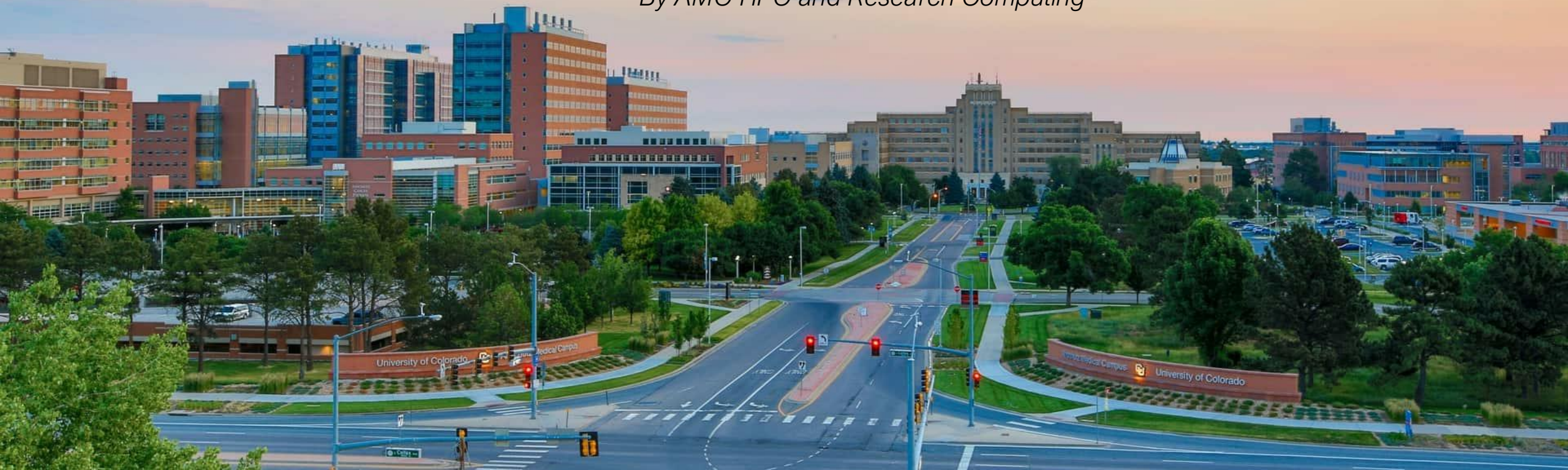
University of Colorado **Anschutz Medical Campus**

Alpine Best Practices Workshop

By AMC HPC and Research Computing



If you do not currently have an active Alpine account you can request one for free!

https://cuanschutz.edu/hpc

High Performance Computing (HPC)

Category: Computer and Device Support, High Performance Computing, Productivity and Business
Audience: Faculty, Researchers, Staff, Students

OIT and the School of Medicine (SOM) are partnering with CU Boulder Research Computing (CURC) to provide compute resources for researchers on the CU Anschutz campus.

Request Access

- You must have a valid cuanschutz.edu email address (we do not take ucdenver.edu email addresses; please request a cuanschutz.edu from your departmental administration team)

- You must have your ACCESS/XSEDE ID. If you do not have one, you can create one for free at: https://identity.access-ci.org/new-user

Reminder! Alpine and PetaLibrary cannot be used for highly sensitive data sets. Not sure about your data set? Please ask us!

- No HIPAA or Protected Health Information (PHI)
- No General Data Protection Regulation (GDPR)
- No CMMC
- No FERPA
- No data that requires special security and compliance

FERPA

Family Educational Rights & Privacy Act

EUROPEAN GENOME-PHENOME ARCHIVE

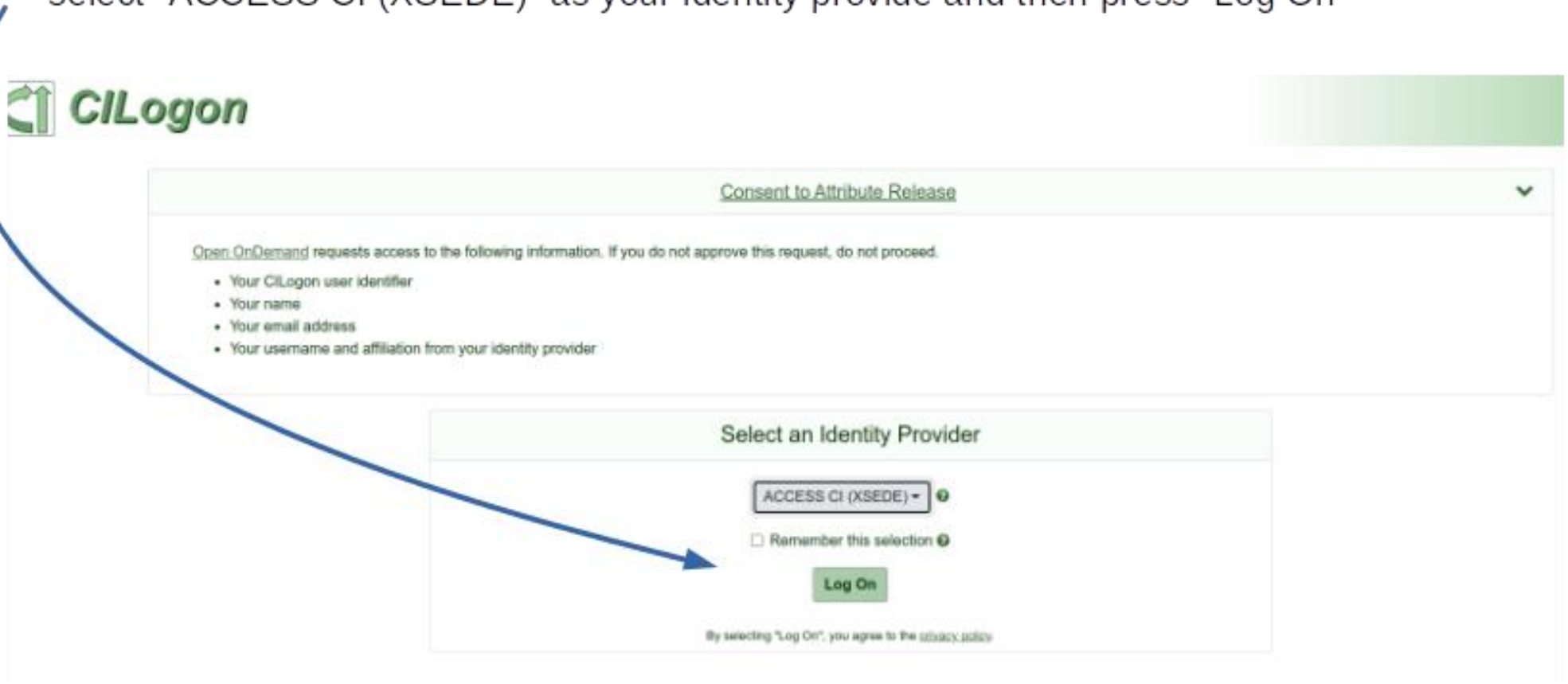
University of Colorado Anschutz Medical Campus

All data on Alpine must be 100% de-identified and please make sure it is compliant with your IRB, DUA, data repository requirements, and/or grant funding agency

A refresher on how to log into Alpine

1. Go to https://ondemand-rmacc.rc.colorado.edu

2. It should redirect you to CILogon which is how you authenticate your Alpine session. Make sure you select “ACCESS CI (XSEDE)” as your identity provide and then press “Log On”

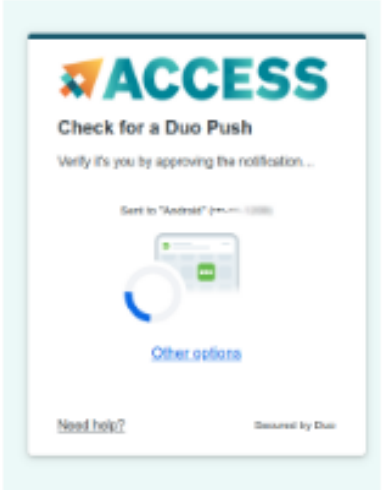


A refresher on how to log into Alpine

3 Next it will take you to this page where you will put in your ACCESS ID and ACCESS password and press Login. This is NOT your CU Anschutz ID!!

ACCESS ID and ACCESS password **not** CU Anschutz credentials!!

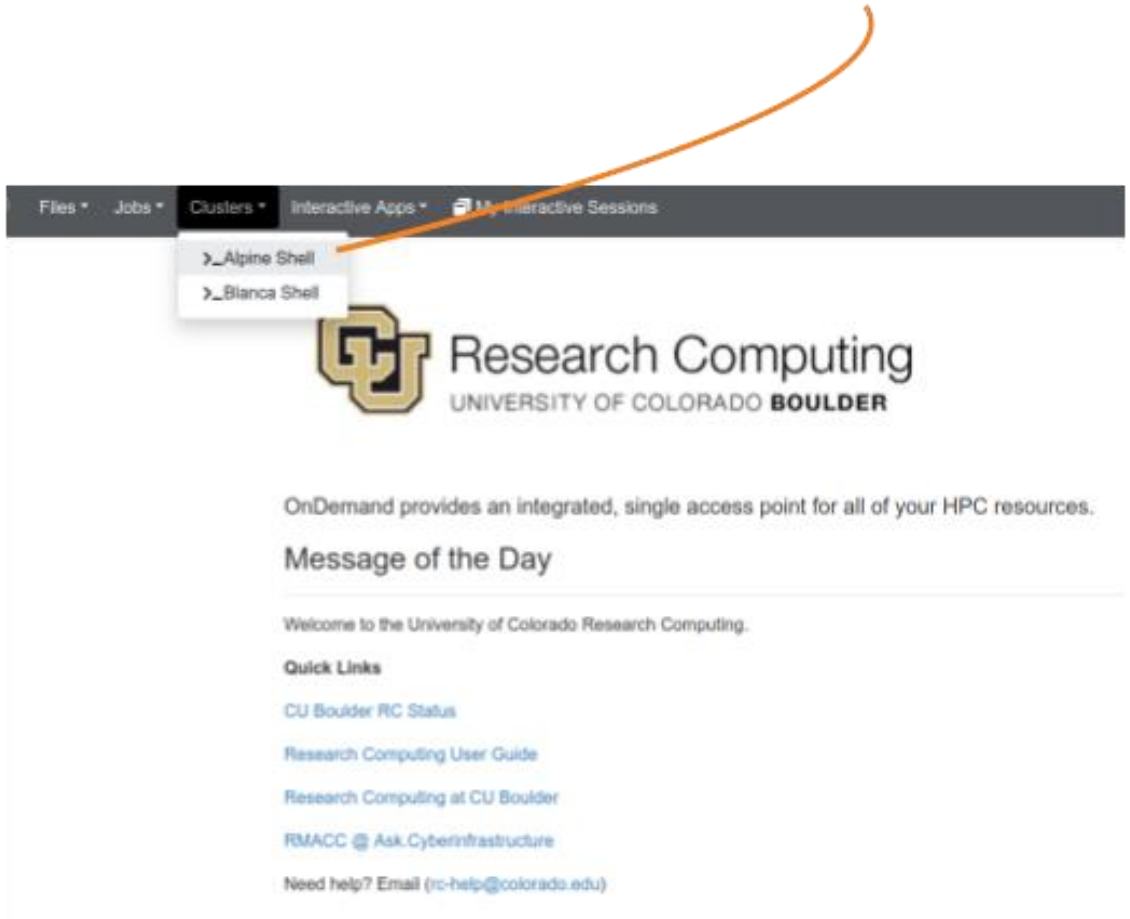
4 This will prompt a DUO MFA push to your phone or whichever way you have DUO set up on your phone to authenticate. Accept the push sent to your device.



University of Colorado
Anschutz Medical Campus

A refresher on how to log into Alpine

5 This will take to the official CURC page. Let's select the Alpine shell.



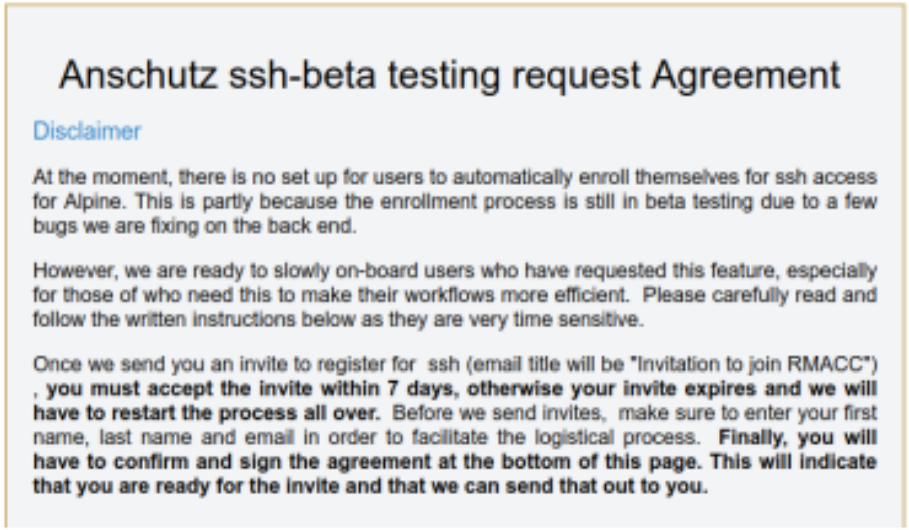
6 This will log you into the head node of Alpine. It will **always default to your home directory**.

Users can apply for ssh access to Alpine and we will notify applicants to attend one of our monthly meetings to configure your account

The benefits of having ssh access:

- Logging into Alpine can be done on the terminal via ssh; no need for the Open onDemand portal
- Access to transfer protocols such as rsync, scp, and sftp are available to you and therefore you can bypass globus and any globus related issues

University of Colorado
Anschutz Medical Campus



Anschutz ssh-beta testing request Agreement

Disclaimer

At the moment, there is no set up for users to automatically enroll themselves for ssh access for Alpine. This is partly because the enrollment process is still in beta testing due to a few bugs we are fixing on the back end.

However, we are ready to slowly on-board users who have requested this feature, especially for those of who need this to make their workflows more efficient. Please carefully read and follow the written instructions below as they are very time sensitive.

