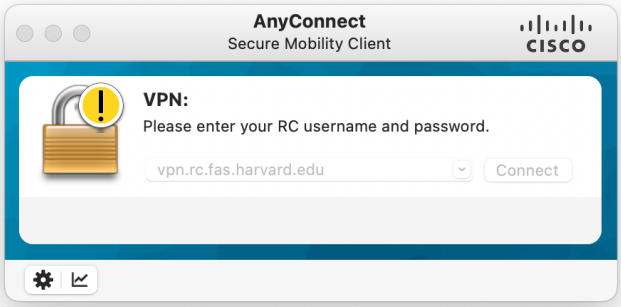
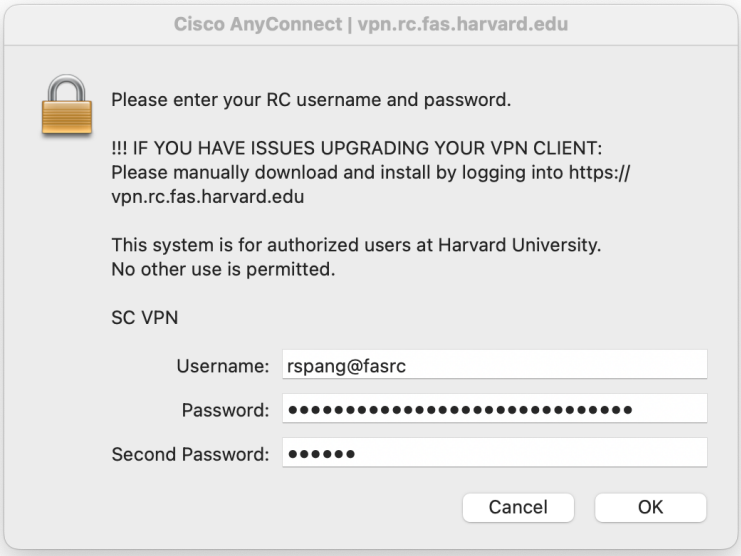
Exercise 1

1. Connect to the VPN using Cisco AnyConnect.

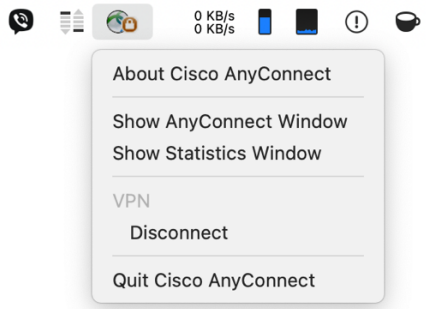
To interact with the FASRC from your laptop, you have to be connected to the VPN.
Start the Cisco AnyConnect App. Upon click on “Connect”, you’ll be asked to provide two passwords. Password: your FASRC account password. Second Password: the six digit one-time-password from your authenticator app, e.g., Google Authenticator





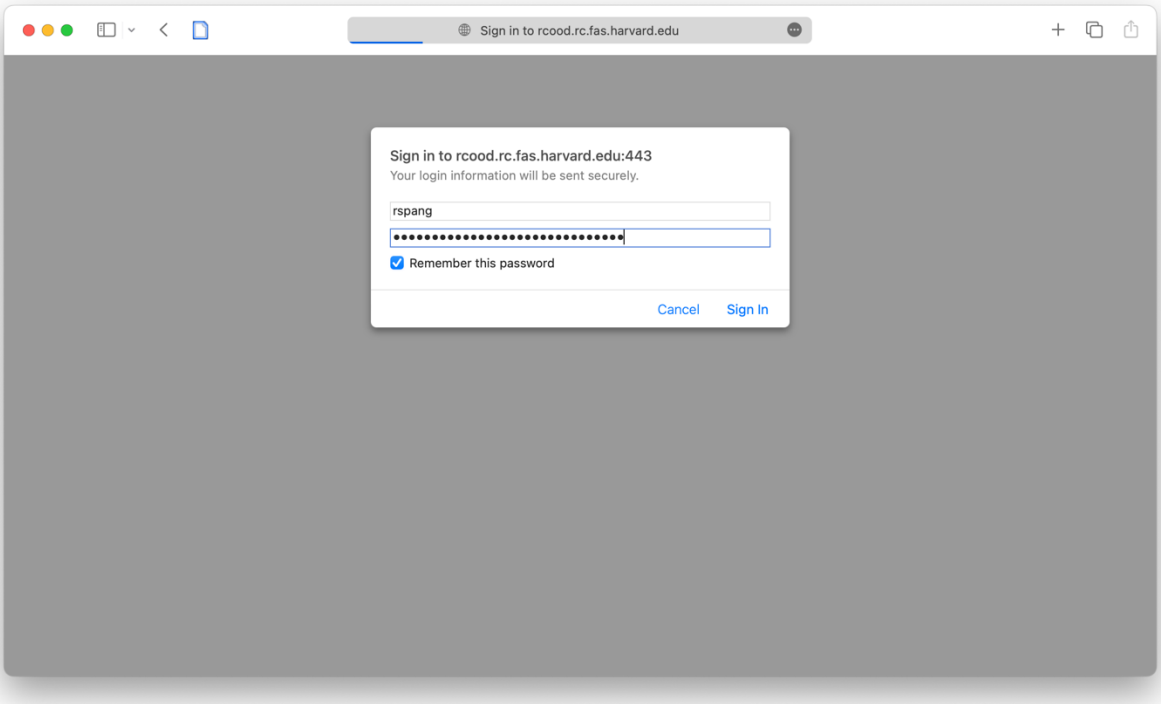
After you clicked “OK”, the VPN connection will be established, both windows disappear after a successful connection.

The AnyConnect icon indicates a successful connection with a padlock-icon.

