

Last updated: May 2025

Last updated: May 2025

AI Cluster - Slurm

Please send in a ticket requesting to be added if it is your first time using the AI cluster. You will need to be involved in research with a CS faculty member.

Feedback is requested. Find us in the Slack ai-cluster channel (channel ID: C02KW3M0BDK).

Infrastructure

Summary of nodes installed on the cluster:

AI Cluster Specs

CPU Cores: 2960 System Mem: 34389 GB GPU Memory: 8032 GB GPUs:

92	A40	48GB
52	L40S	48GB
14	H100	80GB

Storage: 483 TB

Computer/GPU Nodes

We like the alphabet, so we have compute node groups for just about every letter in it.

- "a" series: 3 nodes, each with 64 CPU threads, 192GB RAM, four RTX2080ti GPUs
- "aa" series: 2 nodes, each with 32 CPU threads, 32GB RAM, four RTX2080 GPUs
- "b", "d", "e", "k", "r" series: 15 nodes, each with 64 CPU threads, 512GB RAM, four A40's
- "c" series: 1 node with 48 CPU threads, 64GB RAM, two A30's
- "f" & "j" series: 6 nodes, each with 32 CPU threads, 128GB RAM, four A40's
- "g" & "q" series: 4 nodes, each with 96 CPU threads, 1TB RAM, eight L40S GPUs
- "h" series: 1 node with 96 CPU threads, 1TB RAM, four H100 SXM GPUs
- "l" series: 1 node with 256 CPU threads, 1.5TB RAM, six H100 PCI GPUs
- "m" series: 3 nodes with 128 CPU threads, 1.5TB RAM, no GPU's
- "n" series: 1 node with 96 CPU threads, 1.5TB RAM, four H100 SXM GPUs
- "t" series: 5 nodes with 48 CPU threads, 512GB RAM, four L40S GPUs
- all compute nodes:

- Each node has a /local space for times when it's beneficial to not write over NFS. Space in /local varies from node to node. Please try to clean up when you're done.
- Home directories and project space are mounted over NFS. Default quota for home directories is 50GB, but it may be increased as needed with permission.
- Research groups may additionally be allocated project space that exists outside the home directory quota on different storage servers, for collaboration and shared storage.

Storage

- ai-storage1:
 - 63T total storage
 - uplink to cluster network: 25G
 - /home/<username>
 - 50G quota per user.
- ai-storage2:
 - 63T total storage
 - uplink to cluster network: 2x 25G
 - /net/scratch: Create yourself a directory /net/scratch/\$USER. Use it for whatever you want.
 - Eventually data will be auto deleted after X amount of time. Maybe 90 days or whatever we determine makes sense.
- ai-storage3:
 - zfs mirror with previous snapshots of ai-storage1 and ai-storage4.
 - NOT a backup.
- ai-storage4:
 - 70TB total storage
 - uplink to cluster network: 10G
 - /net/projects:
 - Idea would be to create a dataset with a quota for people in a collaboration group to use.
 - Normal LDAP groups that you are used to and available everywhere else would control access to these directories. e.g. jonaslab, sandlab
- peanut-storage1:
 - 273TB total storage
 - uplink to cluster network: 25G fiber
 - /net/bulk:
 - A nice place for large datasets that either don't change much, or are being used and re-used a lot.
- peanut-storage3:
 - 224TB total storage
 - uplink to cluster network: 100G fiber
 - /net/projects2:
 - Even more project space for your projects that you can put your projects in.

Login

There are a set of front end nodes (currently, fe01 and fe02) that give you access to the Slurm cluster. You will connect through these nodes and need to be on these nodes to submit jobs to the cluster.

If you ssh to just "fe", it will pick one for you.

```
ssh cnetid@fe.ai.cs.uchicago.edu
```

File Transfer

You will use the FE nodes to transfer your files onto the cluster storage infrastructure over SSH, with a tool like rsync or scp.