Once we send you an invite to register for ssh (email title will be "Invitation to join RMACC") , **you must accept the invite within 7 days, otherwise your invite expires and we will have to restart the process all over.** Before we send invites, make sure to enter your first name, last name and email in order to facilitate the logistical process. **Finally, you will have to confirm and sign the agreement at the bottom of this page. This will indicate that you are ready for the invite and that we can send that out to you.**

https://ucdenverdata.formstack.com/forms/alpine_ssh_request_form

If you do not have ssh access there are many other methods you can use to transfer data

- Globus for large data transfers

  https://curc.readthedocs.io/en/latest/compute/data-transfer.html#data-transfer

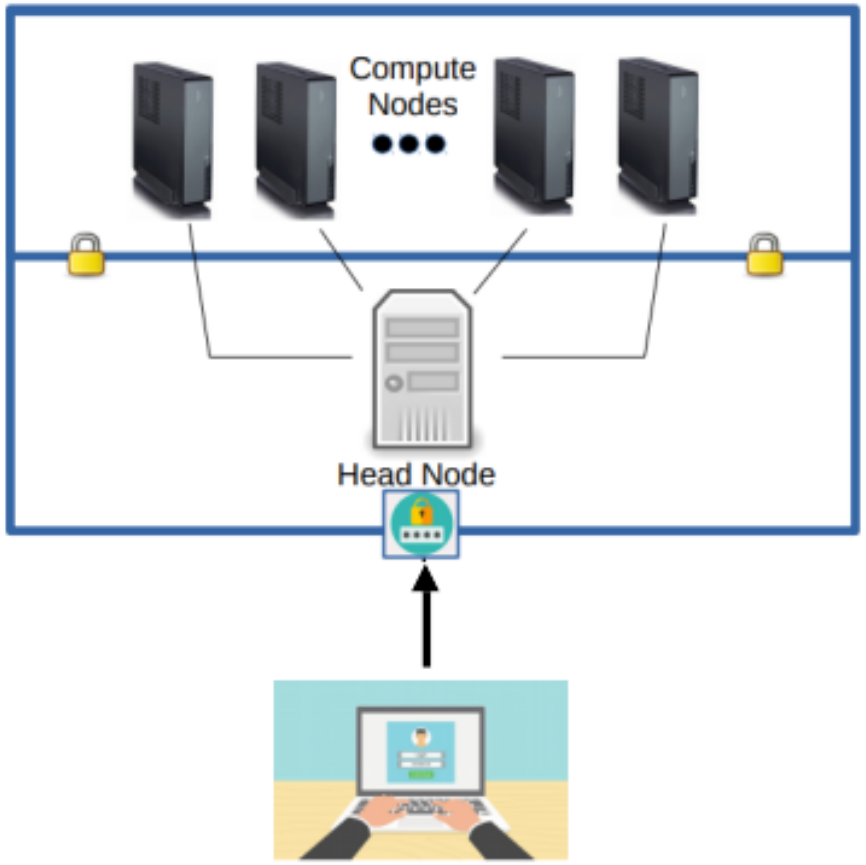
- Open OnDemand for small data transfers

- Rclone on OneDrive

  https://curc.readthedocs.io/en/latest/storage/petalibrary/onedrive.html#using-rclone

Please keep in mind, sometimes our networks become overwhelmed on globus. In that case using ssh or Rclone to transfer data will help alleviate these bottlenecks so you can continue to transfer data regardless if the globus endpoint is being bogged down.

Head node vs Compute Node: never run software, installs, or analysis on a head node!



The compute nodes are where all the power of the HPC is harnessed. **This is where all the compute nodes are located**. On Alpine, this is where your CPU nodes (amilan, amem, ahub, acompile) and GPU nodes and located (aa100, ami100, al40, atesting_a100, atesting_mi100)

Sometimes referred to as the **login node**. Its main purpose is to take your job requests and find compute resources for you to use.

This node is the master or brain of the HPC.

Compute Node	Head/Login Node
Any really long (> few minutes) and/or large (>500MB of memory) programs or calls needs to be run on compute nodes. The compute nodes wait for the head node to assign them a job or task. Software compiling and installations	The head node is very small and limited in ability. Its main function is to work with the scheduler and delegate tasks and jobs to the compute nodes that users want to utilize Strictly used to navigate the Alpine file system with basic linux commands or submit jobs to the scheduler using sbatch

Rule of Thumb

Always request a compute node for running any scripts or software related tasks, *including compiling and installing local software*

How can I tell if I am on the head node or a compute node?

Anything that says `login-` tells **you are on the login/head node**

You **can:**

- submit jobs using sbatch,
- use basic Linux commands (`mv`, `cd`, `rm`, `ls`, etc...),
- use slurm commands to check job status and efficiency (`seff`, `seff-array`, `jobstats`, `sacct`, `sstat`)
- Use slurm commands to request interactive use of compute resources (`acompile`, `sinteractive`)
- Use text editors (nano, emacs, vi/vim) that are very low memory (i.e. do not open large (>500 Mb) files)

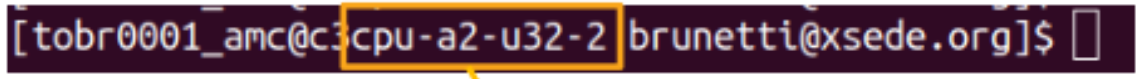
Please **do not:**

- run software
- excecute code or scripts
- install custom software

You are on a compute node if...

Anything that says `gpu-` tells you are on the inside of a GPU node in an interactive session

It is safe to run any software, installs, programs, code here.



Anything that says `cpu-` tells you are on the inside of a CPU node in an interactive session

It is safe to run any software, installs, programs, code here.

Job efficiency matters – it costs the campus available service units

Although, you as users, do not have a personal financial cost associated for using Alpine (with the exception of PetaLibrary), there is a cost to reserving resources. The currency we use are called **service units (SUs)**.

Compute *requested** **SUs charged**

Every time you request a set amount of compute resources, that costs us a portion of SUs we have available to the Anschutz Alpine community. We have a finite amount of SUs available.

IMPORTANT NOTE

We are charged, from our finite pool of SUs, **based on the resource reservation requested, regardless if you use those resources efficiently or not**. Therefore, we want to make sure that when you request large amounts of resources you are using those resources efficiently so that we don't drain our pool of community SUs unnecessarily, resulting in everyones' jobs having longer wait times.

The SU CPU currency has the following **approximate** conversion rate

As of March 2025, we have 5,195 total SUs available per hour for our entire campus to use which equates to roughly 124,680 SUs available in a given 24 hour period.

CPU nodes	cost
amilan CPU nodes (partition = amilan)	1 SU/core/hour
High memory CPU nodes (partition = amem)	4 SUs/core/hour

Example calculation

One typical amilan CPU node can have up to 64 cores, so if you request a single amilan CPU node and all 64 cores and your job runs for a full 24 hours we are charged:

1 amilan node x 64 cores x 1 SU per core hour x 24 hours = 1,536 SUs charged for your job

On Alpine, `seff` allow you to check the efficiency of CPU based jobs once they are complete

```
[brunetti@xsede.org@login-ci2 brunetti@xsede.org]$ seff 10873469
```

Your job ID

```
Job ID: 10873469
Cluster: alpine
User/Group: tobr0001_amc/brunettipgrp@xsede.org
State: COMPLETED (exit code 0)
Nodes: 1
Cores per node: 27
CPU Utilized: 22:54:40
CPU Efficiency: 5.89% of 16-05:14:06 core-walltime
Job Wall-clock time: 14:24:58
Memory Utilized: 73.04 GB
Memory Efficiency: 73.04% of 100.00 GB
```

Goal: ensure either CPU efficiency **or** memory efficiency is at least 70% or better

University of Colorado
Anschutz Medical Campus

What did job ID 10873469 cost us in SUs?

```
Job ID: 10873469
Cluster: alpine
User/Group: tobr0001_amc/brunettipgrp@xsede.org
State: COMPLETED (exit code 0)
Nodes: 1
Cores per node: 27
CPU Utilized: 22:54:40
CPU Efficiency: 5.89% of 16-05:14:06 core-walltime
Job Wall-clock time: 14:24:58
Memory Utilized: 73.04 GB
Memory Efficiency: 73.04% of 100.00 GB
```

We know we used 27 cores

We know we used it for about 14 hours and 24 minutes

However, what we don't know is whether this was an amilan node or an amem high memory node. Luckily, we can use `sacct` to grab the script associated with this job ID.

```
[brunetti@xsede.org@login-ci2 brunetti@xsede.org]$ sacct -B -j 10873469
```

Print the batch script associated with the listed jobID

Parameter to tell SLURM the following value is a jobID

Your job ID

What did job ID 10873469 cost us in SUs?

I requested amilan nodes, so now we can calculate the SUs.

```
Batch Script for 10873469
--------------------------------------------------------------------------------
#!/bin/bash

#SBATCH --nodes=1 # use one node
#SBATCH --time=5-00:00:00 #5 days
#SBATCH --account=amc-general # normal, amc, long, mem (use mem when using the amem partition)
#SBATCH --partition=amilan # amilian, ami100, aa100, amem, amc
#SBATCH --qos=long
#SBATCH --ntasks=25 # total processes/threads
#SBATCH --job-name=cellranger
#SBATCH --output=step1_cellranger_LI047_gex_out_01062025_%J.log
#SBATCH --mem=100G # suffix K,M,G,T can be used with default to M
#SBATCH --mail-user=tonya.brunetti@cuanschutz.edu
#SBATCH --mail-type=END
#SBATCH --error=step1_cellranger_LI047_gex_out_01062025_%J.err

cellranger="/projects/$USER/software/cellranger-7.0.1/cellranger"
id="step1_gex_output"
inputcsv="/scratch/alpine/tobr0001_amc/[redacted]cite_vdjB_hto_LI047_noLIBRA_01022024/step1_input_csv_cellranger_multi.csv"

${cellranger} multi --id=${id} --csv=${inputcsv} --disable-ui
```

1 amilan node x 27 cores x 1 SU per core hour x 14.5 hours = ~391.5 SUs charged for the job

Notice the cores come from the `seff` command, not necessarily what we requested which was 25 – why?

We are only charged for the time we use for the job to complete. Notice even though I requested a 5 day run, the job finished in 14.5 hours so we are charged for 14.5 hours

Our core architecture is tied directly to shared RAM, so SLURM takes the ceiling of the requested resources

https://curc.readthedocs.io/en/latest/clusters/alpine/alpine-hardware.html#partitions

Partition	Description	# of nodes	cores/node	RAM/core (GB)	Billing_weight/ core	Default/Max Walltime	Resource Limits
amilan	AMD Milan (default)	386	32 or 48 or 64	3.75	1	24H, 7D	see qos table
ami100	GPU-enabled (3x AMD MI100)	8	64	3.75	6.1[3]	24H, 7D	15 GPUs across all jobs
aa100	GPU-enabled (3x NVIDIA A100)[4]	12	64	3.75	6.1[3]	24H, 7D	21 GPUs across all jobs
al40	GPU-enabled (3x NVIDIA L40)[4]	3	64	3.75	6.1[3]	24H, 7D	6 GPUs across all jobs
amem[1]	High-memory	24	48 or 64 or 128	16[2]	4.0	4H, 7D	128 cores across all jobs

I requested 25 cores and 100 GB of RAM in my script, however, each core on an amilan node has 3.75 GB of RAM associated with it.

Calculation #1: 25 cores requested x 3.75 GB RAM = 93.75GB of RAM available,
but I requested 100GB, so SLURM adds the minimum number of cores to reach 100GB of RAM.

Calculation #2: 100GB RAM/3.75 = ~26.667, so it SLURM does me a favor and bumps me up to 27 cores to achieve my max memory. Note, that SLURM will kill my job however, if I exceed 100G of max RAM.

Let's take a look at the efficiency of job ID 4381567

```
[brunetti@xsede.org@login-ci2 brunetti@xsede.org]$ seff 4381567
```

```
Job ID: 4381567
Cluster: alpine
User/Group: tobr0001_amc/brunettipgrp@xsede.org
State: COMPLETED (exit code 0)
Nodes: 1
Cores per node: 14
CPU Utilized: 3-16:02:39
CPU Efficiency: 14.46% of 25-08:44:22 core-walltime
Job Wall-clock time: 1-19:28:53
Memory Utilized: 87.38 MB
Memory Efficiency: 0.17% of 50.00 GB
```

used 14 cores

Used resources for 1 day, 19 hours, and 28 minutes

Not good!
CPU efficiency **or** memory efficiency is at not at least 70% or better.

We need to optimize resources before submitting this job again in the future

Let's check the script and see which nodes partitons I requested and what resources I requested

```
[brunetti@xsede.org@login-ci2 brunetti@xsede.org]$ sacct -B -j 4381567
```

Let's take a look at the efficiency of job ID 4381567

I requested amilan nodes, so now we can calculate the SUs.

```
Batch Script for 4381567
--------------------------------------------------------------------------------
#!/bin/bash
#SBATCH --nodes=1 # use one node
#SBATCH --time=72:00:00
#SBATCH --account=amc-general # normal, amc, long, mem (use mem when using the amem partition)
#SBATCH --partition=amilan # amilian, ami100, aa100, amem, amc
#SBATCH --qos=long
#SBATCH --ntasks=5 # total processes/threads
#SBATCH --job-name=demuxTB
#SBATCH --mem=50G # suffix K,M,G,T can be used with default to M
#SBATCH --mail-user=tonya.brunetti@cuanschutz.edu
#SBATCH --mail-type=END
#SBATCH --output=ALAW-AS049-multiplexGroup_1-2_%J.log
#SBATCH --error=ALAW-AS049-multiplexGroup_1-2_%J.err
```

1 amilan node x 14 cores x 1 SU per core hour x 43.5 hours = ~609 SUs charged for the job

SLURM charged me for 14 cores instead of the 5 I requested because the memory I requeseted could not be achieved with just 5 cores:

Calculation #1: 5 cores requested x 3.75 GB RAM = 18.75 GB of RAM available,
but I requested 50GB, so SLURM adds the minimum number of cores to reach 50GB of RAM.

Calculation #2: 50GB RAM/3.75 = ~13.333, so it SLURM does me a favor and bumps me up to 14 cores to achieve my max memory. Note, that SLURM will kill my job however, if I exceed 50GB of max RAM.

How can I make better use of resources and use less SUs to essentially complete the same job while reducing resource waste?

```
Job ID: 4381567
Cluster: alpine
User/Group: tobr0001_amc/brunettipgrp@xsede.org
State: COMPLETED (exit code 0)
Nodes: 1
Cores per node: 14
CPU Utilized: 3-16:02:39
CPU Efficiency: 14.46% of 25-08:44:22 core-walltime
Job Wall-clock time: 1-19:28:53
Memory Utilized: 87.38 MB
Memory Efficiency: 0.17% of 50.00 GB
```

#SBATCH --ntasks=4
#SBATCH --mem =1G

A better optimized use of resources that likely **reduces the cost of SUs consumed by over 50%!**

Here, I only used 14.46% of CPUs efficiently, so technically I could have dropped this to 3 CPUs:

14 cores * .1446 = 2.02 cores; however let's pad it a bit to account for ineffiecienies to either 3 or 4 cores

50 GB * 0.0017 = ~0.085 GB used; well 1 core has 3.75 GB so we would have been fine on using just a single core.

University of Colorado
Anschutz Medical Campus

Arrays should never be used without running a representative small batch of samples first, in order to optimize efficiency of resources

Arrays are an easy way to parallelize upwards of thousands of jobs in a script; however, you can imagine that when slurm submits a 1000 jobs, if the job is already incredibly inefficient, then that now gets compounded over the number of jobs you submitted.

For job arrays, you can get a summarized view of all jobs in that array using:

```
seff-array yourJobID
```

Arrays should never be used without running a representative small batch of samples first, in order to optimize efficiency of resources

```
seff-array yourJobID
```

In this example, this array has 900 jobs associated with it and by running the above command on this jobID, you can see a summary of efficiency across all 900 jobs. Just make sure all jobs have completed! Any actively running jobs will not provide an accurate estimation of efficiency.