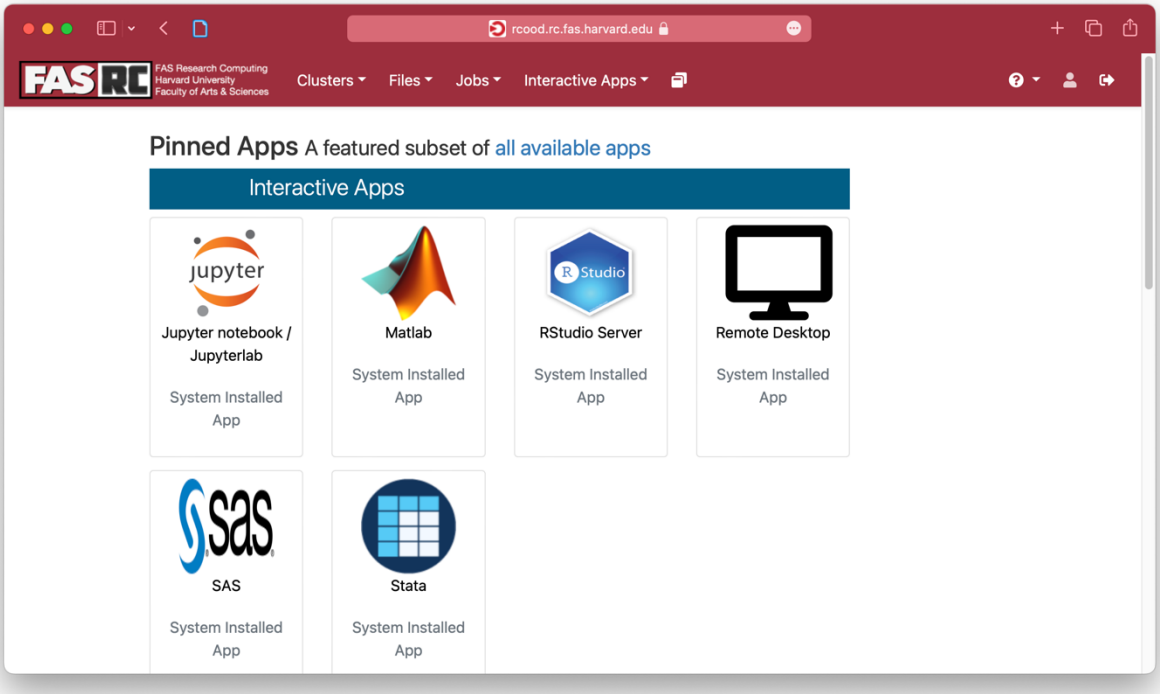


2. Login to the FASRC web interface

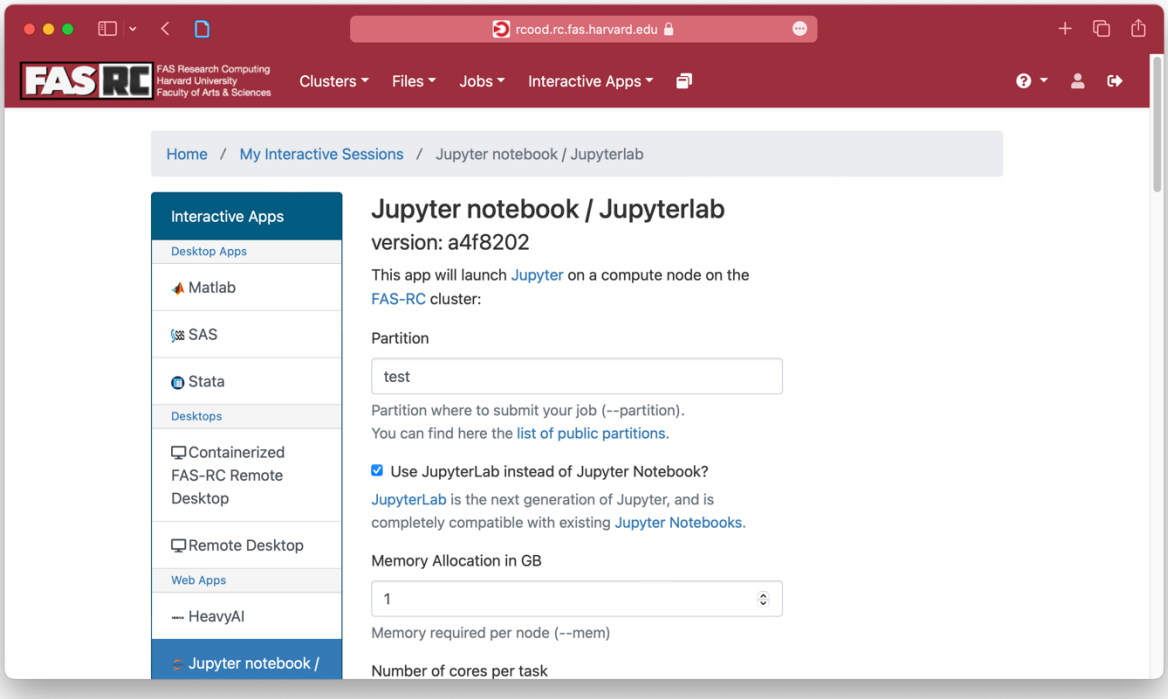
Open https://rcood.rc.fas.harvard.edu/ and login with your FASRC credentials (not your HarvardKey, but the separate account you created at sign up with the FASRC)



