SCC Setup

- Ref: https://www.bu.edu/tech/files/2017/06/2017_summer-Tutorial-Intro-to-SCC.pdf

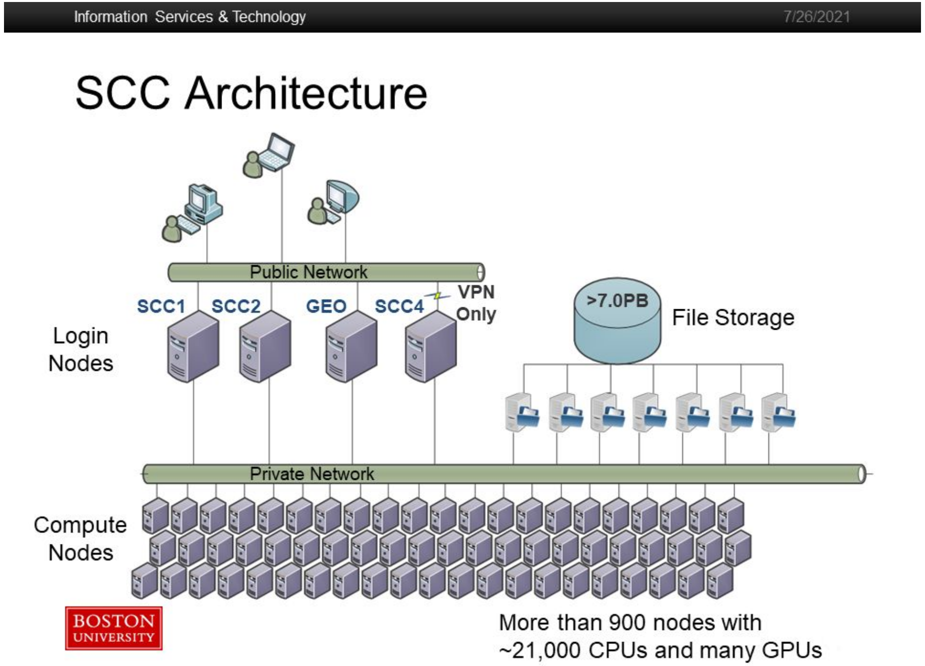
What is an SCC?

- SCC - Shared Computing Cluster - A Linux cluster that composes of both Shared and Buy-in components.

- The system currently includes over 12,000 shared CPU cores, over 16,000 Buy-in CPU cores, 300 GPU cores, and 12 petabytes of storage for research data (approximately 92% of this is Buy-in storage). (Source)

- Login Nodes - Mainly for Auth, file management, job submission and monitoring

- Compute Nodes - Running intensive computational jobs



Information Services & Technology 7/26/2021

SCC Architecture

More than 900 nodes with ~21,000 CPUs and many GPUs

Why?

- Collaborate on shared data.
- Run code that exceeds workstation capability (RAM, Network, Disk).
- Run code that runs for long periods of time (days, weeks, months)
- GPUs 💸💸💸