Goal: ensure either CPU efficiency **or** memory efficiency is at least 70% or better across at least 70-80% of all jobs submitted under that job array

```
Requested CPUs: 12 cores on 1 node(s)
Requested Memory: 45G
Requested Time: 07:00:00
--------------------------------------------------
Job Status
COMPLETED: 900
--------------------------------------------------
--------------------------------------------------
Finished Job Statistics
(excludes pending, running, and cancelled jobs)
Average CPU Efficiency 11.83%
Average Memory Usage 8.02G
Average Run-time 1146.37s
```

Here you can see what was requested for each job in the array: 12 CPU cores with a max memory of 45G. There were 900 jobs in this array therefore, **this request was made 900 separate times**

Here you can see that on average across all 900 jobs, **11.83% of CPUs were efficiently used**, and 8.02G/45G requested = **17.82% of available requested memory was efficiently used**. **Neither one of these is at least 70%!**

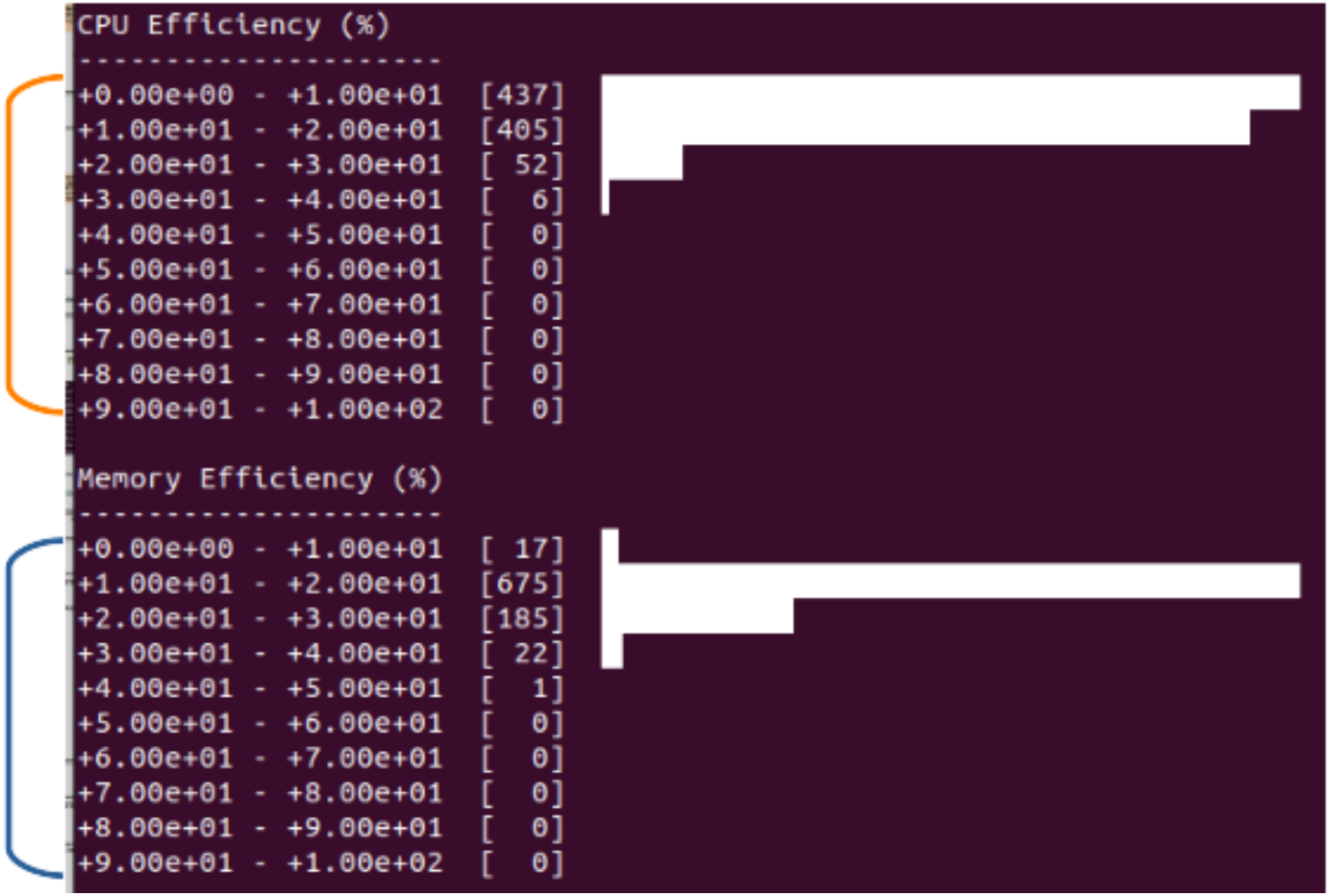
`seff-array` also provides a more granular detailed visual breakdown of CPU and memory efficiency across all completed jobs in your array

Not a single job had a CPU efficiency greater than 30-40%!

842/900 = 93.56% of jobs never exceed 20% CPU efficiency

Not a single job had a memory efficiency greater than 40-50%!

877/900 = 97.44% of jobs never exceeding 30% memory efficiency



The problem here is we are charged for the amount we reserved even though we did not need that amount

This was on an amilan node so the charge is 1 SU/1 core hour with 45G of memory with the average run time being ~20 minutes. This means we were charged roughly the following SUs per job:

1 amilan node x 12 cores x 1 SU per core hour x 0.3 hours = ~3.6 SUs charged for the job

However, we requested this 900 times:

~3.6 SUs per job x 900 jobs = 3,240 SUs charged

Based on our `seff-array` results we know that > 90% of the jobs could have been run withthin with 1/5 the requested CPUs and ~1/3 the amount of memory requested.

What could we have done to make our SUs go farther and our jobs are more efficient?

We should have scaled down our resources before submitting 900 jobs in an array

Not a single job had a CPU efficiency greater than 30-40%!

842/900 = 93.56% of jobs never exceed 20% CPU efficiency

Requested 12 cores but 20% of this would have covered 93.56% perfectly fine:

12 cores * .20 efficiency = 2.4 cores

The nearest ceiling core is 3.

Not a single job had a memory efficiency greater than 40-50%!

877/900 = 97.44% of jobs never exceed 30% memory efficiency

Requested 45 GB of RAM but 30% of this would have covered 97.44% perfectly fine:

45GB* .30 efficiency = 13.5 GB of RAM

13.5GB/3.75GB per core = 3.6 cores needed to achieve this RAM.

The nearest ceiling core is 4.

Take the max of these 2 values

Original Version (not correct)

Charged ~3,240 SUs for 900 jobs inefficiently used

```
#SBATCH --ntasks=12
#SBATCH --mem=45G
```

Corrected Version

```
#SBATCH --ntasks=4
#SBATCH --mem=15G
```

4 cores x 3.75GB per core = 15G RAM

Charged ~1,080 SUs for the same 900 jobs efficiently used

Best Practices for Using Arrays on Alpine

1) ALWAYS run a small subset of samples in an array first before scaling to your full dataset.

   Had the previously job just tested 20-40 samples first, the number of wasted resources would have been drastically reduced.

2) After your test run above, get efficiency statistics using `seff-array` and adjust your resource SBATCH directives accordingly.

3) Request resources based on what covers 70-80% of use cases, not the max! It is more efficient to run jobs mostly efficient and re-run the small number of job failures separately at a different resource amount.

Document your error and log files with unique Job IDs

The benefits of knowing what job ID are tied to error and log outputs is incredibly useful and makes troubleshooting malfunctioning code much easier

```
Batch Script for 10873469
--------------------------------------------------------------------------------
#!/bin/bash

#SBATCH --nodes=1 # use one node
#SBATCH --time=5-00:00:00 #5 days
#SBATCH --account=amc-general # normal, amc, long, mem (use mem when using the amem partition)
#SBATCH --partition=amilan # amilian, ami100, aa100, amem, amc
#SBATCH --qos=long
#SBATCH --ntasks=25 # total processes/threads
#SBATCH --job-name=cellranger
#SBATCH --output=step1_cellranger_LI047_gex_out_01062025_%J.log
#SBATCH --mem=100G # suffix K,M,G,T can be used with default to M
#SBATCH --mail-user=tonya.brunetti@cuanschutz.edu
#SBATCH --mail-type=END
#SBATCH --error=step1_cellranger_LI047_gex_out_01062025_%J.err
```

`%J` will automatically substitute the job ID into the error and log outputs.

This means you can always grab the script submitted with this job by using,

```
sacct -B -j yourJobID
```

as we have seen before, and that can help our team troubleshoot where the problems in your script may be if you reach out to us when a job fails

“How do I know what resources my job needs?”

```
#!/bin/bash


#SBATCH --job-name=job_scaling
#SBATCH --output=dump.%j.out
#SBATCH --account=amc-general
#SBATCH --nodes=1
#SBATCH --partition=amilan
#SBATCH --time=01:00:00
#SBATCH --ntasks=4
#   #SBATCH --mem=10G


## module load

echo "Job started on $(hostname) at $(date)"

# Say we want to sequence 1,000,000,000 numbers, but can start small with 1,000,000 first to see what that costs!
# Simple computation: simulate CPU load with a loop
echo "Running a simple computation..."
for i in $(seq 1 1000000); do
    result=$((result + i))
done

# Print the result
echo "Computation finished. Result: $result"

echo "Job completed at $(date)"
```

“How do I know what resources my job needs?”

```
[msherren@xsede.org@login-ci1 best_practices_scripts]$ seff 12535812
Job ID: 12535812
Cluster: alpine
User/Group: msherren@xsede.org/msherrenpgrp@xsede.org
State: COMPLETED (exit code 0)
Nodes: 1
Cores per node: 4
CPU Utilized: 00:00:03
CPU Efficiency: 8.33% of 00:00:36 core-walltime
Job Wall-clock time: 00:00:09
Memory Utilized: 308.00 KB
Memory Efficiency: 0.00% of 15.00 GB
```

“How do I know what resources my job needs?”

```
[msherren@xsede.org@login-ci1 best_practices_scripts]$ seff 12535812
Job ID: 12535812
Cluster: alpine
User/Group: msherren@xsede.org/msherrenpgrp@xsede.org
State: COMPLETED (exit code 0)
Nodes: 1
Cores per node: 4
CPU Utilized: 00:00:03
CPU Efficiency: 8.33% of 00:00:36 core-walltime
Job Wall-clock time: 00:00:09
Memory Utilized: 308.00 KB
Memory Efficiency: 0.00% of 15.00 GB
```

“How do I know what resources my job needs?”

```
[msherren@xsede.org@login-ci1 best_practices_scripts]$ seff 12535812
Job ID: 12535812
Cluster: alpine
User/Group: msherren@xsede.org/msherrenpgrp@xsede.org
State: COMPLETED (exit code 0)
Nodes: 1
Cores per node: 4
CPU Utilized: 00:00:03
CPU Efficiency: 8.33% of 00:00:36 core-walltime
Job Wall-clock time: 00:00:09
Memory Utilized: 308.00 KB
Memory Efficiency: 0.00% of 15.00 GB
```

From seff we learn:

- CPU usage and efficiency
- Memory usage and efficiency
- Wall time

Diagnosing a job that doesn't finish correctly

```
#!/bin/bash


#SBATCH --job-name=job_scaling
#SBATCH --output=dump.%j.out
#SBATCH --account=amc-general
#SBATCH --nodes=1
#SBATCH --partition=amilan
#SBATCH --time=00:05:00
#SBATCH --ntasks=4
#    #SBATCH --mem=10G


## module load

echo "Job started on $(hostname) at $(date)"

echo "Running a simple computation..."
# The loop below is intentionally designed to run indefinitely/until resources are exhausted
for i in $(seq 1 1000000000000); do
    arr=($(seq 1 1000))  # This array will take up memory with each iteration

    # Simulate heavy computation by simply doing nothing but waste CPU
    # (Note: this could be replaced with a non-useful computation)
    result=$((result + i))  # We are updating an uninitialized variable `result`

    # This loop will never end and will continue using CPU and memory.
done

# Print the result
echo "Computation finished. Result: $result"

```

```
[msherren@xsede.org@login-ci1 best_practices_scripts]$ sbatch broken_for_loop_example.sh
Submitted batch job 12535838
[msherren@xsede.org@login-ci1 best_practices_scripts]$ seff 12535838
Job ID: 12535838
Cluster: alpine
User/Group: msherren@xsede.org/msherrenpgrp@xsede.org
State: FAILED (exit code 2)
```

University of Colorado
Anschutz Medical Campus

Diagnosing a job that doesn't finish correctly

With that Job ID you can:

- seff <JobID>
- Check the logs: dump.<JobID>, slurm-<JobID>.out, error.<JobID>.out

With module load slurmtools:

- sacct –j <JobID>
- sacct –u <username>
- jobstats

See slides ***40-56*** For more details about these commands

- https://curc.readthedocs.io/en/latest/compute/monitoring-resources.html

```
#!/bin/bash


#SBATCH --job-name=job_scaling
#SBATCH --output=dump.%j.out
#SBATCH --account=amc-general
#SBATCH --nodes=1
#SBATCH --partition=amilan
#SBATCH --time=00:05:00
#SBATCH --ntasks=4
#    #SBATCH --mem=10G


## module load

echo "Job started on $(hostname) at $(date)"

echo "Running a simple computation..."
# The loop below is intentionally designed to run indefinitely/until resources are exhausted
for i in $(seq 1 1000000000000); do
    arr=($(seq 1 1000))  # This array will take up memory with each iteration

    # Simulate heavy computation by simply doing nothing but waste CPU
    # (Note: this could be replaced with a non-useful computation)
    result=$((result + i))  # We are updating an uninitialized variable `result`

    # This loop will never end and will continue using CPU and memory.
done

# Print the result
echo "Computation finished. Result: $result"

echo "Job completed at $(date)"
```

```
[msherren@xsede.org@login-ci1 best_practices_scripts]$ head dump.12535838.out
The job ID is: 12535838
Job started on c3cpu-c15-u17-1.rc.int.colorado.edu at Wed Mar 26 14:51:04 MDT 2025
Running a simple computation...
[msherren@xsede.org@login-ci1 best_practices_scripts]$ head error.12535838.out
/var/spool/slurmd/job12535838/slurm_script: xrealloc: cannot allocate 18446744071562067968 bytes
```

University of Colorado Anschutz Medical Campus

CPU resource requirements for each QOS/Partition

#SBATCH --qos	Partitions	#SBATCH --time	#SBATCH --mem	RAM per core (GB)	Max cores per node
normal	amilan	Max: 24 hrs	Max: 240GB	3.75	32/48/64
long	amilan	Min: 24 hrs Max: 7 Days	Max: 240GB	3.75	32/48/64
mem	amem only	Max: 7 Days	Min: 240GB Max: 2TB	16	48/64/128

GPU resource requirements for each QOS/Partition

#SBATCH --qos	Partitions	#SBATCH --time	#SBATCH --mem	RAM per core (GB)	Max cores per node
normal	aa100,ami100, al40	Max: 24 hrs	Max: 240GB	3.75	32/48/64
long	aa100,ami100, al40	Min: 24 hrs Max: 7 Days	Max: 240GB	3.75	32/48/64

*To specify how many GPUs (up to 3 per node), remember to include the --gres flag:
#SBATCH --gres=gpu for a single GPU, #SBATCH --gres=gpu:2 for 2, and #SBATCH --gres=gpu:3 for 3

**Make sure your job is set up for multiple GPUs before requesting multiple!

Which of these 4 scripts will fail?

```
#!/bin/bash


#SBATCH --job-name=qos_batch_examples
#SBATCH --output=dump.%j.out
#SBATCH --account=amc-general
#SBATCH --nodes=1
#SBATCH --qos=normal
#SBATCH --partition=amilan
#SBATCH --mem=1000MB
#SBATCH --time=01:00:00
#SBATCH --ntasks=4
```

```
#!/bin/bash


#SBATCH --job-name=qos_batch_examples
#SBATCH --output=dump.%j.out
#SBATCH --account=amc-general
#SBATCH --nodes=1
#SBATCH --qos=normal
#SBATCH --partition=amilan
#SBATCH --mem=10GB
#SBATCH --time=01:00:00
#SBATCH --ntasks=4
```

```
#!/bin/bash


#SBATCH --job-name=qos_batch_examples
#SBATCH --output=dump.%j.out
#SBATCH --account=amc-general
#SBATCH --nodes=1
#SBATCH --qos=mem
#SBATCH --partition=amem
#SBATCH --mem=100GB
#SBATCH --time=01:00:00
#SBATCH --ntasks=4
```

```
#!/bin/bash


#SBATCH --job-name=qos_batch_examples
#SBATCH --output=dump.%j.out
#SBATCH --account=amc-general
#SBATCH --nodes=1
#SBATCH --qos=mem
#SBATCH --partition=amem
#SBATCH --mem=500GB
#SBATCH --time=01:00:00
#SBATCH --ntasks=4
```