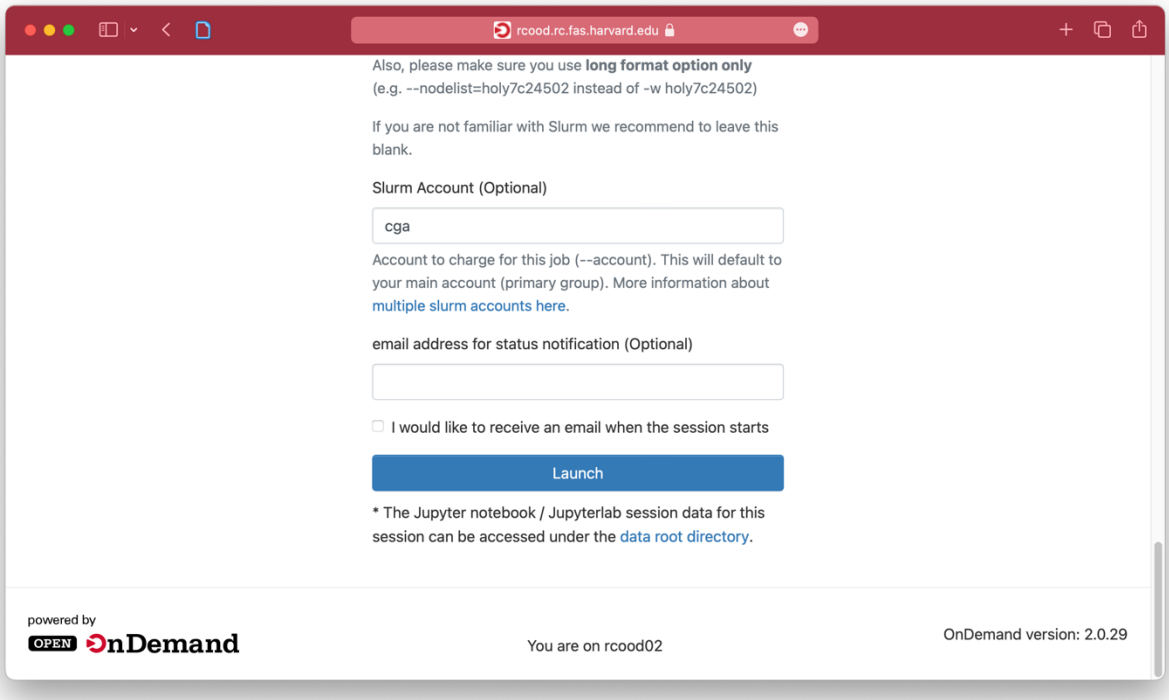
Start an interactive Jupyter notebook session



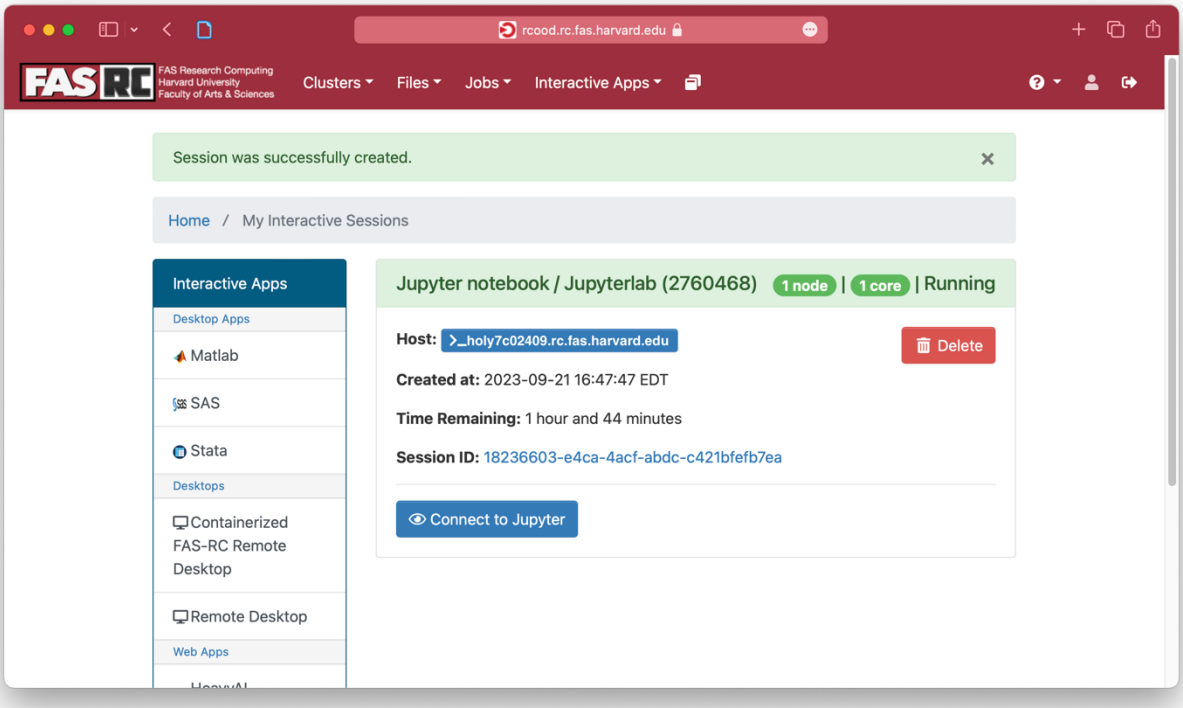
Default settings are ok… (scroll down)



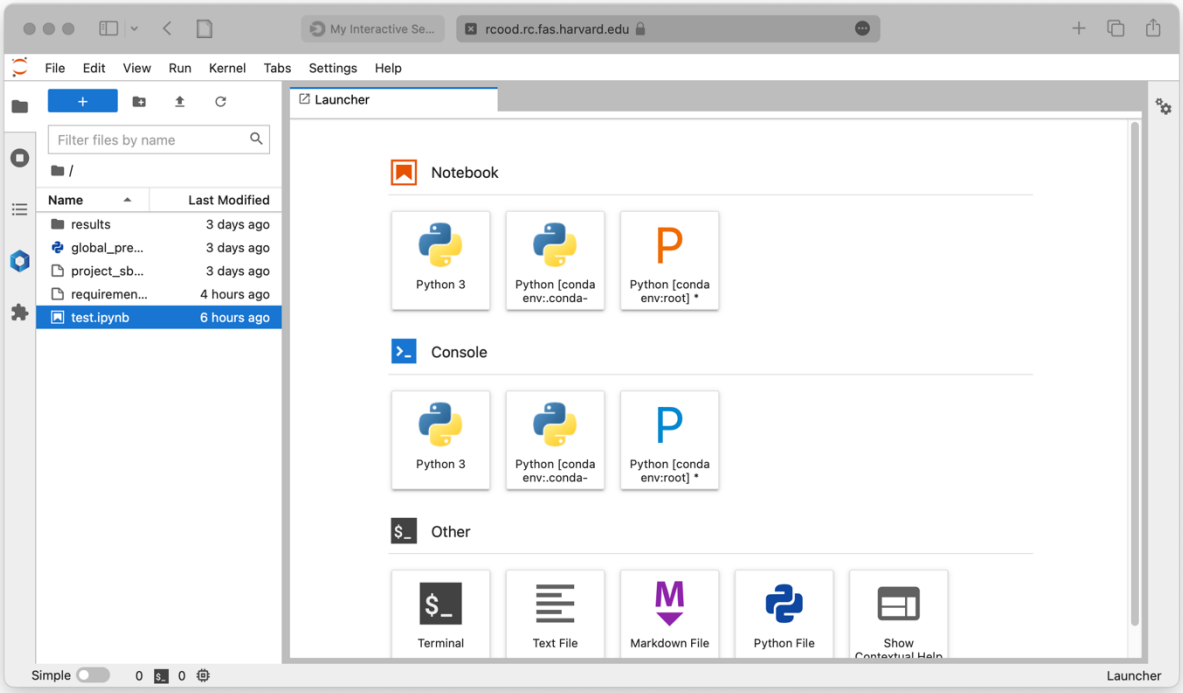
... and click “Launch” to request a new Jupyter session