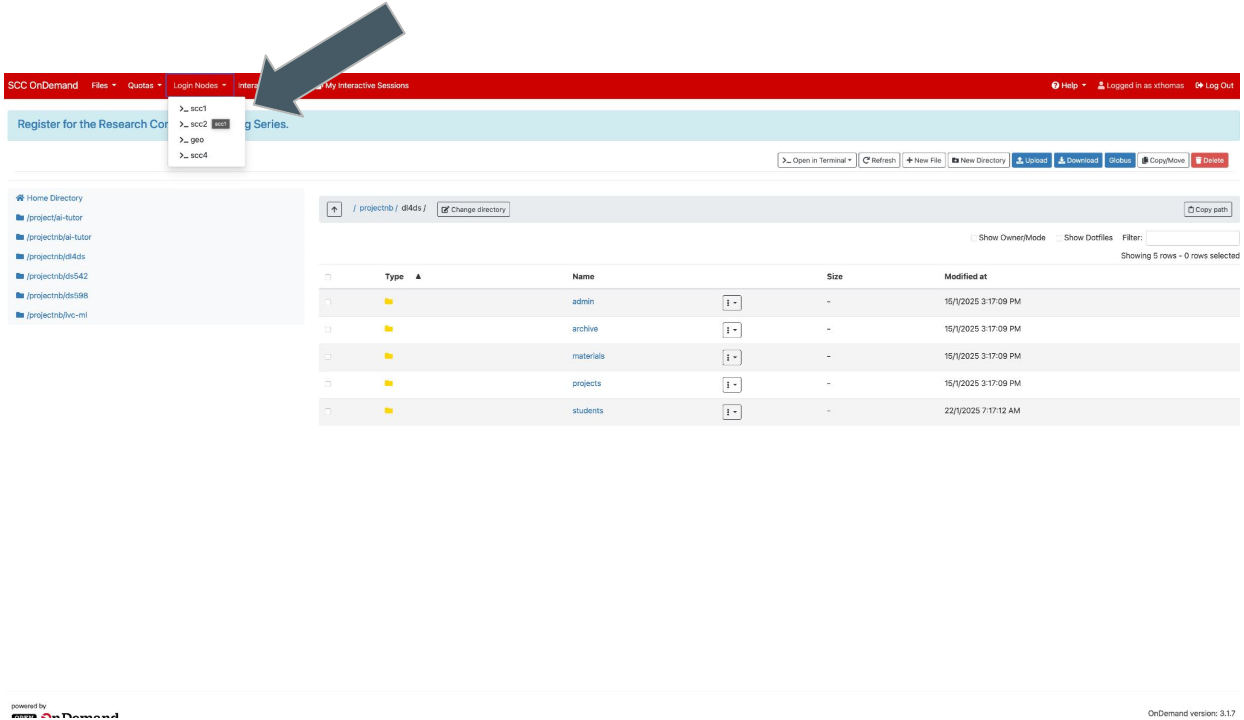
Connecting to SCC

- Launch scc-ondemand.bu.edu (check if you have '/projectnb /dl4ds /' under files)

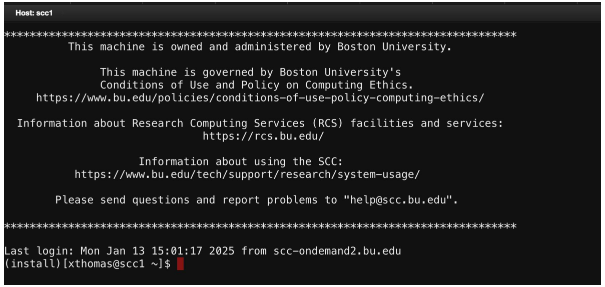
- Three ways to work on / connect to the SCC:
  - Through your browser
  - Through your terminal
  - Through your Local IDE (**Recommended**)

SCC on your browser:

- Login Nodes – Click on SCC1 or SCC2
- Will open the below in a new tab:



Type ▲	Name	Size	Modified at
	admin	-	15/1/2025 3:17:09 PM
	archive	-	15/1/2025 3:17:09 PM
	materials	-	15/1/2025 3:17:09 PM
	projects	-	15/1/2025 3:17:09 PM
	students	-	22/1/2025 7:17:12 AM



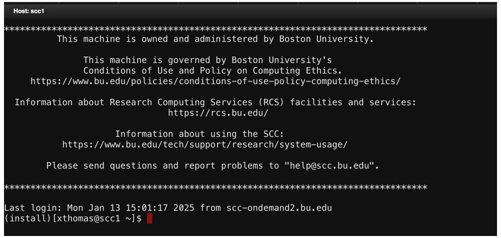
```
*********************************************************************************
          This machine is owned and administered by Boston University.

               This machine is governed by Boston University's
               Conditions of Use and Policy on Computing Ethics.
     https://www.bu.edu/policies/conditions-of-use-policy-computing-ethics/

  Information about Research Computing Services (RCS) facilities and services:
                               https://rcs.bu.edu/

                     Information about using the SCC:
           https://www.bu.edu/tech/support/research/system-usage/

        Please send questions and report problems to "help@scc.bu.edu".

*********************************************************************************

Last login: Mon Jan 13 15:01:17 2025 from scc-ondemand2.bu.edu
(install)[xthomas@scc1 ~]$ 
```

Via VSCode (Recommended)

- Install: https://code.visualstudio.com/download
- Follow these steps to establish a remote connection to the SCC via SSH (through your local VSCode):
  - https://www.bu.edu/tech/support/research/system-usage/scc-environment/editors-viewers-and-ides/vscode/

- And install these extensions:
  - GitHub Copilot (Recommendation)
  - Black Formatter - To ensure consistent code formatting for your python files (Recommendation)
  - Remote-SSH – Installed form previous step
  - Working with GitHub in VSCode