Demo

kauffman3 is my CS test account.

```
$ ssh kauffman3@fe.ai.cs.uchicago.edu
```

I've created a couple scripts that run some of the Slurm commands but with more useful output. cs-sinfo and cs-squeue being the only two right now.

```
kauffman3@fe01:~$ cs-sinfo
```

NODELIST	NODES	PARTITION	STATE	CPUS	S:C:T	MEMORY	TMP_DISK	WEIGHT
a[001-006]	6	geforce*	idle	64	2:16:2	190000	0	1
'turing,geforce,rtx2080ti,11g'			none		gpu:rtx2080ti:4			
a[007-008]	2	quadro	idle	64	2:16:2	383000	0	1
'turing,quadro,rtx8000,48g'			none		gpu:rtx8000:4			

```
kauffman3@fe01:~$ cs-squeue
```

JOBID	PARTITION	USER	NAME	NODELIST
TRES_PER_NSTATE	TIME			

Run my job script on four GPUs in "f" series systems 1-4, using the Slurm partition your research group has access to. (Ask your advisor if there is such a special partition available to you.)

```
kauffman3@fe01:~$ sbatch -p my-advisors-partition --gres=gpu:4 -w f[001-004] job.sh
```

Refer to 'man sbatch' and 'man srun' for more.

Asked Questions

Do we have a max job runtime?

Yes, 4 hours. This is done per partition. You are expected to write your code to accommodate for this.

Jupyter Notebook Tips

Batch

The process for a batch job is very similar.

jupyter-notebook.sbatch

```
#!/bin/bash
unset XDG_RUNTIME_DIR
NODEIP=$(hostname -i)
NODEPORT=$(( $RANDOM + 1024 ))
echo "ssh command: ssh -N -L 8888:$NODEIP:$NODEPORT
`whoami`@fe01.ai.cs.uchicago.edu"
. ~/myenv/bin/activate
jupyter-notebook --ip=$NODEIP --port=$NODEPORT --no-browser
```

Check the output of your job to find the ssh command to use when accessing your notebook.

Make a new ssh connection to tunnel your traffic. The format will be something like:

```
ssh -N -L 8888:###.###.###.###:#### user@fe01.ai.cs.uchicago.edu
```

This command will appear to hang since we are using the -N option which tells ssh not to run any commands including a shell on the remote machine.

Open your local browser and visit: <http://localhost:8888>

Interactive

1. `srun --pty bash` run an interactive job
2. `unset XDG_RUNTIME_DIR` jupyter tries to use the value of this environment variable to store some files, by default it is set to ' ' and that causes errors when trying to run jupyter notebook.
3. `export NODEIP=$(hostname -i)` get the ip address of the node you are using
4. `export NODEPORT=$(( $RANDOM + 1024 ))` get a random port above 1024
5. `echo $NODEIP:$NODEPORT` echo the env var values to use later
6. `jupyter-notebook --ip=$NODEIP --port=$NODEPORT --no-browser` start the jupyter notebook
7. Make a new ssh connection with a tunnel to access your notebook
8. `ssh -N -L 8888:$NODEIP:$NODEPORT user@fe01.ai.cs.uchicago.edu` using the values not variables

9. This will make an ssh tunnel on your local machine that forwards traffic sent to `localhost:8888` to `$NODEIP:$NODEPORT` via the ssh tunnel. This command will appear to hang since we are using the `-N` option which tells ssh not to run any commands including a shell on the remote machine.
10. Open your local browser and visit: `http://localhost:8888`

Copy the following code snippet to the interactive node directly:

```
unset XDG_RUNTIME_DIR
NODEIP=$(hostname -i)
NODEPORT=$(( $RANDOM + 1024 ))
echo "ssh command: ssh -N -L 8888:$NODEIP:$NODEPORT
`whoami`@fe01.ai.cs.uchicago.edu"
jupyter-notebook --ip=$NODEIP --port=$NODEPORT --no-browser
```

Contribution Policy

This section can be ignored by most people. [If you contributed to the cluster or are in a group that has you can read more here.](#)

From:
<https://howto.cs.uchicago.edu/> - **How do I?**

Permanent link:
<https://howto.cs.uchicago.edu/slurm:ai?rev=1748621066>

Last update: **2025/05/30 11:04**