Which of these 4 scripts will fail?

```
#!/bin/bash


#SBATCH --job-name=qos_batch_examples
#SBATCH --output=dump.%j.out
#SBATCH --account=amc-general
#SBATCH --nodes=1
#SBATCH --qos=normal
#SBATCH --partition=amilan
#SBATCH --mem=1000MB
#SBATCH --time=01:00:00
#SBATCH --ntasks=4
```

```
#!/bin/bash


#SBATCH --job-name=qos_batch_examples
#SBATCH --output=dump.%j.out
#SBATCH --account=amc-general
#SBATCH --nodes=1
#SBATCH --qos=normal
#SBATCH --partition=amilan
#SBATCH --mem=10GB
#SBATCH --time=01:00:00
#SBATCH --ntasks=4
```

```
#!/bin/bash


#SBATCH --job-name=qos_batch_examples
#SBATCH --output=dump.%j.out
#SBATCH --account=amc-general
#SBATCH --nodes=1
#SBATCH --qos=mem
#SBATCH --partition=amem
#SBATCH --mem=100GB
#SBATCH --time=01:00:00
#SBATCH --ntasks=4
```

This doesn't work because the partition has a minimum memory requirement of 240GB

```
#!/bin/bash


#SBATCH --job-name=qos_batch_examples
#SBATCH --output=dump.%j.out
#SBATCH --account=amc-general
#SBATCH --nodes=1
#SBATCH --qos=mem
#SBATCH --partition=amem
#SBATCH --mem=500GB
#SBATCH --time=01:00:00
#SBATCH --ntasks=4
```

Which of these 4 scripts will fail?

```
#!/bin/bash


#SBATCH --job-name=job_scaling
#SBATCH --output=dump.%j.out
#SBATCH --account=amc-general
#SBATCH --nodes=1
#SBATCH --partition=aa100
#SBATCH --time=02:00:00
#SBATCH --ntasks=50
#SBATCH --gres=gpu:2
```

```
#!/bin/bash


#SBATCH --job-name=job_scaling
#SBATCH --output=dump.%j.out
#SBATCH --account=amc-general
#SBATCH --nodes=1
#SBATCH --qos=long
#SBATCH --partition=al40
#SBATCH --time=7-00:00:00
#SBATCH --ntasks=16
#SBATCH --gres=gpu:3
```

```
#!/bin/bash


#SBATCH --job-name=job_scaling
#SBATCH --output=dump.%j.out
#SBATCH --account=amc-general
#SBATCH --nodes=1
#SBATCH --time=4:00:00
#SBATCH --ntasks=8
#SBATCH --gres=gpu
#SBATCH --mem=20GB
```

```
#!/bin/bash


#SBATCH --job-name=job_scaling
#SBATCH --output=dump.%j.out
#SBATCH --account=amc-general
#SBATCH --qos=normal
#SBATCH --partition=ami100
#SBATCH --nodes=1
#SBATCH --ntasks=16
#SBATCH --time=48:00:00
#SBATCH --gres=gpu:3
```

Which of these 4 scripts will fail?

```
#!/bin/bash


#SBATCH --job-name=job_scaling
#SBATCH --output=dump.%j.out
#SBATCH --account=amc-general
#SBATCH --nodes=1
#SBATCH --partition=aa100
#SBATCH --time=02:00:00
#SBATCH --ntasks=50
#SBATCH --gres=gpu:2
```

```
#!/bin/bash


#SBATCH --job-name=job_scaling
#SBATCH --output=dump.%j.out
#SBATCH --account=amc-general
#SBATCH --nodes=1
#SBATCH --qos=long
#SBATCH --partition=al40
#SBATCH --time=7-00:00:00
#SBATCH --ntasks=16
#SBATCH --gres=gpu:3
```

```
#!/bin/bash


#SBATCH --job-name=job_scaling
#SBATCH --output=dump.%j.out
#SBATCH --account=amc-general
#SBATCH --nodes=1
#SBATCH --time=4:00:00
#SBATCH --ntasks=8
#SBATCH --gres=gpu
#SBATCH --mem=20GB
```

```
#!/bin/bash


#SBATCH --job-name=job_scaling
#SBATCH --output=dump.%j.out
#SBATCH --account=amc-general
#SBATCH --qos=normal
#SBATCH --partition=ami100
#SBATCH --nodes=1
#SBATCH --ntasks=16
#SBATCH --time=48:00:00
#SBATCH --gres=gpu:3
```

This doesn't work because the normal qos has a maximum time limit of 24 hours

University of Colorado
Anschutz Medical Campus

```
sbatch: error: Error: The normal QOS has a maximum wall time limit of 24 hours. Please adjust your reques
ted time or choose an appropriate QOS.
sbatch: error: Batch job submission failed: Unspecified error
```

Which of these 4 scripts will fail?

```
#!/bin/bash


#SBATCH --job-name=job_scaling
#SBATCH --output=dump.%j.out
#SBATCH --account=amc-general
#SBATCH --nodes=1
#SBATCH --partition=aa100
#SBATCH --time=02:00:00
#SBATCH --ntasks=50
#SBATCH --gres=gpu:2
```

```
#!/bin/bash


#SBATCH --job-name=job_scaling
#SBATCH --output=dump.%j.out
#SBATCH --account=amc-general
#SBATCH --nodes=1
#SBATCH --qos=long
#SBATCH --partition=al40
#SBATCH --time=7-00:00:00
#SBATCH --ntasks=16
#SBATCH --gres=gpu:3
```

```
#!/bin/bash


#SBATCH --job-name=job_scaling
#SBATCH --output=dump.%j.out
#SBATCH --account=amc-general
#SBATCH --nodes=1
#SBATCH --time=4:00:00
#SBATCH --ntasks=8
#SBATCH --gres=gpu
#SBATCH --mem=20GB
```

```
#!/bin/bash


#SBATCH --job-name=job_scaling
#SBATCH --output=dump.%j.out
#SBATCH --account=amc-general
#SBATCH --qos=long
#SBATCH --partition=ami100
#SBATCH --nodes=1
#SBATCH --ntasks=16
#SBATCH --time=20:00:00
#SBATCH --gres=gpu:3
```

This also does not work! Requesting long with less than 24 hours will queue for longer and waste resources

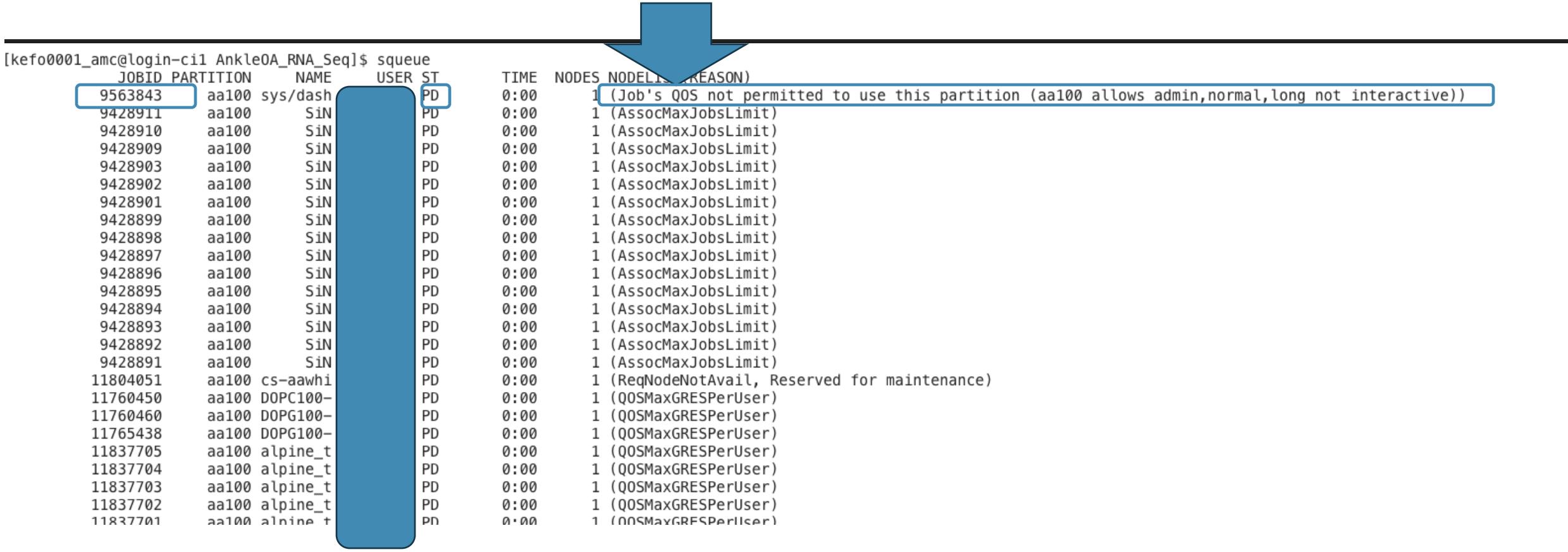
squeue

- Allows to see jobs located in the slurm scheduling queue.

```
[kefo0001_amc@login-ci1 AnkleOA_RNA_Seq]$ squeue
             JOBID PARTITION     NAME [redacted]ST       TIME  NODES NODELIST(REASON)
           9563843     aa100 sys/dash [redacted]PD       0:00      1 (Job's QOS not permitted to use this partition (aa100 allows admin,normal,long not interactive))
           9428911     aa100      SiN [redacted]PD       0:00      1 (AssocMaxJobsLimit)
           9428910     aa100      SiN [redacted]PD       0:00      1 (AssocMaxJobsLimit)
           9428909     aa100      SiN [redacted]PD       0:00      1 (AssocMaxJobsLimit)
           9428903     aa100      SiN [redacted]PD       0:00      1 (AssocMaxJobsLimit)
           9428902     aa100      SiN [redacted]PD       0:00      1 (AssocMaxJobsLimit)
           9428901     aa100      SiN [redacted]PD       0:00      1 (AssocMaxJobsLimit)
           9428899     aa100      SiN [redacted]PD       0:00      1 (AssocMaxJobsLimit)
           9428898     aa100      SiN [redacted]PD       0:00      1 (AssocMaxJobsLimit)
           9428897     aa100      SiN [redacted]PD       0:00      1 (AssocMaxJobsLimit)
           9428896     aa100      SiN [redacted]PD       0:00      1 (AssocMaxJobsLimit)
           9428895     aa100      SiN [redacted]PD       0:00      1 (AssocMaxJobsLimit)
           9428894     aa100      SiN [redacted]PD       0:00      1 (AssocMaxJobsLimit)
           9428893     aa100      SiN [redacted]PD       0:00      1 (AssocMaxJobsLimit)
           9428892     aa100      SiN [redacted]PD       0:00      1 (AssocMaxJobsLimit)
           9428891     aa100      SiN [redacted]PD       0:00      1 (AssocMaxJobsLimit)
          11804051     aa100 cs-aawhi [redacted]PD       0:00      1 (ReqNodeNotAvail, Reserved for maintenance)
          11760450     aa100 DOPC100- [redacted]PD       0:00      1 (QOSMaxGRESPerUser)
          11760460     aa100 DOPG100- [redacted]PD       0:00      1 (QOSMaxGRESPerUser)
          11765438     aa100 DOPG100- [redacted]PD       0:00      1 (QOSMaxGRESPerUser)
          11837705     aa100 alpine_t [redacted]PD       0:00      1 (QOSMaxGRESPerUser)
          11837704     aa100 alpine_t [redacted]PD       0:00      1 (QOSMaxGRESPerUser)
          11837703     aa100 alpine_t [redacted]PD       0:00      1 (QOSMaxGRESPerUser)
          11837702     aa100 alpine_t [redacted]PD       0:00      1 (QOSMaxGRESPerUser)
          11837701     aa100 alpine_t [redacted]PD       0:00      1 (QOSMaxGRESPerUser)
```

squeue

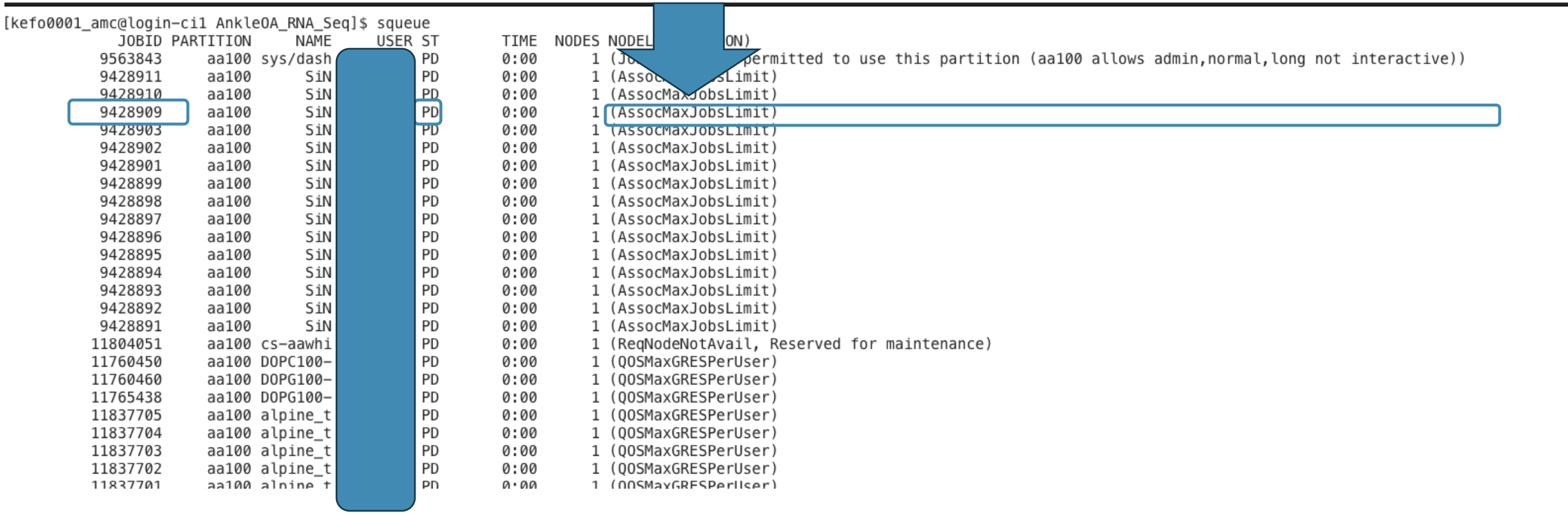
- Job 9563843 is pending because it is trying to use aa100 as an interactive partition for #SBATCH --qos



```
[kefo0001_amc@login-ci1 AnkleOA_RNA_Seq]$ squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
           9563843     aa100 sys/dash [redacted] PD       0:00      1 (Job's QOS not permitted to use this partition (aa100 allows admin,normal,long not interactive))
           9428911     aa100      SiN [redacted] PD       0:00      1 (AssocMaxJobsLimit)
           9428910     aa100      SiN [redacted] PD       0:00      1 (AssocMaxJobsLimit)
           9428909     aa100      SiN [redacted] PD       0:00      1 (AssocMaxJobsLimit)
           9428903     aa100      SiN [redacted] PD       0:00      1 (AssocMaxJobsLimit)
           9428902     aa100      SiN [redacted] PD       0:00      1 (AssocMaxJobsLimit)
           9428901     aa100      SiN [redacted] PD       0:00      1 (AssocMaxJobsLimit)
           9428899     aa100      SiN [redacted] PD       0:00      1 (AssocMaxJobsLimit)
           9428898     aa100      SiN [redacted] PD       0:00      1 (AssocMaxJobsLimit)
           9428897     aa100      SiN [redacted] PD       0:00      1 (AssocMaxJobsLimit)
           9428896     aa100      SiN [redacted] PD       0:00      1 (AssocMaxJobsLimit)
           9428895     aa100      SiN [redacted] PD       0:00      1 (AssocMaxJobsLimit)
           9428894     aa100      SiN [redacted] PD       0:00      1 (AssocMaxJobsLimit)
           9428893     aa100      SiN [redacted] PD       0:00      1 (AssocMaxJobsLimit)
           9428892     aa100      SiN [redacted] PD       0:00      1 (AssocMaxJobsLimit)
           9428891     aa100      SiN [redacted] PD       0:00      1 (AssocMaxJobsLimit)
          11804051     aa100 cs-aawhi [redacted] PD       0:00      1 (ReqNodeNotAvail, Reserved for maintenance)
          11760450     aa100 DOPC100- [redacted] PD       0:00      1 (QOSMaxGRESPerUser)
          11760460     aa100 DOPG100- [redacted] PD       0:00      1 (QOSMaxGRESPerUser)
          11765438     aa100 DOPG100- [redacted] PD       0:00      1 (QOSMaxGRESPerUser)
          11837705     aa100 alpine_t [redacted] PD       0:00      1 (QOSMaxGRESPerUser)
          11837704     aa100 alpine_t [redacted] PD       0:00      1 (QOSMaxGRESPerUser)
          11837703     aa100 alpine_t [redacted] PD       0:00      1 (QOSMaxGRESPerUser)
          11837702     aa100 alpine_t [redacted] PD       0:00      1 (QOSMaxGRESPerUser)
          11837701     aa100 alpine_t [redacted] PD       0:00      1 (QOSMaxGRESPerUser)
```