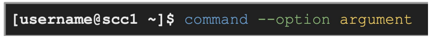
Important!!



```
*********************************************************************************
          This machine is owned and administered by Boston University.

               This machine is governed by Boston University's
               Conditions of Use and Policy on Computing Ethics.
     https://www.bu.edu/policies/conditions-of-use-policy-computing-ethics/

  Information about Research Computing Services (RCS) facilities and services:
                                https://rcs.bu.edu/

                     Information about using the SCC:
           https://www.bu.edu/tech/support/research/system-usage/

        Please send questions and report problems to "help@scc.bu.edu".

*********************************************************************************

Last login: Mon Jan 13 15:01:17 2025 from scc-ondemand2.bu.edu
(install)[xthomas@scc1 ~]$ 
```

- Everytime you connect via a login node, it will take you to what's called your “home” directory.

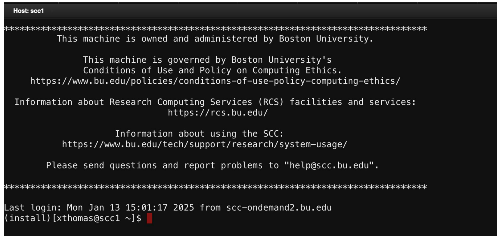
- Check your **p**resent **w**orking **d**irectory: Type “pwd” – something like “/usr3/….” would be displayed – this is your home directory.

- “du -sh .” - To check the size of your home directory – **This should always be under 10G** – or else, you may not be able to write/execute files.
  - Linux basics: 

    ```
    [username@scc1 ~]$ command --option argument
    ```
  - Here, command: du (Stands for disk usage), option: -s (stands for total size), -h (human readable format), argument: “.” – refers to the current directory.

- What does this mean?
  - You SHOULD NOT install anything to your home directory
  - No pip installs, conda installs, etc to your home directory! – Next slide on how to ensure this

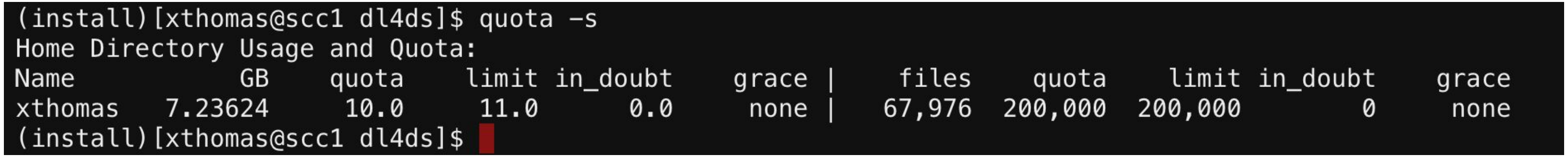
Important!!



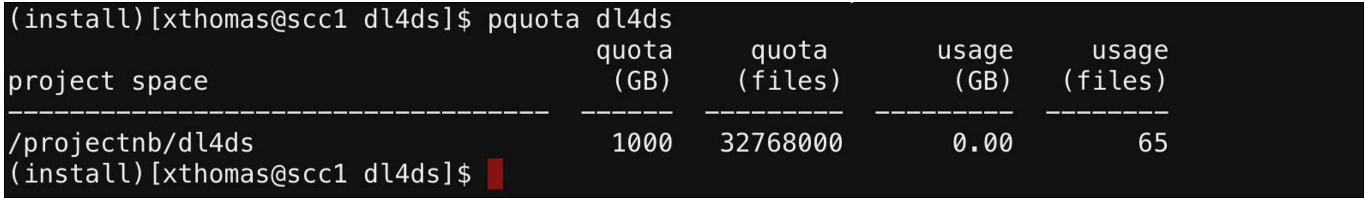
```
********************************************************************************
          This machine is owned and administered by Boston University.

               This machine is governed by Boston University's
               Conditions of Use and Policy on Computing Ethics.
     https://www.bu.edu/policies/conditions-of-use-policy-computing-ethics/

  Information about Research Computing Services (RCS) facilities and services:
                              https://rcs.bu.edu/

                     Information about using the SCC:
           https://www.bu.edu/tech/support/research/system-usage/

        Please send questions and report problems to "help@scc.bu.edu".

********************************************************************************

Last login: Mon Jan 13 15:01:17 2025 from scc-ondemand2.bu.edu
(install)[xthomas@scc1 ~]$
```