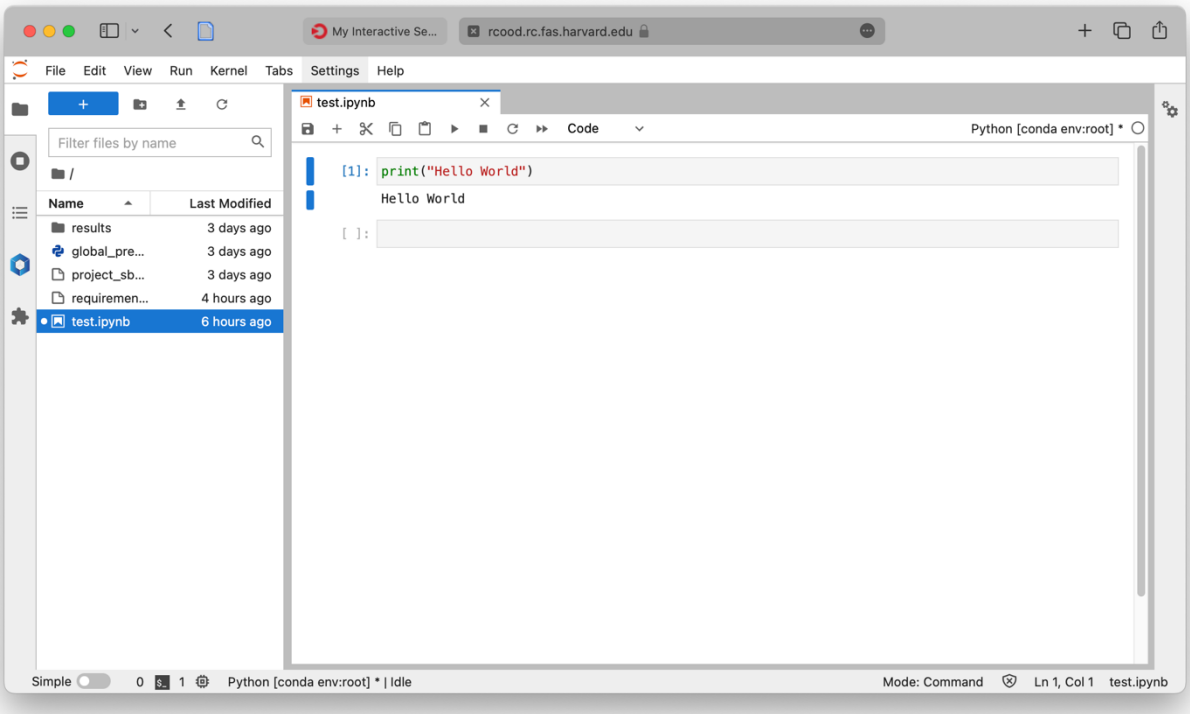
After a few seconds, the session is running. Click “Connect to Jupyter” to connect.



In Jupyter, create a new Python 3 Notebook, name it `test.ipynb`

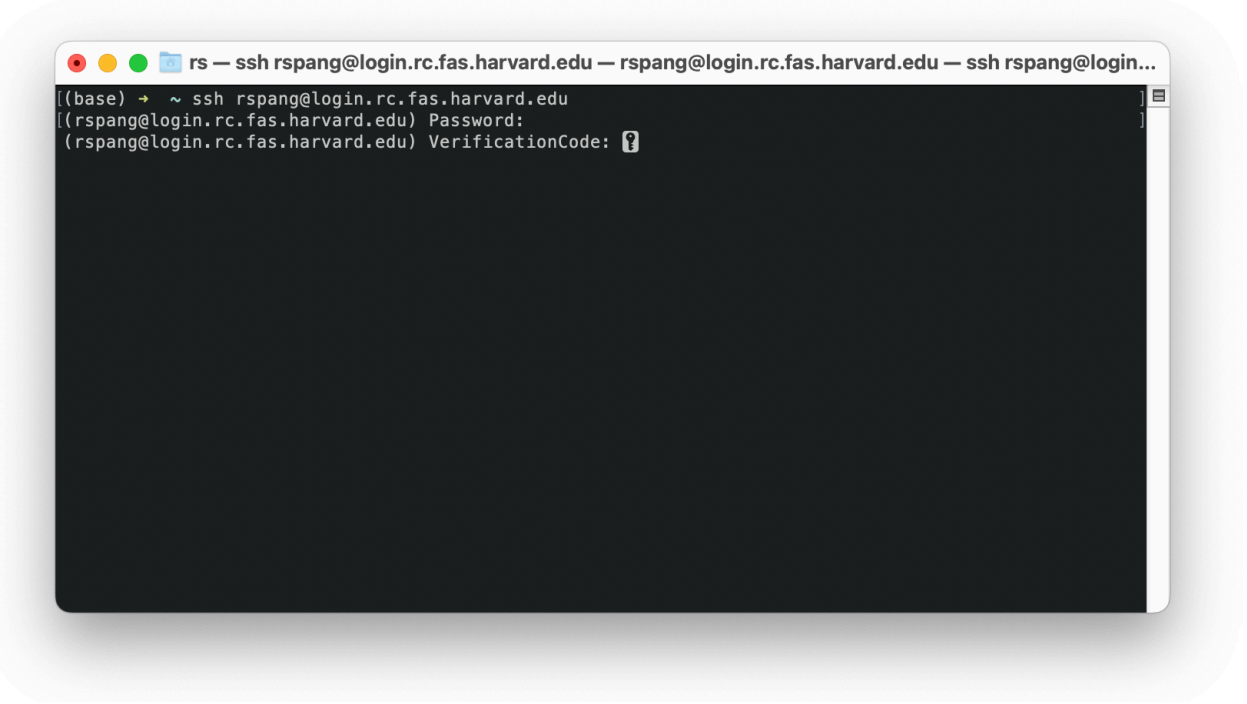


Type and run a simple demo script. This code is executed on a FASRC compute node!

