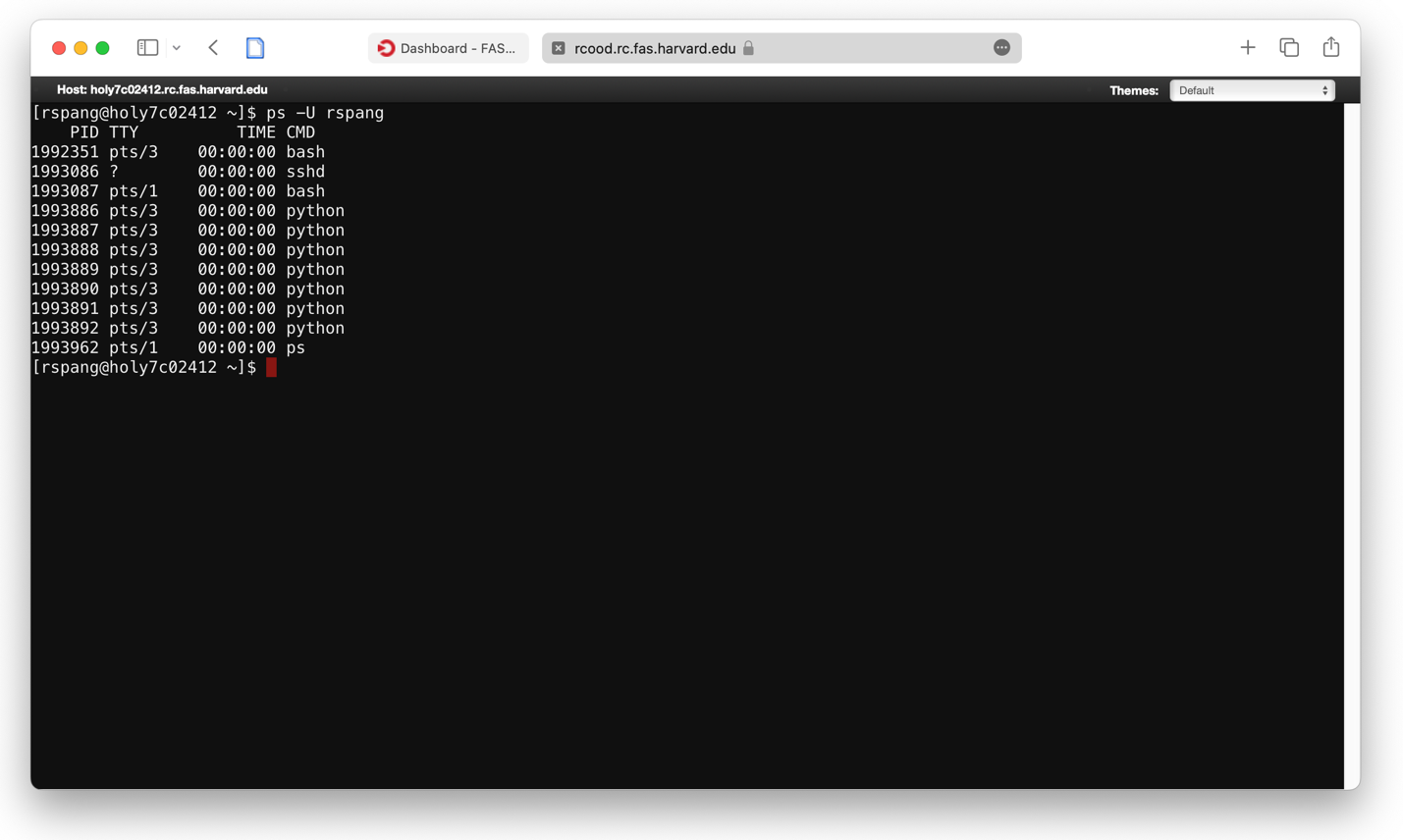
If you are asked to login first, provide your FASRC credentials:

Then, a terminal-like web page shows up.

Here, you can run all commands as you would in a normal terminal. For example, you can monitor how many processes you are currently running. Use the following command to obtain a list of all processes in your name:

ps -U username

Replace “username” with your FASRC username.



# What you learned in this exercise:

* How to create a python script using the Jupyter web interface
* How to create a sbatch script and how to submit a SLURM job
* How to start a browser-based shell to connect to an ongoing (interactive) session
* How to monitor processes and jobs while running