- Change your working directory from your home directory to “/projectnb/dl4ds/students/{your_bu_username}”
- Type “cd /projectnb/dl4ds/students/{your_bu_username}”
- In the “/projectnb” directory, there are no user-specific space limitations → Install all your packages here → Instructions - Next Slides…

Good to know commands

- To check your disk quota (keep this under 10GB):
  - 

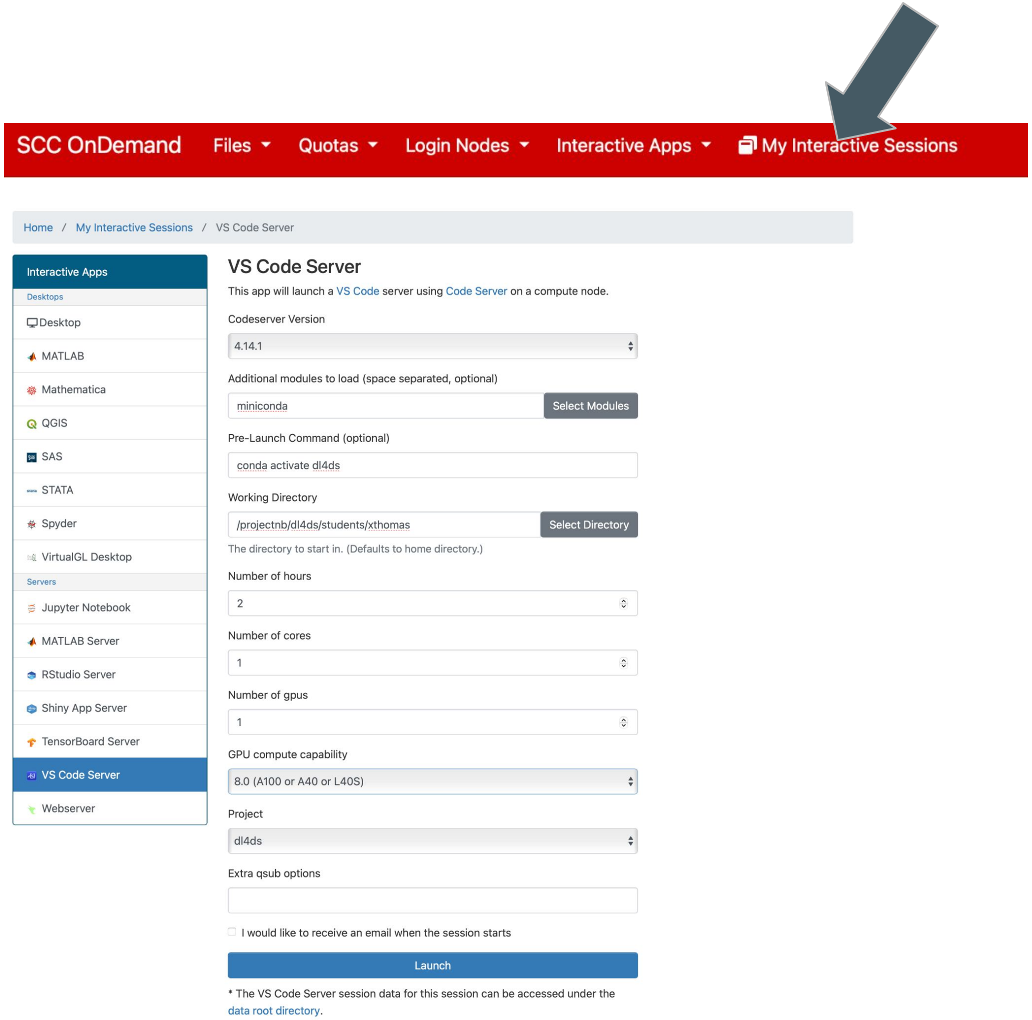
```
(install)[xthomas@scc1 dl4ds]$ quota -s
Home Directory Usage and Quota:
Name           GB    quota    limit in_doubt    grace |    files    quota    limit in_doubt    grace
xthomas   7.23624     10.0     11.0      0.0     none |   67,976  200,000  200,000        0     none
(install)[xthomas@scc1 dl4ds]$ 
```

- To check the project's disk quota:
  - 

```
(install)[xthomas@scc1 dl4ds]$ pquota dl4ds
                                      quota      quota      usage     usage
project space                          (GB)    (files)       (GB)   (files)
-----------------------------------  ------  ---------  ---------  --------
/projectnb/dl4ds                       1000   32768000       0.00        65
(install)[xthomas@scc1 dl4ds]$ 
```

Types of Jobs

1. <u>Interactive</u>
    a. Can type, view output, open files, run commands
    b. Follow the image to the right to spin up an interactive VS Code Server on your browser
    c. Make number of gpus to 0 to decrease wait time / decrease the compute capability of the gpu requested (denoted by the number 8.0)



VS Code Server

This app will launch a VS Code server using Code Server on a compute node.