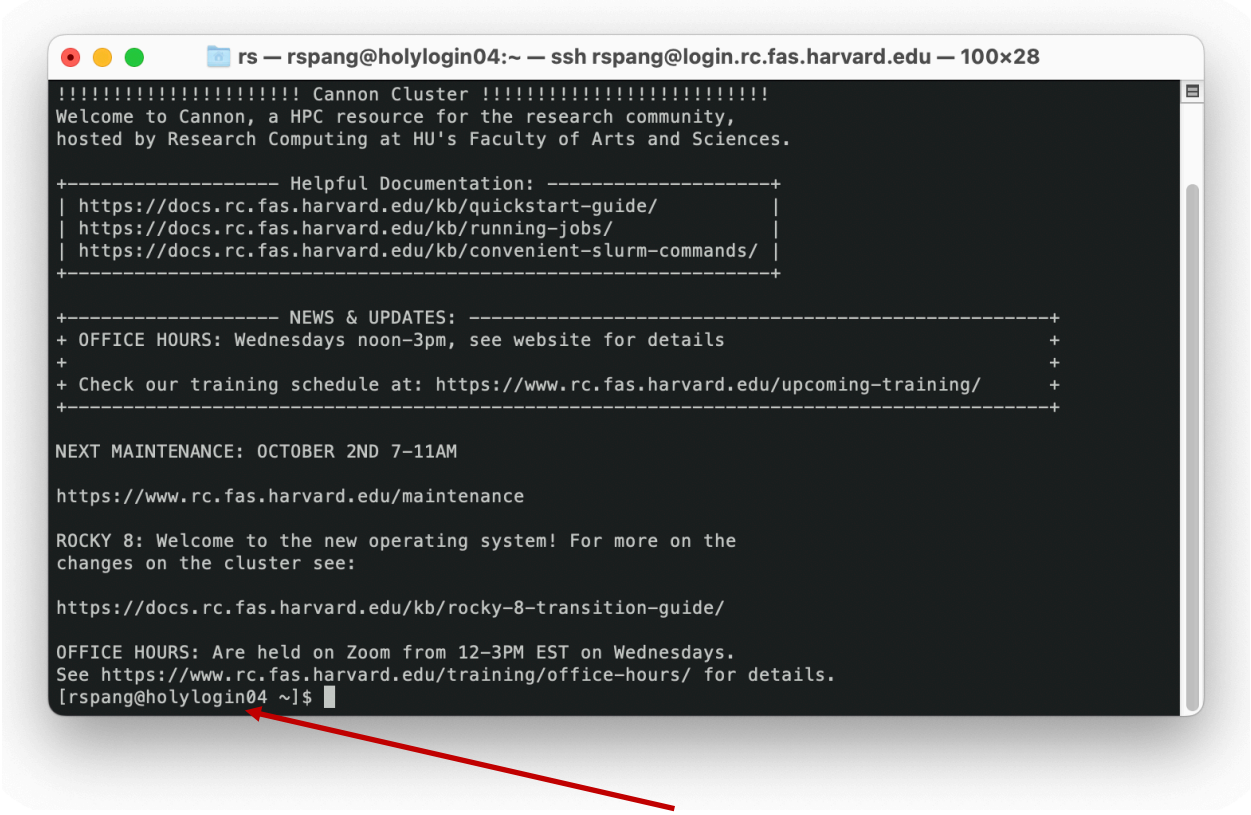


3. Login to the FASRC via the command line

Login to the cluster via SSH.
Open a terminal / console and connect to the FASRC login-node using the command:
`ssh user@login.rc.fas.harvard.edu` (replace “user” with your username)
Similarly to the VPN connection, you’ll be asked to provide two passwords: your FASRC account password, and a one-time-pad password from your authenticator app (“VerificationCode”).



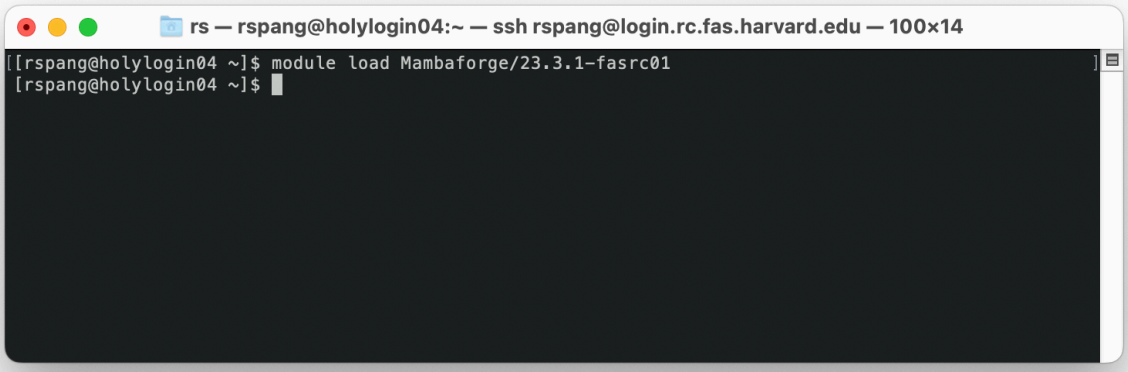
If the connection was successful, you'll be greeted with a welcome message and a few announcements:



The shell is configured in a way that it displays the hostname of the server you are using. This is helpful to check if you're on a compute node or the login node. Here, we are on a login node.