squeue

- Job 9428909 is requesting too many jobs!!



```
[kefo0001_amc@login-ci1 AnkleOA_RNA_Seq]$ squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODEL          ON)
           9563843     aa100 sys/dash [redacted] PD       0:00      1 (JO          permitted to use this partition (aa100 allows admin,normal,long not interactive))
           9428911     aa100      SiN [redacted] PD       0:00      1 (Assoc       sLimit)
           9428910     aa100      SiN [redacted] PD       0:00      1 (AssocMaxJobsLimit)
           9428909     aa100      SiN [redacted] PD       0:00      1 (AssocMaxJobsLimit)
           9428903     aa100      SiN [redacted] PD       0:00      1 (AssocMaxJobsLimit)
           9428902     aa100      SiN [redacted] PD       0:00      1 (AssocMaxJobsLimit)
           9428901     aa100      SiN [redacted] PD       0:00      1 (AssocMaxJobsLimit)
           9428899     aa100      SiN [redacted] PD       0:00      1 (AssocMaxJobsLimit)
           9428898     aa100      SiN [redacted] PD       0:00      1 (AssocMaxJobsLimit)
           9428897     aa100      SiN [redacted] PD       0:00      1 (AssocMaxJobsLimit)
           9428896     aa100      SiN [redacted] PD       0:00      1 (AssocMaxJobsLimit)
           9428895     aa100      SiN [redacted] PD       0:00      1 (AssocMaxJobsLimit)
           9428894     aa100      SiN [redacted] PD       0:00      1 (AssocMaxJobsLimit)
           9428893     aa100      SiN [redacted] PD       0:00      1 (AssocMaxJobsLimit)
           9428892     aa100      SiN [redacted] PD       0:00      1 (AssocMaxJobsLimit)
           9428891     aa100      SiN [redacted] PD       0:00      1 (AssocMaxJobsLimit)
          11804051     aa100 cs-aawhi [redacted] PD       0:00      1 (ReqNodeNotAvail, Reserved for maintenance)
          11760450     aa100 DOPC100- [redacted] PD       0:00      1 (QOSMaxGRESPerUser)
          11760460     aa100 DOPG100- [redacted] PD       0:00      1 (QOSMaxGRESPerUser)
          11765438     aa100 DOPG100- [redacted] PD       0:00      1 (QOSMaxGRESPerUser)
          11837705     aa100 alpine_t [redacted] PD       0:00      1 (QOSMaxGRESPerUser)
          11837704     aa100 alpine_t [redacted] PD       0:00      1 (QOSMaxGRESPerUser)
          11837703     aa100 alpine_t [redacted] PD       0:00      1 (QOSMaxGRESPerUser)
          11837702     aa100 alpine_t [redacted] PD       0:00      1 (QOSMaxGRESPerUser)
          11837701     aa100 alpine_t [redacted] PD       0:00      1 (QOSMaxGRESPerUser)
```

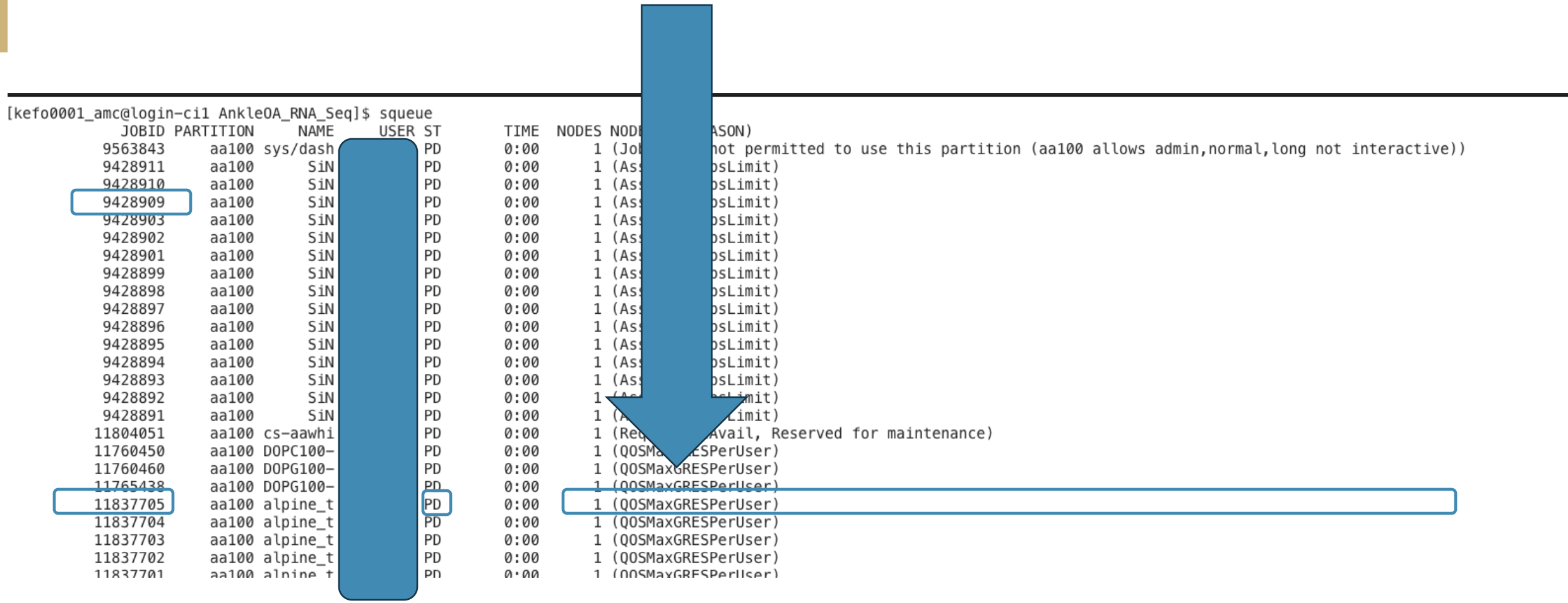
squeue

- Job 9428909 is requesting too many jobs!!
- Note that all jobs associated with that username will eventually run.

```
[kefo0001_amc@login-ci1 AnkleOA_RNA_Seq]$ squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODEL          ON)
           9563843     aa100 sys/dash [redacted] PD       0:00      1 (JO          permitted to use this partition (aa100 allows admin,normal,long not interactive))
           9428911     aa100      SiN [redacted] PD       0:00      1 (Assoc       sLimit)
           9428910     aa100      SiN [redacted] PD       0:00      1 (AssocMaxJobsLimit)
           9428909     aa100      SiN [redacted] PD       0:00      1 (AssocMaxJobsLimit)
           9428903     aa100      SiN [redacted] PD       0:00      1 (AssocMaxJobsLimit)
           9428902     aa100      SiN [redacted] PD       0:00      1 (AssocMaxJobsLimit)
           9428901     aa100      SiN [redacted] PD       0:00      1 (AssocMaxJobsLimit)
           9428899     aa100      SiN [redacted] PD       0:00      1 (AssocMaxJobsLimit)
           9428898     aa100      SiN [redacted] PD       0:00      1 (AssocMaxJobsLimit)
           9428897     aa100      SiN [redacted] PD       0:00      1 (AssocMaxJobsLimit)
           9428896     aa100      SiN [redacted] PD       0:00      1 (AssocMaxJobsLimit)
           9428895     aa100      SiN [redacted] PD       0:00      1 (AssocMaxJobsLimit)
           9428894     aa100      SiN [redacted] PD       0:00      1 (AssocMaxJobsLimit)
           9428893     aa100      SiN [redacted] PD       0:00      1 (AssocMaxJobsLimit)
           9428892     aa100      SiN [redacted] PD       0:00      1 (AssocMaxJobsLimit)
           9428891     aa100      SiN [redacted] PD       0:00      1 (AssocMaxJobsLimit)
          11804051     aa100 cs-aawhi [redacted] PD       0:00      1 (ReqNodeNotAvail, Reserved for maintenance)
          11760450     aa100 DOPC100- [redacted] PD       0:00      1 (QOSMaxGRESPerUser)
          11760460     aa100 DOPG100- [redacted] PD       0:00      1 (QOSMaxGRESPerUser)
          11765438     aa100 DOPG100- [redacted] PD       0:00      1 (QOSMaxGRESPerUser)
          11837705     aa100 alpine_t [redacted] PD       0:00      1 (QOSMaxGRESPerUser)
          11837704     aa100 alpine_t [redacted] PD       0:00      1 (QOSMaxGRESPerUser)
          11837703     aa100 alpine_t [redacted] PD       0:00      1 (QOSMaxGRESPerUser)
          11837702     aa100 alpine_t [redacted] PD       0:00      1 (QOSMaxGRESPerUser)
          11837701     aa100 alpine_t [redacted] PD       0:00      1 (QOSMaxGRESPerUser)
```

squeue

- Job 11837705 probably requested too many gpus at once!!!
- In this case, the user needs to decrease their amount of gpus requested.



```
[kefo0001_amc@login-ci1 AnkleOA_RNA_Seq]$ squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODE      ASON)
           9563843     aa100 sys/dash          PD       0:00      1 (Jo      ot permitted to use this partition (aa100 allows admin,normal,long not interactive))
           9428911     aa100      SiN          PD       0:00      1 (As      bsLimit)
           9428910     aa100      SiN          PD       0:00      1 (As      bsLimit)
           9428909     aa100      SiN          PD       0:00      1 (As      bsLimit)
           9428903     aa100      SiN          PD       0:00      1 (As      bsLimit)
           9428902     aa100      SiN          PD       0:00      1 (As      bsLimit)
           9428901     aa100      SiN          PD       0:00      1 (As      bsLimit)
           9428899     aa100      SiN          PD       0:00      1 (As      bsLimit)
           9428898     aa100      SiN          PD       0:00      1 (As      bsLimit)
           9428897     aa100      SiN          PD       0:00      1 (As      bsLimit)
           9428896     aa100      SiN          PD       0:00      1 (As      bsLimit)
           9428895     aa100      SiN          PD       0:00      1 (As      bsLimit)
           9428894     aa100      SiN          PD       0:00      1 (As      bsLimit)
           9428893     aa100      SiN          PD       0:00      1 (As      bsLimit)
           9428892     aa100      SiN          PD       0:00      1 (As      bsLimit)
           9428891     aa100      SiN          PD       0:00      1 (A       Limit)
          11804051     aa100 cs-aawhi          PD       0:00      1 (Req     Avail, Reserved for maintenance)
          11760450     aa100 DOPC100-          PD       0:00      1 (QOSMa   ESPerUser)
          11760460     aa100 DOPG100-          PD       0:00      1 (QOSMaxGRESPerUser)
          11765438     aa100 DOPG100-          PD       0:00      1 (QOSMaxGRESPerUser)
          11837705     aa100 alpine_t          PD       0:00      1 (QOSMaxGRESPerUser)
          11837704     aa100 alpine_t          PD       0:00      1 (QOSMaxGRESPerUser)
          11837703     aa100 alpine_t          PD       0:00      1 (QOSMaxGRESPerUser)
          11837702     aa100 alpine_t          PD       0:00      1 (QOSMaxGRESPerUser)
          11837701     aa100 alpine_t          PD       0:00      1 (QOSMaxGRESPerUser)
```

University of Colorado
Anschutz Medical Campus

squeue

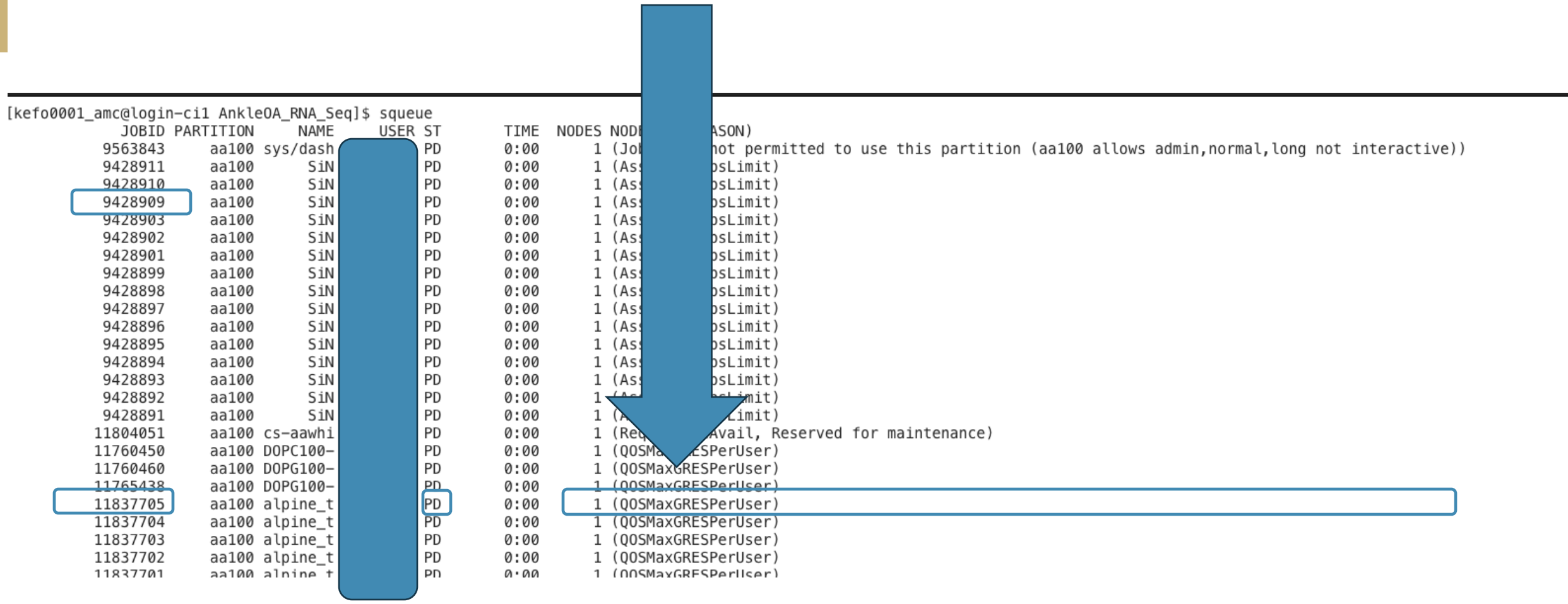
- Users are limited to a maximum of 16,15 and 6 total GPUs on aa100, ami100 and al40 partitions respectively

```
[kefo0001_amc@login-ci1 AnkleOA_RNA_Seq]$ squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODE        ASON)
           9563843     aa100 sys/dash          PD       0:00      1 (Jo        ot permitted to use this partition (aa100 allows admin,normal,long not interactive))
           9428911     aa100      SiN          PD       0:00      1 (As        bsLimit)
           9428910     aa100      SiN          PD       0:00      1 (As        bsLimit)
           9428909     aa100      SiN          PD       0:00      1 (As        bsLimit)
           9428903     aa100      SiN          PD       0:00      1 (As        bsLimit)
           9428902     aa100      SiN          PD       0:00      1 (As        bsLimit)
           9428901     aa100      SiN          PD       0:00      1 (As        bsLimit)
           9428899     aa100      SiN          PD       0:00      1 (As        bsLimit)
           9428898     aa100      SiN          PD       0:00      1 (As        bsLimit)
           9428897     aa100      SiN          PD       0:00      1 (As        bsLimit)
           9428896     aa100      SiN          PD       0:00      1 (As        bsLimit)
           9428895     aa100      SiN          PD       0:00      1 (As        bsLimit)
           9428894     aa100      SiN          PD       0:00      1 (As        bsLimit)
           9428893     aa100      SiN          PD       0:00      1 (As        bsLimit)
           9428892     aa100      SiN          PD       0:00      1 (As        asLimit)
           9428891     aa100      SiN          PD       0:00      1 (A         Limit)
          11804051     aa100 cs-aawhi          PD       0:00      1 (Req       Avail, Reserved for maintenance)
          11760450     aa100 DOPC100-          PD       0:00      1 (QOSMa     ESPerUser)
          11760460     aa100 DOPG100-          PD       0:00      1 (QOSMaxGRESPerUser)
          11765438     aa100 DOPG100-          PD       0:00      1 (QOSMaxGRESPerUser)
          11837705     aa100 alpine_t          PD       0:00      1 (QOSMaxGRESPerUser)
          11837704     aa100 alpine_t          PD       0:00      1 (QOSMaxGRESPerUser)
          11837703     aa100 alpine_t          PD       0:00      1 (QOSMaxGRESPerUser)
          11837702     aa100 alpine_t          PD       0:00      1 (QOSMaxGRESPerUser)
          11837701     aa100 alpine_t          PD       0:00      1 (QOSMaxGRESPerUser)
```