Codeserver Version: 4.14.1

Additional modules to load (space separated, optional): miniconda

Pre-Launch Command (optional): conda activate dl4ds

Working Directory: /projectnb/dl4ds/students/xthomas

The directory to start in. (Defaults to home directory.)

Number of hours: 2

Number of cores: 1

Number of gpus: 1

GPU compute capability: 8.0 (A100 or A40 or L40S)

Project: dl4ds

Extra qsub options:

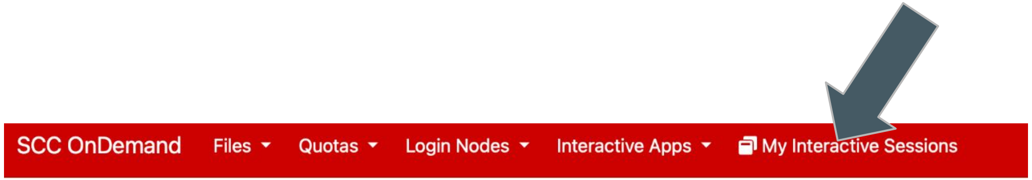
□ I would like to receive an email when the session starts

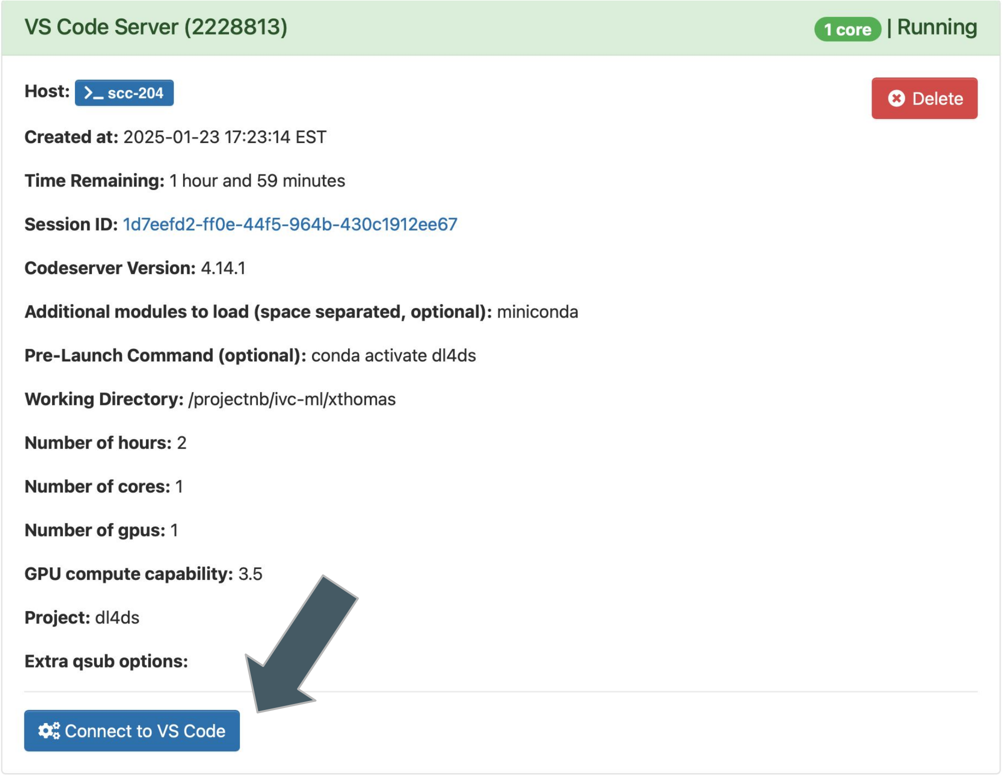
* The VS Code Server session data for this session can be accessed under the data root directory.