Next, setup a python environment: To do so, we first have to load a "module" that makes python available for us to use. Use the following command:

```
module load Mambaforge/23.3.1-fasrc01
```

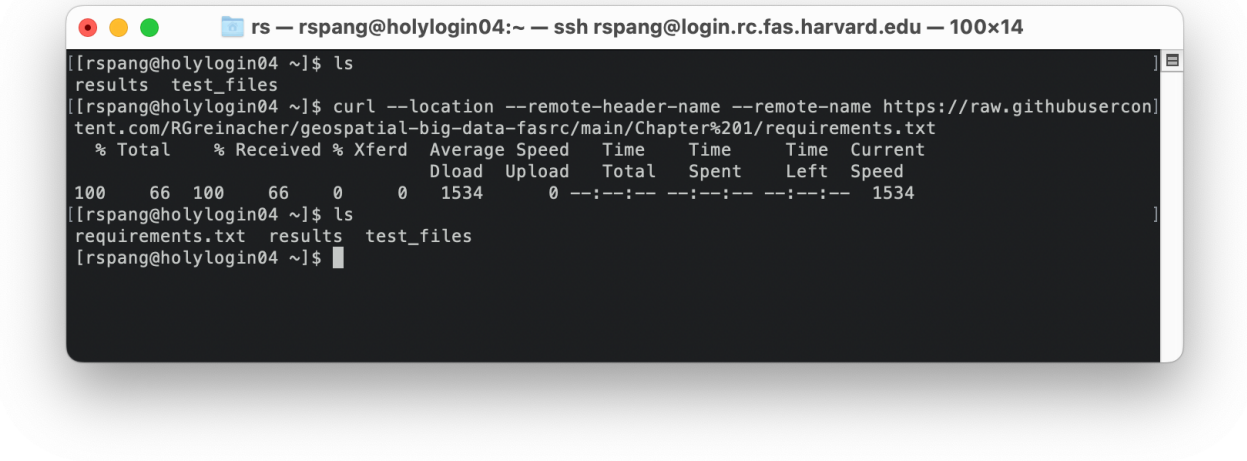


Next, we want to create a new Python environment to work with. We prepared a list of python packages that we will use today in the GitHub repository for this course. Download the file “requirements.txt” from the repo to your home-folder. Copy the following code and execute it:

```
curl --location --remote-header-name --remote-name
https://raw.githubusercontent.com/cga-harvard/python-workshop-gis-big-data/main/Chapter%201/requirements.txt
```

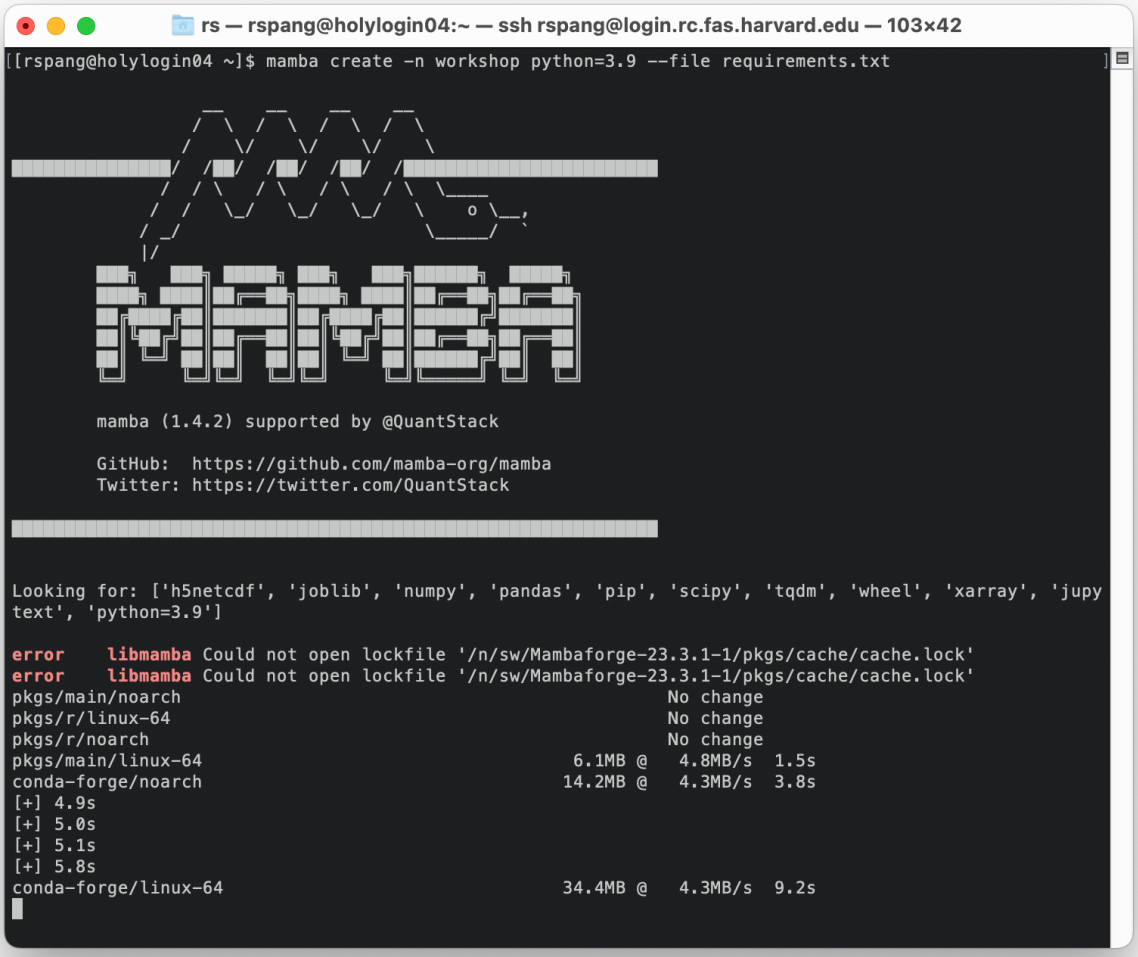
The screenshot below shows the contents of my home folder, then the curl command, and lastly the new contents of my home folder, having a new file: requirements.txt

Caution: generally, be careful what to download from the internet and make sure you trust the resource you download. It makes sense to inspect the file in your browser first.