University of Colorado
Anschutz Medical Campus

squeue

- A maximum of 22 total gpus per users.



```
[kefo0001_amc@login-ci1 AnkleOA_RNA_Seq]$ squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODE        ASON)
           9563843     aa100 sys/dash          PD       0:00      1 (Jo        ot permitted to use this partition (aa100 allows admin,normal,long not interactive))
           9428911     aa100      SiN          PD       0:00      1 (As        bsLimit)
           9428910     aa100      SiN          PD       0:00      1 (As        bsLimit)
           9428909     aa100      SiN          PD       0:00      1 (As        bsLimit)
           9428903     aa100      SiN          PD       0:00      1 (As        bsLimit)
           9428902     aa100      SiN          PD       0:00      1 (As        bsLimit)
           9428901     aa100      SiN          PD       0:00      1 (As        bsLimit)
           9428899     aa100      SiN          PD       0:00      1 (As        bsLimit)
           9428898     aa100      SiN          PD       0:00      1 (As        bsLimit)
           9428897     aa100      SiN          PD       0:00      1 (As        bsLimit)
           9428896     aa100      SiN          PD       0:00      1 (As        bsLimit)
           9428895     aa100      SiN          PD       0:00      1 (As        bsLimit)
           9428894     aa100      SiN          PD       0:00      1 (As        bsLimit)
           9428893     aa100      SiN          PD       0:00      1 (As        bsLimit)
           9428892     aa100      SiN          PD       0:00      1 (A         mit)
           9428891     aa100      SiN          PD       0:00      1 (A         Limit)
          11804051     aa100 cs-aawhi          PD       0:00      1 (Re        Avail, Reserved for maintenance)
          11760450     aa100 DOPC100-          PD       0:00      1 (QOSMa     ESPerUser)
          11760460     aa100 DOPG100-          PD       0:00      1 (QOSMaxGRESPerUser)
          11765438     aa100 DOPG100-          PD       0:00      1 (QOSMaxGRESPerUser)
          11837705     aa100 alpine_t          PD       0:00      1 (QOSMaxGRESPerUser)
          11837704     aa100 alpine_t          PD       0:00      1 (QOSMaxGRESPerUser)
          11837703     aa100 alpine_t          PD       0:00      1 (QOSMaxGRESPerUser)
          11837702     aa100 alpine_t          PD       0:00      1 (QOSMaxGRESPerUser)
          11837701     aa100 alpine_t          PD       0:00      1 (QOSMaxGRESPerUser)
```

University of Colorado
Anschutz Medical Campus

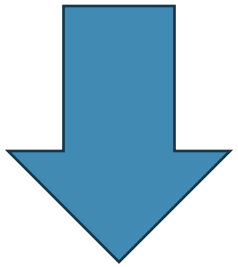
squeue

- --account helps to list only jobs related Anschutz Medical Campus

```
[kefo0001_amc@login-ci1 AnkleOA_RNA_Seq]$ squeue --account=amc-general
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
          11838139  acompile acompile [redacted]  R      19:44      1 c3cpu-a2-u32-1
          11838067  acompile acompile [redacted]  R      25:38      1 c3cpu-a2-u32-1
          11838097  acompile acompile [redacted]  R      23:45      1 c3cpu-a2-u32-1
          11838498  acompile acompile [redacted]  R       2:30      1 c3cpu-a2-u32-1
          11717781      amem  dissect [redacted] PD       0:00      1 (Job's QOS not permitted to use this partition (amem allows admin,mem not normal))
          11836668      amem ICM_2nd_ [redacted]  R    1:02:40      1 c3mem-a9-u22-1
          11797601    amilan nf-NFCOR [redacted] PD       0:00      1 (QOSGrpNodeLimit)
          11836259    amilan sys/dash [redacted] PD       0:00      1 (Job's QOS not permitted to use this partition (amilan allows admin,normal,long not interactive))
          11836309    amilan    kp_rf [redacted] PD       0:00      1 (QOSGrpNodeLimit)
          11838140    amilan NKcellra [redacted] PD       0:00      1 (QOSGrpNodeLimit)
          11838141    amilan NKcellra [redacted] PD       0:00      1 (QOSGrpNodeLimit)
          11757411    amilan   sbatch [redacted]  R 4-19:54:23      1 c3cpu-c15-u3-4
          11788871    amilan    sarek [redacted]  R 2-00:53:37      1 c3cpu-c11-u15-1
          11838133    amilan cellrang [redacted]  R      19:52      1 c3cpu-c15-u34-1
          11837053    amilan 50b981f1 [redacted]  R      55:15      1 c3cpu-c13-u3-3
          11838490    amilan ID.nPOD6 [redacted]  R       3:00      1 c3cpu-c9-u1-3
          11838473    amilan ID.nPOD6 [redacted]  R       4:08      1 c3cpu-a5-u3-1
          11837051    amilan snakemak [redacted]  R      55:28      1 c3cpu-c15-u3-3
       11837862_18    amilan run_rfmi [redacted]  R      14:30      1 c3cpu-a7-u1-1
       11837862_17    amilan run_rfmi [redacted]  R      14:31      1 c3cpu-a2-u1-1
       11837862_16    amilan run_rfmi [redacted]  R      14:33      1 c3cpu-a7-u7-4
       11837862_14    amilan run_rfmi [redacted]  R      14:34      1 c3cpu-a5-u1-2
       11837862_15    amilan run_rfmi [redacted]  R      14:34      1 c3cpu-c15-u7-1
       11837862_13    amilan run_rfmi [redacted]  R      16:48      1 c3cpu-a7-u28-4
       11837862_12    amilan run_rfmi [redacted]  R      30:14      1 c3cpu-a7-u28-1
       11837862_11    amilan run_rfmi [redacted]  R      30:33      1 c3cpu-a5-u7-1
          11836253    amilan sinterac [redacted]  R    1:47:30      2 c3cpu-c15-u1-2,c3cpu-c15-u3-1
          11836307    amilan sys/dash [redacted]  R    1:25:53      1 c3cpu-c15-u3-1
          11836249    amilan   picard [redacted]  R    1:49:23      1 c3cpu-c15-u1-3
          11836680 atesting_ sinterac [redacted]  R      59:29      1 c3gpu-c2-u13
```

squeue

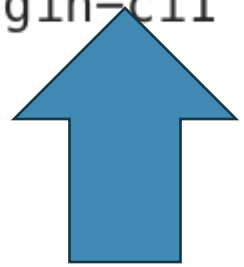
- “**sbatch <slurm_jobscript>** to submit jobs



```
[kefo0001_amc@login-ci1 AnkleOA_RNA_Seq]$ sbatch scripts/dummy_script.sh
Submitted batch job 11838571
[kefo0001_amc@login-ci1 AnkleOA_RNA_Seq]$
```

squeue

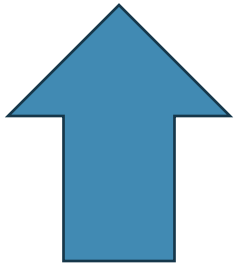
```
[kefo0001_amc@login-ci1 AnkleOA_RNA_Seq]$ sbatch scripts/dummy_script.sh
Submitted batch job 11838571
[kefo0001_amc@login-ci1 AnkleOA_RNA_Seq]$
```



- A submitted job can be monitored with its jobID.

squeue

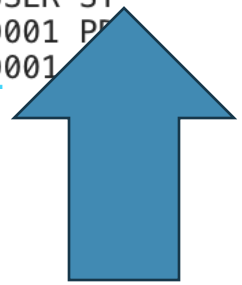
```
[kefo0001_amc@login-ci1 AnkleOA_RNA_Seq]$ sbatch scripts/dummy_script.sh
Submitted batch job 11838630
[kefo0001_amc@login-ci1 AnkleOA_RNA_Seq]$ squeue -u $USER --start
             JOBID PARTITION     NAME     USER ST          START_TIME  NODES SCHEDNODES           NODELIST(REASON)
          11838630    amilan rsem_pre kefo0001 PD                 N/A      1 (null)                (Priority)
```



- “--start” option may sometimes provide an estimate as to when the job will actually begin. In this case, it is not working. The job is still pending and we do not have an estimated start time

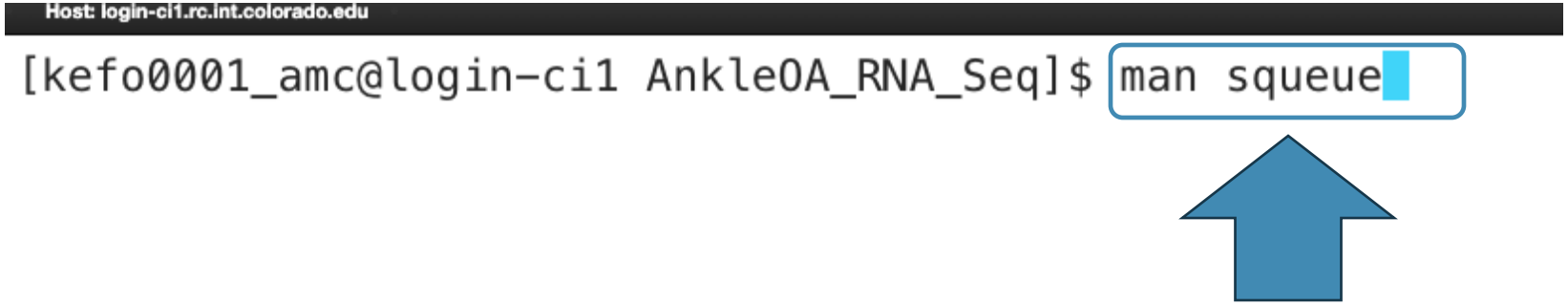
squeue

```
[kefo0001_amc@login-ci1 AnkleOA_RNA_Seq]$ squeue --me
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
          11838630    amilan rsem_pre kefo0001 PD       0:00      1 (Priority)
          11838615    amilan rsem_pre kefo0001  R       2:59      1 c3cpu-a7-u19-3
```



- “squeue –me” gives you information about your jobs only.

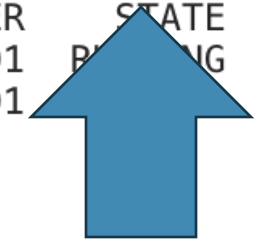
squeue



- To get more information about the command.

squeue

```
[kefo0001_amc@login-ci1 AnkleOA_RNA_Seq]$ squeue --me --long
Thu Feb 27 12:00:54 2025
             JOBID PARTITION     NAME     USER    STATE       TIME TIME_LIMI  NODES NODELIST(REASON)
          11838630    amilan rsem_pre kefo0001  RUNNING       1:24   3:00:00      1 c3cpu-a2-u21-1
          11838615    amilan rsem_pre kefo0001  RUNNING       5:30   3:00:00      1 c3cpu-a7-u19-3
```



- --long give more detailed information about a running job.