Types of Jobs

1. Interactive

   a. After a couple of minutes, you should be seeing this
   b. Click on Connect to VS Code




VS Code Server (2228813)

Host: scc-204

Created at: 2025-01-23 17:23:14 EST

Time Remaining: 1 hour and 59 minutes

Session ID: 1d7eefd2-ff0e-44f5-964b-430c1912ee67

Codeserver Version: 4.14.1

Additional modules to load (space separated, optional): miniconda

Pre-Launch Command (optional): conda activate dl4ds

Working Directory: /projectnb/ivc-ml/xthomas

Number of hours: 2

Number of cores: 1

Number of gpus: 1

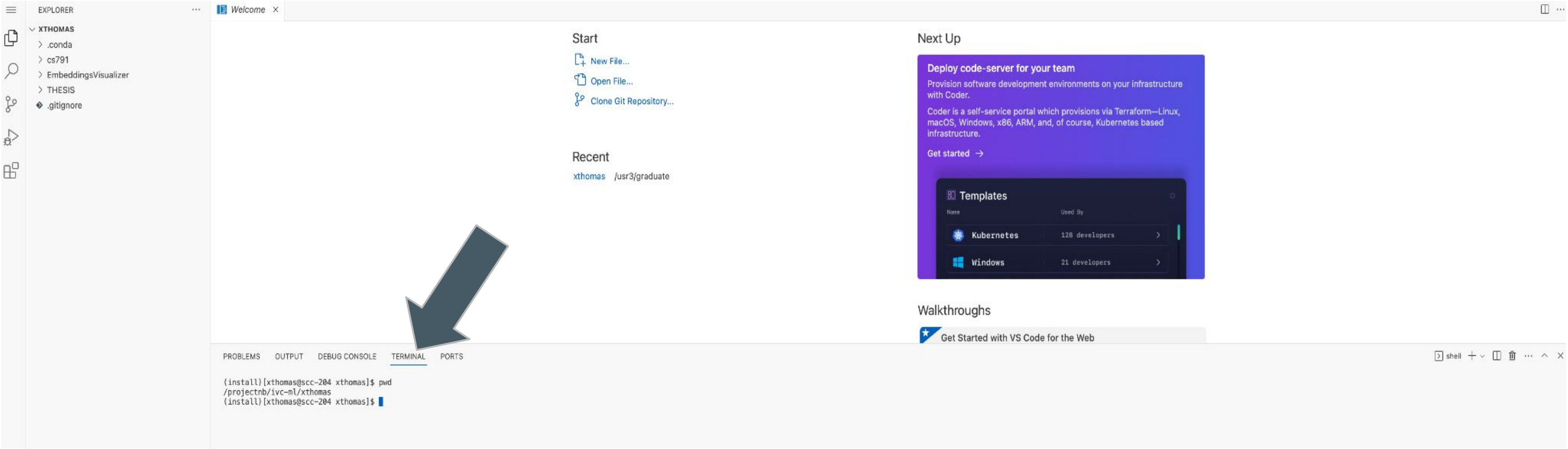
GPU compute capability: 3.5

Project: dl4ds

Extra qsub options:

Interactive Session (VSCode Server)

- You’ll see a VSCode window open in your browser
- Use the terminal for the next steps



Deploy code-server for your team

Provision software development environments on your infrastructure with Coder.

Coder is a self-service portal which provisions via Terraform—Linux, macOS, Windows, x86, ARM, and, of course, Kubernetes based infrastructure.

Get started →

```
(install)[xthomas@scc-204 xthomas]$ pwd
/projectnb/ivc-ml/xthomas
(install)[xthomas@scc-204 xthomas]$ 
```

Modules on SCC

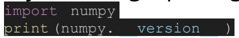
- What? - Collection of Software available to use
- Why? - For ease of use of different versions of standard packages
- How? -
  - Type “module avail“ – to see all available packages
  - “module load {module_load}” – to load the module

- Type “module load miniconda” – to load miniconda to your env → to create separate virtual environments for your projects, where you can pip install packages

- Follow the next slide after “module load miniconda”