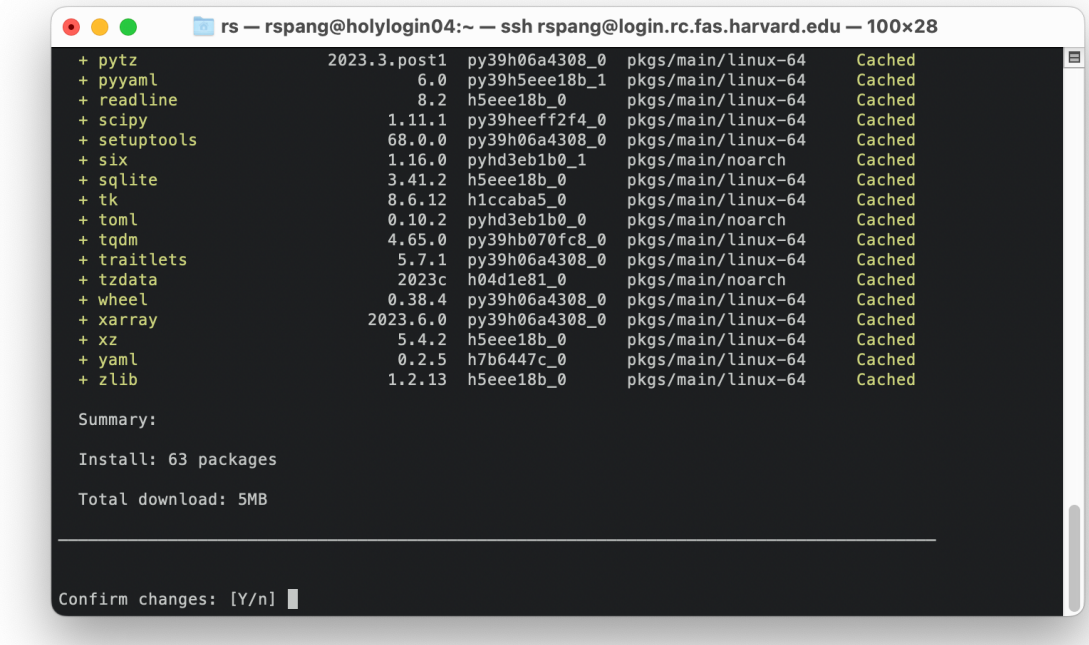


Now, having the requirements file in place, we can go ahead and create a new Python environment with all packages from the requirements file:

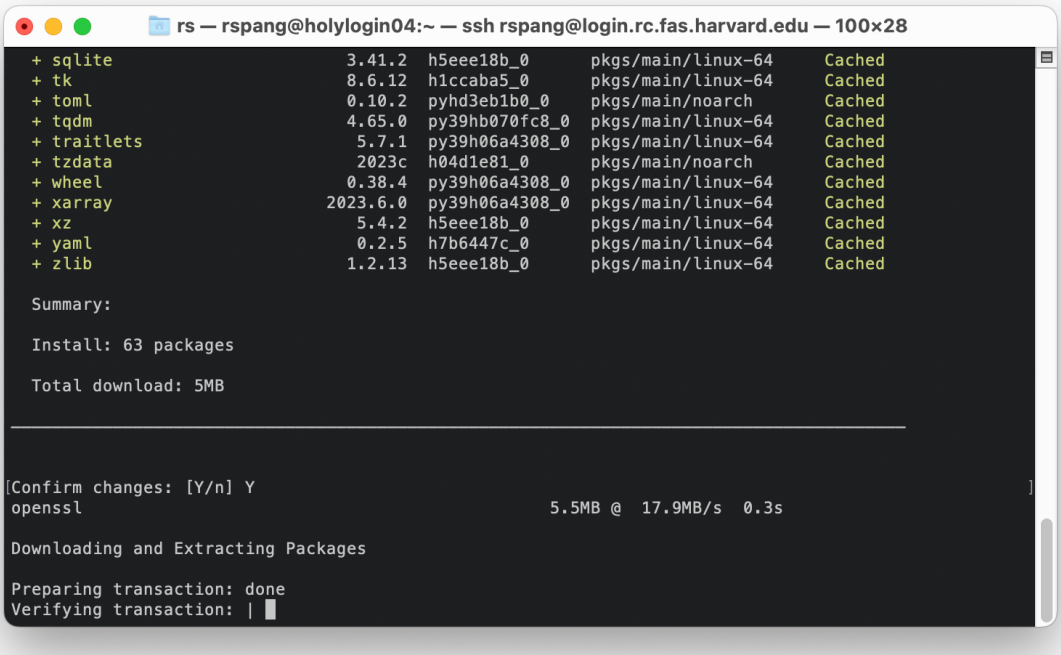
```
mamba create -n workshop python=3.9 --file requirements.txt
```



This starts a process of collecting packages that all are compatible to each other and can work together. This might take a little while. Eventually, the process lists all packages it identified it should download, and asks for a confirmation. Confirm these changes by typing "`Y`":

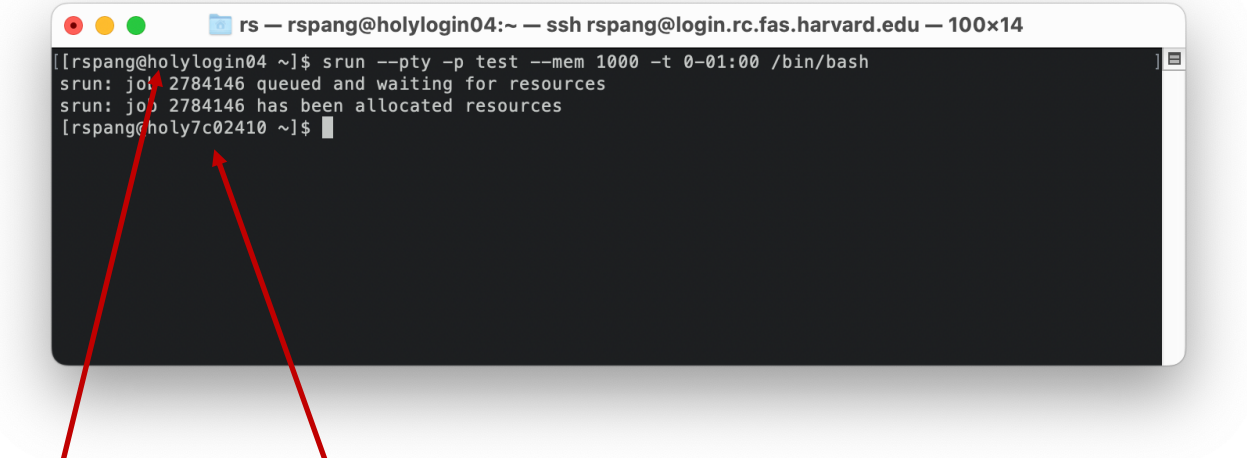


Subsequently, all packages listed are being downloaded and installed.
This also might take a moment.