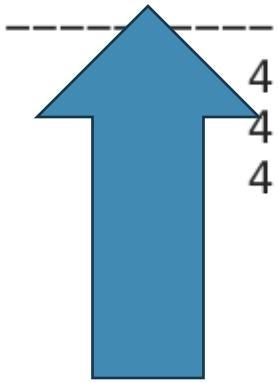
jobstats

```
[kefo0001_amc@login-ci1 AnkleOA_RNA_Seq]$ module load slurmtools
[kefo0001_amc@login-ci1 AnkleOA_RNA_Seq]$ jobstats $USER 1
job stats for user kefo0001_amc over past 1 days
jobid       jobname  partition    qos          accoun       cpus state    start-date-time     elapsed    wait
-------------------------------------------------------------------------------------------------------------
11807474    acompile acompile     compile      amc          1    TIMEOUT  2025-02-26T11:02:12 01:00:28   0 hrs
11808633    acompile acompile     compile      amc-         4    COMPLETE 2025-02-26T12:05:40 00:22:17   0 hrs
11808861    acompile acompile     compile      amc-         4    COMPLETE 2025-02-26T12:27:58 00:03:19   0 hrs
11808893    acompile acompile     compile      amc-         4    FAILED   2025-02-26T12:31:18 00:42:23   0 hrs
11810653    acompile acompile     compile      amc-         4    COMPLETE 2025-02-26T13:13:44 02:37:01   0 hrs
11820224    acompile acompile     compile      amc-         4    FAILED   2025-02-26T15:50:47 01:25:33   0 hrs
11823581    acompile acompile     compile      amc-         4    COMPLETE 2025-02-26T17:16:52 00:42:26   0 hrs
11824523    acompile acompile     compile      amc-         4    FAILED   2025-02-26T18:00:40 00:24:44   0 hrs
11824714    acompile acompile     compile      amc-         4    FAILED   2025-02-26T18:25:28 04:15:57   0 hrs
11838571    rsem_pre amilan       normal       amc-         30   CANCELLE 2025-02-27T11:52:06 00:01:11   0 hrs
11838615    rsem_pre amilan       normal       amc-         30   CANCELLE 2025-02-27T11:55:24 00:06:09   0 hrs
11838630    rsem_pre amilan       normal       amc-         30   COMPLETE 2025-02-27T11:59:30 00:44:34   0 hrs
11838957    sinterac aa100        normal       amc-         5    CANCELLE 2025-02-27T12:16:13 00:00:02   0 hrs
```

- With jobstats, you can print informations about previous jobs such as start date, jobIDs etc …

sacct

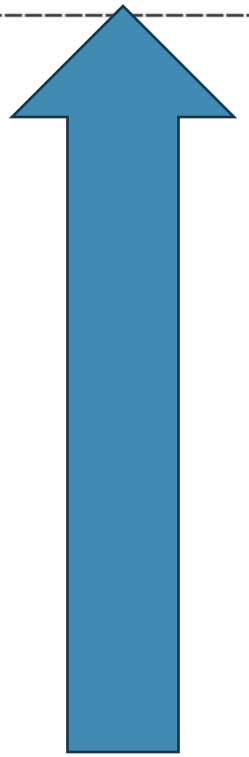
```
[kefo0001_amc@login-ci1 AnkleOA_RNA_Seq]$ sacct -j 11810653
JobID           JobName  Partition    Account  AllocCPUS      State ExitCode
------------ ---------- ---------- ---------- ---------- ---------- --------
11810653       acompile   acompile amc-gener+          4  COMPLETED      0:0
11810653.ex+     extern             amc-gener+          4  COMPLETED      0:0
11810653.0         bash             amc-gener+          4  COMPLETED      0:0
```



- With sacct you can give informations about past jobs

sacct

```
[kefo0001_amc@login-ci1 AnkleOA_RNA_Seq]$ sacct -B -j 11838630
Batch Script for 11838630
--------------------------------------------------------------------------------
#!/bin/bash

#SBATCH --nodes=1
#SBATCH --partition=amilan
#SBATCH --time=03:00:00
#SBATCH --ntasks=30
##SBATCH --mem=100G
#SBATCH --job-name="rsem_prepare_and_run"
#SBATCH -o "rsem_star.%j.out"
#SBATCH -e "rsem_star.%j.err"
#SBATCH --account=amc-general
#SBATCH --qos=normal

#module load apptainer
```



- With sacct you can print the batch script that was used to submit the job

Parallel jobs

```
#!/bin/bash

#SBATCH --partition=amilan            # Name of the partition. Remember that amilan is the CPU partition.
#SBATCH --job-name=first_mpirun           # Name of the job. You can check it with "squeue --me -l". Retrospectively you may
#SBATCH --output=first_mpirun-job.%j.out   # Name of the outout file. Note that %j stands for the jobID number
#SBATCH --error=first_mpirun-job.%j.err    # Name of the error file. Note that %j stands for the jobID number as well
#SBATCH --account=amc-general  # Account for the campus
#SBATCH --qos=normal   # Quality of service. (e.g. Normal qos allows for max walltime of 24 hours)
#SBATCH --nodes=1    # Total number of nodes requested. Always use 1 node unless you are runnign MPI!!!
#SBATCH --ntasks=4   # Total number of cores requested
#SBATCH --mail-type=ALL  # Begin, End and abort
#SBATCH --mail-user=kevin.fotso@cuanschutz.edu  # Email of the user
#SBATCH --time=00:01                 # Walltime that you are requesting to Slurm


# Load the c      that Openmpi depends on.
# The compil      ws to transfer source code to machine code
module load

#Load OpenMP     hieve tasks parallelization
module load

# Run the mp      mmand
# $SLURM_NTA      the variable associated with the number of tasks requested.
mpirun -n $S      ASKS hostname
```

- Only use 1 node unless you are using **MPI, gnu parallel, or sparkcluster**!!!

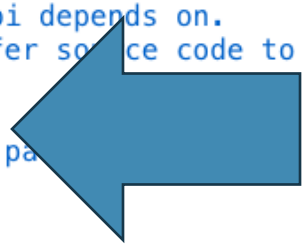
Parallel jobs

```
#!/bin/bash

#SBATCH --partition=amilan            # Name of the partition. Remember that amilan is the CPU partition.
#SBATCH --job-name=first_mpirun           # Name of the job. You can check it with "squeue --me -l". Retrospectively you may
#SBATCH --output=first_mpirun-job.%j.out   # Name of the outout file. Note that %j stands for the jobID number
#SBATCH --error=first_mpirun-job.%j.err    # Name of the error file. Note that %j stands for the jobID number as well
#SBATCH --account=amc-general  # Account for the campus
#SBATCH --qos=normal   # Quality of service. (e.g. Normal qos allows for max walltime of 24 hours)
#SBATCH --nodes=1    # Total number of nodes requested. Always use 1 node unless you are runnign MPI!!!
#SBATCH --ntasks=4   # Total number of cores requested
#SBATCH --mail-type=ALL  # Begin, End and abort
#SBATCH --mail-user=kevin.fotso@cuanschutz.edu  # Email of the user
#SBATCH --time=00:01                 # Walltime that you are requesting to Slurm


# Load the c      that Openmpi depends on.
# The compil      ws to transfer source code to machine code
module load

#Load OpenMP     hieve tasks parallelization
module load

# Run the mp      mmand
# $SLURM_NTA      the variable associated with the number of tasks requested.
mpirun -n $S      ASKS hostname
```

- **Warning**: If you call for 2 nodes but your software only uses 1, you will still waste core hours associated with using 2!!!

Parallel jobs

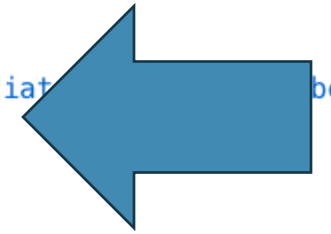
```
#!/bin/bash

#SBATCH --partition=amilan            # Name of the partition. Remember that amilan is the CPU partition.
#SBATCH --job-name=first_mpirun           # Name of the job. You can check it with "squeue --me -l". Retrospectively you may
#SBATCH --output=first_mpirun-job.%j.out   # Name of the outout file. Note that %j stands for the jobID number
#SBATCH --error=first_mpirun-job.%j.err    # Name of the error file. Note that %j stands for the jobID number as well
#SBATCH --account=amc-general  # Account for the campus
#SBATCH --qos=normal   # Quality of service. (e.g. Normal qos allows for max walltime of 24 hours)
#SBATCH --nodes=1    # Total number of nodes requested. Always use 1 node unless you are runnign MPI!!!
#SBATCH --ntasks=4   # Total number of cores requested
#SBATCH --mail-type=ALL  # Begin, End and abort
#SBATCH --mail-user=kevin.fotso@cuanschutz.edu  # Email of the user
#SBATCH --time=00:00:01                # Walltime that you are requesting to Slurm


# Load the compiler that Openmpi depends on.
# The compiler allows to transfer source code to machine code
module load gcc

#Load OpenMPI to achieve tasks pa
module load openmpi

# Run the mpirun command
# $SLURM_NTASKS is the variable associated with the number of tasks requested.
mpirun -n $SLURM_NTASKS hostname
```



- gcc is a dependency for openmpi. If you tried to load openmpi alone, it would not have worked. “**module spider openmpi**”

Parallel jobs

```
#!/bin/bash

#SBATCH --partition=amilan            # Name of the partition. Remember that amilan is the CPU partition.
#SBATCH --job-name=first_mpirun           # Name of the job. You can check it with "squeue --me -l". Retrospectively you may
#SBATCH --output=first_mpirun-job.%j.out   # Name of the outout file. Note that %j stands for the jobID number
#SBATCH --error=first_mpirun-job.%j.err    # Name of the error file. Note that %j stands for the jobID number as well
#SBATCH --account=amc-general  # Account for the campus
#SBATCH --qos=normal   # Quality of service. (e.g. Normal qos allows for max walltime of 24 hours)
#SBATCH --nodes=1    # Total number of nodes requested. Always use 1 node unless you are runnign MPI!!!
#SBATCH --ntasks=4   # Total number of cores requested
#SBATCH --mail-type=ALL  # Begin, End and abort
#SBATCH --mail-user=kevin.fotso@cuanschutz.edu  # Email of the user
#SBATCH --time=00:00:01                 # Walltime that you are requesting to Slurm


# Load the compiler that Openmpi depends on.
# The compiler allows to transfer source code to machine code
module load gcc

#Load OpenMPI to achieve tasks parallelization
module load openmpi

# Run the mpirun command
# $SLURM_NTASKS is the variable associat                ber of tasks requested.
mpirun -n $SLURM_NTASKS hostname
```



- Running MPI on 1 node with 4 cores.

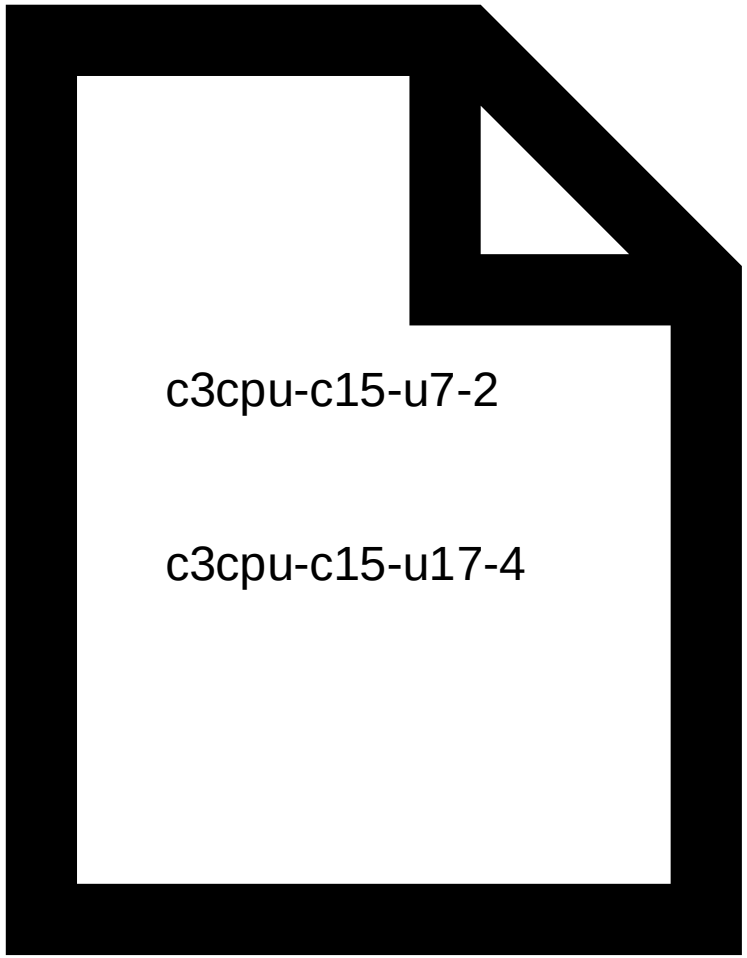
Parallel jobs

```
#!/bin/bash

#SBATCH --partition=amilan            # Name of the partition. Remember that amilan
#SBATCH --job-name=first_mpirun           # Name of the job. You can check it with "
#SBATCH --output=first_mpirun-job.%j.out   # Name of the outout file. Note that %j s
#SBATCH --error=first_mpirun-job.%j.err    # Name of the error file. Note that %j st
#SBATCH --account=amc-general  # Account for the campus
#SBATCH --qos=normal   # Quality of service. (e.g. Normal qos allows for max walltim
#SBATCH --nodes=2    # Total numb
#SBATCH --ntasks-per-node=4   #
#SBATCH --mail-type=ALL  # Begin,
#SBATCH --mail-user=kevin.fotso@cuanschutz.edu  # Email of the user
#SBATCH --time=00:00:01                 # Walltime that you are requesting to Slurm

# Load the compiler that Openmpi depends on.
# The compiler allows to transfer source code to machine code
module load gcc

#Load OpenMPI to achieve tasks parallelization
module load openmpi

# Get information about the node I am running on:
scontrol show hostname > $SLURM_SUBMIT_DIR/nodelist.txt

# Run the mpirun command
# $SLURM_NTASKS is the variable associated with the number of tasks requested.
mpirun --machinefile $SLURM_SUBMIT_DIR/nodelist.txt  -n $SLURM_NTASKS hostname
```

- Now we use run 4 cores each on 2 nodes.

Parallel jobs

```
#!/bin/bash

#SBATCH --partition=amilan            # Name of the partition. Remember that amilan
#SBATCH --job-name=first_mpirun          # Name of the job. You can check it with "
#SBATCH --output=first_mpirun-job.%j.out   # Name of the outout file. Note that %j s
#SBATCH --error=first_mpirun-job.%j.err    # Name of the error file. Note that %j st
#SBATCH --account=amc-general  # Account for the campus
#SBATCH --qos=normal   # Quality of service. (e.g. Normal qos allows for max walltim
#SBATCH --nodes=2    # Total number of nodes requested. Always use 1 node unless you
#SBATCH --ntasks-per-node=4   # Total number of cores requested
#SBATCH --mail-type=ALL  # Begin, End and abort
#SBATCH --mail-user=kevin.fotso@cuanschutz.edu  # Email of the user
#SBATCH --time=00:00:01                # Walltime that you are requesting to Slurm


# Load the compiler that Openmpi depends on.
# The compiler allows to transfer source code to machine code
module load gcc

#Load OpenMPI to achieve tasks parallelization
module load openmpi

# Get information about the node I am running on:
scontrol show hostname > $SLURM_SUBMIT_DIR/nodelist.txt

# Run the mpirun command
# $SLURM_NTASKS is the variable associated with the number of tasks requested.
mpirun --machinefile $SLURM_SUBMIT_DIR/nodelist.txt  -n $SLURM_NTASKS hostname
```



- Saving the address of the 2 nodes

Parallel jobs

```
#!/bin/bash

#SBATCH --partition=amilan            # Name of the partition. Remember that amilan
#SBATCH --job-name=first_mpirun           # Name of the job. You can check it with "
#SBATCH --output=first_mpirun-job.%j.out   # Name of the outout file. Note that %j s
#SBATCH --error=first_mpirun-job.%j.err    # Name of the error file. Note that %j st
#SBATCH --account=amc-general  # Account for the campus
#SBATCH --qos=normal   # Quality of service. (e.g. Normal qos allows for max walltim
#SBATCH --nodes=2    # Total number of nodes requested. Always use 1 node unless you
#SBATCH --ntasks-per-node=4   # Total number of cores requested
#SBATCH --mail-type=ALL  # Begin, End and abort
#SBATCH --mail-user=kevin.fotso@cuanschutz.edu  # Email of the user
#SBATCH --time=00:00:01                # Walltime that you are requesting to Slurm


# Load the compiler that Openmpi depends on.
# The compiler allows to transfer source code to machine code
module load gcc

#Load OpenMPI to achieve tasks parallelization
module load openmpi

# Get information about the node I am running on:
scontrol show hostname > $SLURM_SUBMIT_DIR/nodelist.txt

# Run the mpirun command
# $SLURM_NTASKS is the variable associated with the number of tasks requested.
mpirun --machinefile $SLURM_SUBMIT_DIR/nodelist.txt  -n $SLURM_NTASKS hostname
```

- Running on 2 nodes

Parallel jobs

```
c3cpu-c15-u7-2.rc.int.colorado.edu
c3cpu-c15-u7-2.rc.int.colorado.edu
c3cpu-c15-u7-2.rc.int.colorado.edu
c3cpu-c15-u7-2.rc.int.colorado.edu
c3cpu-c15-u17-4.rc.int.colorado.edu
c3cpu-c15-u17-4.rc.int.colorado.edu
c3cpu-c15-u17-4.rc.int.colorado.edu
c3cpu-c15-u17-4.rc.int.colorado.edu
```

- Proof that MPI ran on 2 nodes.