Things to Note

- Your **home directory** only has **10GB** of storage. Ideally, do not install anything there. Work in the /projectnb directory.
- Recommended steps to create a conda environment **in SCC**:
  - module load miniconda – may give you – (if you get a "`WARNING: You do not have a .condarc file in your home directory`" message, run `setup_scc_condarc.sh` see this link for more instructions: https://www.bu.edu/tech/support/research/software-and-programming/common-languages/python/python-software/miniconda-modules/)
  - Check if "conda config --show pkgs_dirs" returns "/projectnb/...". Else do the below:
    - `conda config --add pkgs_dirs /projectnb/dl4ds/students/{Your_Folder_Name}/.conda`
    - `conda config --add envs_dirs /projectnb/dl4ds/students/{Your_Folder_Name}/.conda`
    - `conda config --show pkgs_dirs` - To confirm
  - By doing the "config –add" commands, the .conda path with "/projectnb/..." should be on top in the `~/.condarc` file. (You could just open this file and add the paths there, skipping the above step, use cat ~/.condarc to view the file) – this is done, so that whenever you do a "pip install" inside a conda env – the packages get installed into to this directory.
  - `cd /projectnb/l4ds/students/{Your_Folder_Name}`
  - Create an environment using, "`conda create -n dl4ds python=3.9`"
  - `conda activate dl4ds`
  - You can then install packages using conda install..., pip install..., etc. Ref: https://www.bu.edu/tech/support/research/software-and-programming/common-languages/python/python-installs/conda/
  - Try installing a package (example: pip install numpy), and try importing it:



```
import numpy
print(numpy.__version__)
```

- If you hit your home quota limit follow this: https://www.bu.edu/tech/support/research/system-usage/using-file-system/storage-quotas/

Types of Jobs

1. Interactive:

    a. For setting up code, debugging etc

    b. Once your session ends / gets interrupted, all runs get stopped.

2. <u>Batch (Non-Interactive)</u>

    Assuming you’re either on a VS Code server on your browser, or on your local IDE remotely connected to SCC:

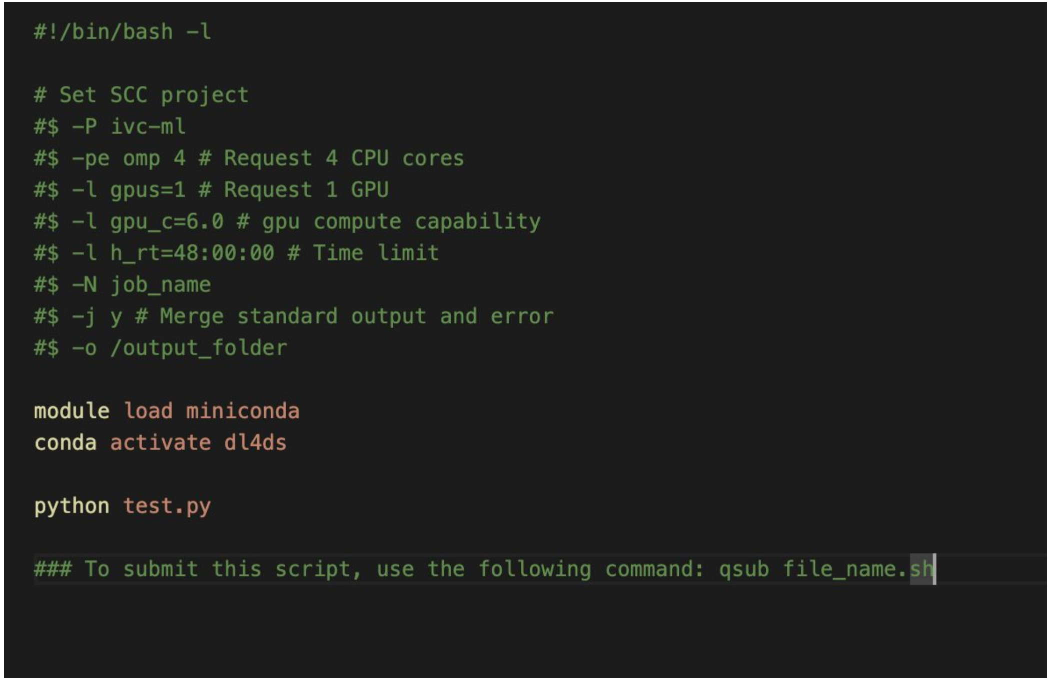
    a. Use it for long training runs, or to start multiple runs

    b. Create a bash script (like the image to the right) – name it example.sh

    c. To submit the job: run “qsub example.sh” – To view your job: “qstat -u {your_usernmae}”

    d. Your job is sent to the scheduler, who will put your job onto a queue, and then when resources free up – it will start running your script

    e. Running the qsub command is all that you need to do, the run will take place even if you are offline/away. – You will start seeing the output logs in the -O directory you define in the script when the job starts running.