Now, the environment is all set and ready to use. To start using Python, switch to a compute-node. Request an interactive session using SLURM:

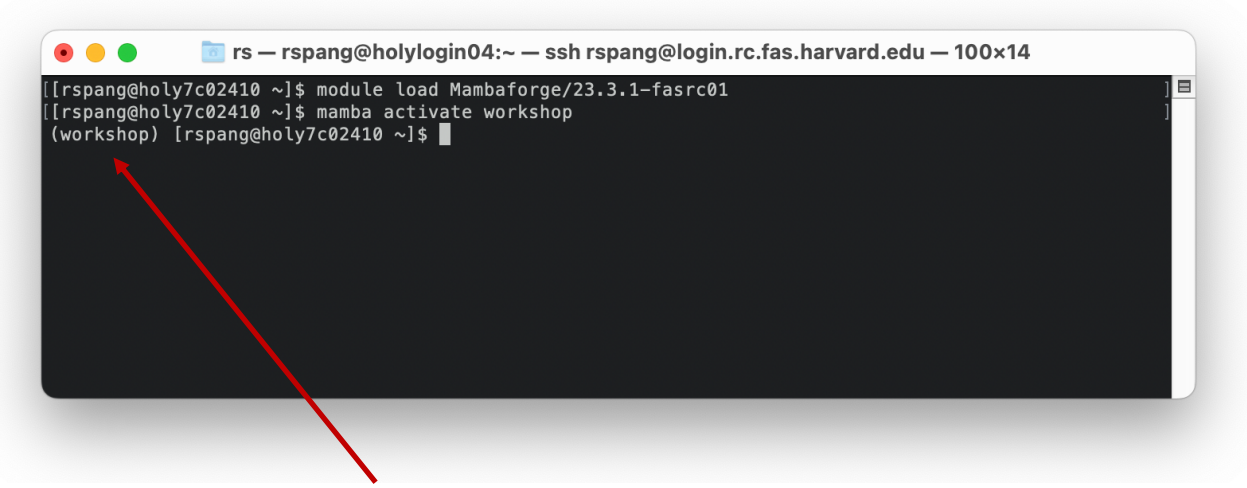
```
srun --pty -p test --mem 1000 -c 2 -t 0-01:00 /bin/bash
```



This will connect you to a different machine. Note how the hostname change from "`holylogin04`" to "`holy7c02410`". Your server names might be different. In any case, we started on a login-node, and switched to a compute-node.

Since we are on a new computer now, we have to load the Python module again. Also, we can now activate the Python environment we created earlier. Use the following two commands:

```
module load Mambaforge/23.3.1-fasrc01
mamba activate workshop
```



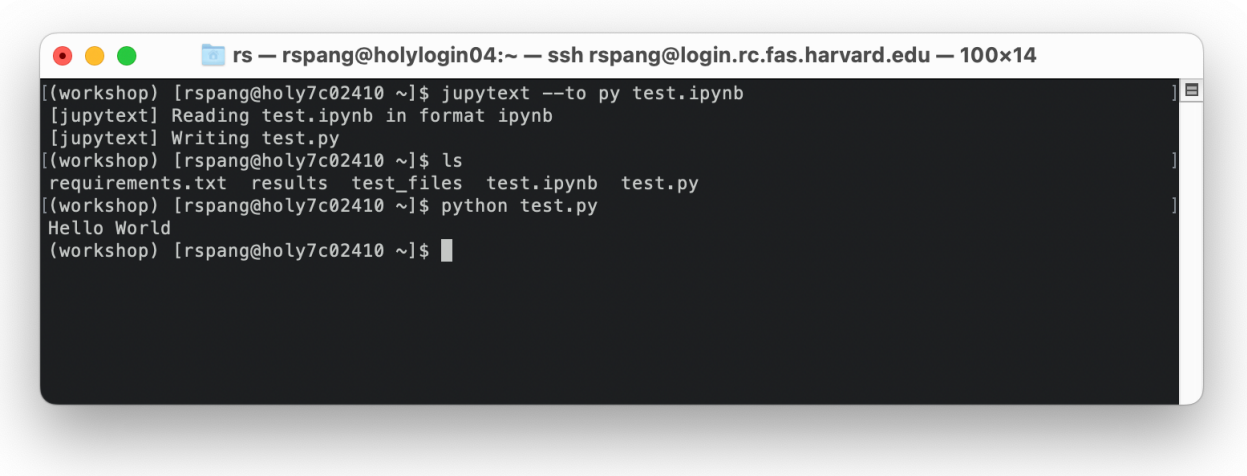
The shell also indicates the active environment.

Now, we're finally ready to run Python code! For a little example, we can recycle the demo code we used in our Jupyter notebook earlier. Since Jupyter notebooks cannot be executed on the command line directly, we can first convert the notebook to plain python using:

```
jupytext --to py test.ipynb
```

This generates a new file that is named like the notebook, but it has a "`.py`" extension. To run this code, simply use

```
python test.py
```



What you learned in this exercise:

- How to start a FASRC based Jupyter session in your browser
- How to connect to the FASRC via SSH
- How to download a file from the internet
- How to load modules
- How to create a new Python environment, providing a list of packages
- How to start an interactive SLURM session
- How to convert Jupyter notebooks to Python scripts
- How to run a Python script on a FASRC compute-node