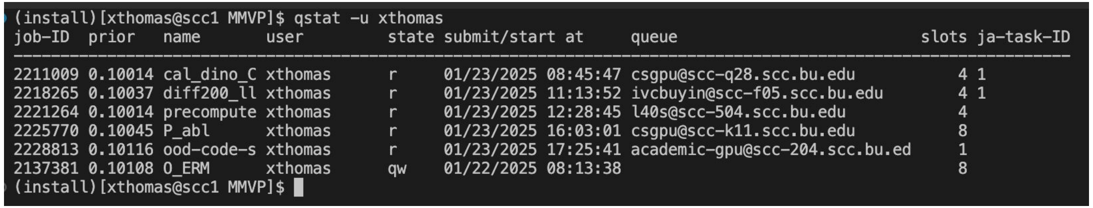


```
#!/bin/bash -l

# Set SCC project
#$ -P ivc-ml
#$ -pe omp 4 # Request 4 CPU cores
#$ -l gpus=1 # Request 1 GPU
#$ -l gpu_c=6.0 # gpu compute capability
#$ -l h_rt=48:00:00 # Time limit
#$ -N job_name
#$ -j y # Merge standard output and error
#$ -o /output_folder

module load miniconda
conda activate dl4ds

python test.py

### To submit this script, use the following command: qsub file_name.sh
```

View the queue

1. Type “qstat -u {your_username}”

2. job-ID - Unique name for your job
3. prior - Priority set by the Job scheduler
4. name - You can set names to your job by passing the -N option to the qsub command
5. state - The state of the job: (r) – running; (qw) – waiting to run; (hqw) – on hold, waiting to run; (Eqw) – job in error state; (s) – suspended; (t) – transfering.
6. queue - The queue name and the node ID on which the job is running.
7. slots - number of slots the job requested. (number of CPU cores)
8. ja-task-ID - task id if submitted a job array (job array – submitting multiple jobs under a single job submission where each job has its own task ID, allowing the system to manage and execute each task individually).



```
(install)[xthomas@scc1 MMVP]$ qstat -u xthomas
job-ID  prior   name       user         state submit/start at     queue                          slots ja-task-ID
-----------------------------------------------------------------------------------------------------------------
2211009 0.10014 cal_dino_C xthomas      r     01/23/2025 08:45:47 csgpu@scc-q28.scc.bu.edu           4 1
2218265 0.10037 diff200_ll xthomas      r     01/23/2025 11:13:52 ivcbuyin@scc-f05.scc.bu.edu        4 1
2221264 0.10014 precompute xthomas      r     01/23/2025 12:28:45 l40s@scc-504.scc.bu.edu            4
2225770 0.10045 P_abl      xthomas      r     01/23/2025 16:03:01 csgpu@scc-k11.scc.bu.edu           8
2228813 0.10116 ood-code-s xthomas      r     01/23/2025 17:25:41 academic-gpu@scc-204.scc.bu.ed     1
2137381 0.10108 O_ERM      xthomas      qw    01/22/2025 08:13:38                                    8
(install)[xthomas@scc1 MMVP]$ 
```

SCC Resources

Take a look at the SCC cheat sheet here:

https://dl4ds.github.io/sp2025/materials/

- Use either the Interactive Sessions (VSCode Server), or SSH into SCC via your local VSCode IDE to work on your code.

SCC - Requesting an Interactive Job (via Terminal)

- Ideal for debugging and you need GPUs

- Once you've logged in, and are on your terminal – you are now using a login node.

- To request for a compute node – "`qrsh -pe omp 4 -P dl4ds -l gpus=1`" – to see all the args that can be passed and what they mean, see https://dl4ds.github.io/sp2025/materials/

- Once the request goes through, you'll be taken to a compute node. Note that you'll be taken to your home directory. So do a "`cd /projectnb/ds542/students/{your_username}`" or "File → Open Folder…" in VSCode and enter this path.

- You can create your directories under the "students/{your_username}" directory – again make sure to create files, directories, etc under "/projectnb"

- Try "nvidia-smi" – to see the GPU assigned and to monitor GPU usage. – You may be assigned specific GPU ids, and notice that some of the GPU cores are already being used. There's no need for additional configuration—SCC automatically sets the CUDA_VISIBLE_DEVICES environment variable for you. You can proceed with your code as normal.