Parallel jobs

```
c3cpu-c15-u7-2.rc.int.colorado.edu
c3cpu-c15-u7-2.rc.int.colorado.edu
c3cpu-c15-u7-2.rc.int.colorado.edu
c3cpu-c15-u7-2.rc.int.colorado.edu
c3cpu-c15-u17-4.rc.int.colorado.edu
c3cpu-c15-u17-4.rc.int.colorado.edu
c3cpu-c15-u17-4.rc.int.colorado.edu
c3cpu-c15-u17-4.rc.int.colorado.edu
```

- Proof that MPI ran on 2 nodes.
- We have 4 tasks per node.
- For gnu parallel related applications please refer to this guide on our repository:

https://github.com/kf-cuanschutz/CU-Anschutz-HPC-documentation/blob/main/Office-hours-presentation-files/GNU_parallel_presentation.pdf

Parallel jobs

```
c3cpu-c15-u7-2.rc.int.colorado.edu
c3cpu-c15-u7-2.rc.int.colorado.edu
c3cpu-c15-u7-2.rc.int.colorado.edu
c3cpu-c15-u7-2.rc.int.colorado.edu
c3cpu-c15-u17-4.rc.int.colorado.edu
c3cpu-c15-u17-4.rc.int.colorado.edu
c3cpu-c15-u17-4.rc.int.colorado.edu
c3cpu-c15-u17-4.rc.int.colorado.edu
```

- Proof that MPI ran on 2 nodes.
- We have 4 tasks per node.
- For gnu parallel related applications please refer to this guide on our repository:

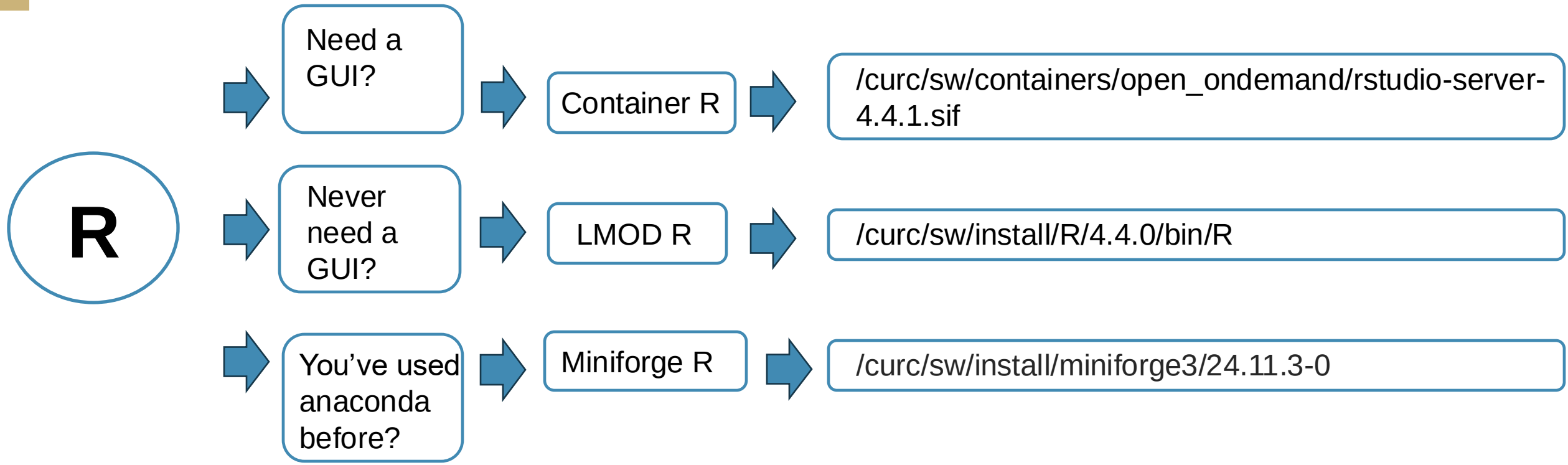
https://github.com/kf-cuanschutz/CU-Anschutz-HPC-documentation/blob/main/Office-hours-presentation-files/GNU_parallel_presentation.pdf

- For information on job arrays please refer to this guide:

https://github.com/kf-cuanschutz/CU-Anschutz-HPC-documentation/blob/main/Job_array_demonstration.md

What are the 3 ways to use R?

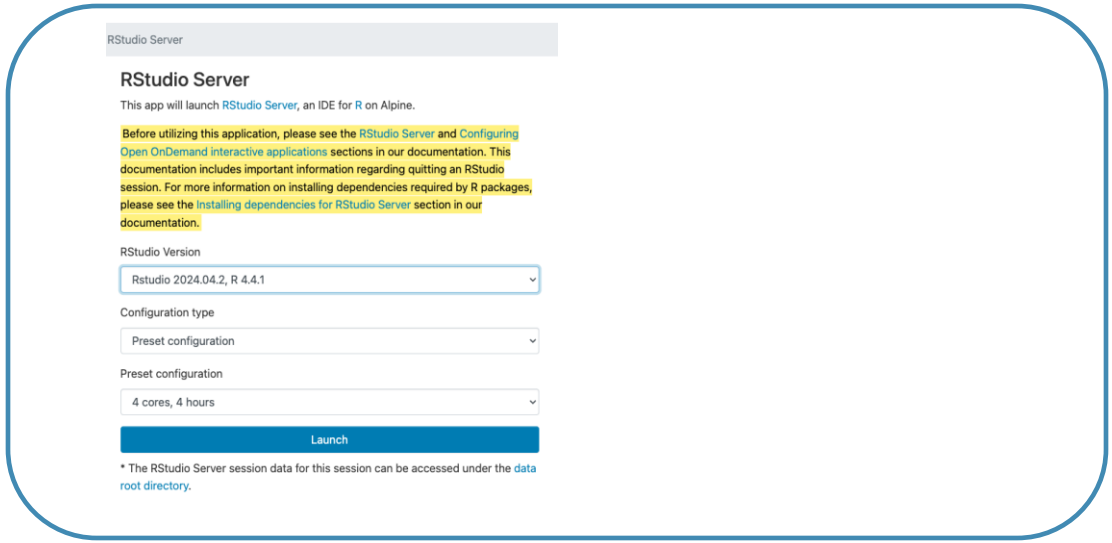
There are 3 totally distinct ways to use R on Alpine.



Preliminary conditions for each way

There are 3 totally distinct ways to use R on Alpine

- Rstudio on Ondemand with versions **4.4.1** and and **4.2.2**. The version **4.4.1** is strongly recommended.



- Under the Alpine shell, there is an R version 4.4.0 that is accessible by using **LMOD**, the modern environment module system for HPC.

```
Currently Loaded Modules:
  1) jdk/18.0.1.1   2) R/4.4.0

module load R/4.4.0
module list
```

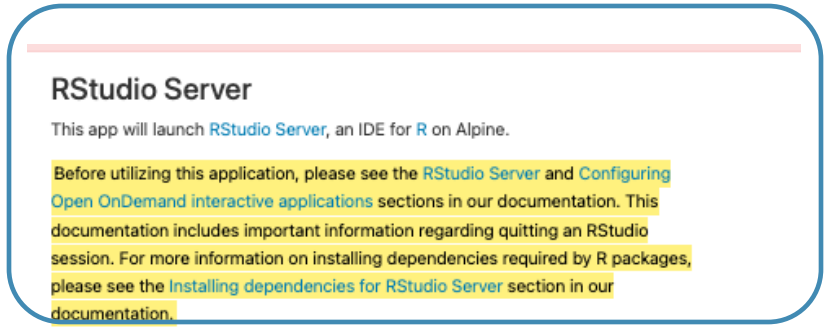
- One may decide to install R through **miniforge** or anaconda. The latest version they can install is R **4.4.2**

```
module load miniforge
```

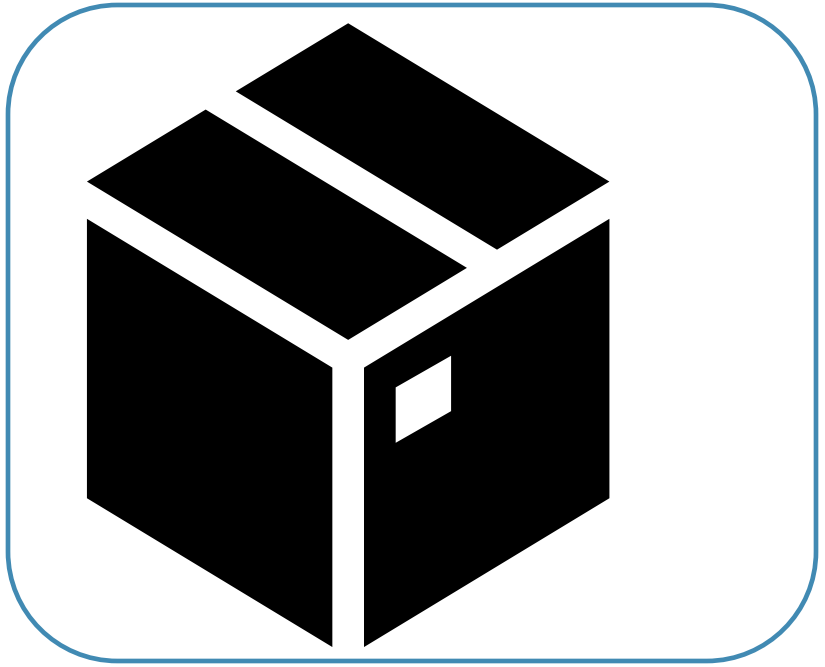
I) Rstudio on Alpine

What is Ondemand Rstudio?

- It was built on top of an apptainer container image, thus, it runs on a different operating system than Alpine.
- A container is a tool that allows you to run different applications that were built on different operating systems on Alpine.
- The Rstudio .sif image is located at */curc/sw/containers/open_ondemand/rstudio-server-4.4.1.sif*



Rstudio



Container

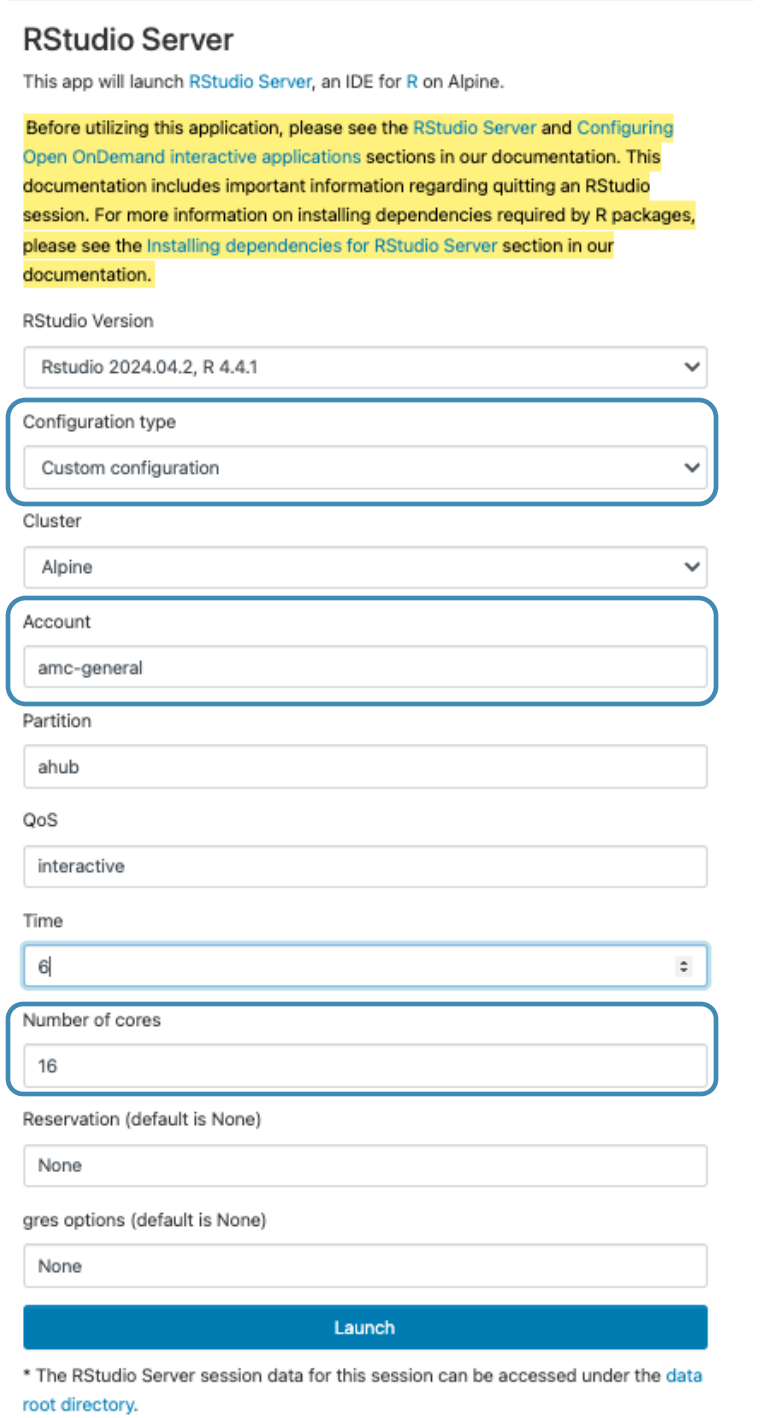
I) Rstudio on Alpine

Use Rstudio on Alpine if:

- If you are used to Rstudio outside of Alpine
- You will need access to a GUI.
- If you think that most of your pipelines will not require more than 16 cores or 60GB of RAM.
- **Note: *** If you are new to Rstudio on Alpine, please refer to this guide: https://github.com/kf-cuanschutz/CU-Anschutz-HPC-documentation/tree/main/Rstudio_related_scripts

* If you wish to run your Rstudio contained environment as a slurm batch script, please refer to this guide: https://github.com/kf-cuanschutz/CU-Anschutz-HPC-documentation/blob/main/Rstudio_on_Slurm_.md

University of Colorado
Anschutz Medical Campus



II) LMOD R

What is LMOD R?

- It is accessible through the use of **LMOD**, the modern environment module system for HPC.
- You need to be on a compute node interactively or you need to submit a slurm batch script.
- We only recommend to use the version **4.4.0** to install your packages!!!
- Note that some advance package installation might requires some dependencies. Please open a ticket at hpcsupport@cuanschutz.edu if you are stuck!

```
acompile --ntasks=4 --time=12:00:00
```

Compute node

```
module avail R
```

Check for LMOD R

```
----------------
   R/3.6.3
   R/4.2.2
   R/4.4.0
```

```
module load R/4.4.0
which R
/curc/sw/install/R/4.4.0/bin/R
```

Load R/4.4.0 and confirm

II) LMOD R

Use LMOD R if:

- You will run heavy job on Alpine, such as high memory computation jobs.
- You will not need to use a GUI interactively.
- You need to run distributed computations, such as with gnu parallel.

```
#!/bin/bash

#SBATCH --partition=amilan              #
#SBATCH --job-name=Rscript_test
#SBATCH --output=Rscript_test.%j.out    #
#SBATCH --error=Rscript_test.%j.err     #
#SBATCH --account=amc-general  # Account
#SBATCH --qos=normal   # Quality of serv
#SBATCH --nodes=1    # Total number of n
#SBATCH --ntasks=4   # Total number of c
#SBATCH --mail-type=ALL  # Begin, End an
#SBATCH --mail-user=kevin.fotso@cuanschu
#SBATCH --time=00:00:01


# Load the compiler that R  depends on.
module load R/4.4.0

# Run the R command
Rscript R_test.
```

III) Miniforge R

What is miniforge/anaconda R?

- It gives you the possibility to install many R related packages through miniforge.

- Note that your miniforge R located in your miniforge environment will be very distinct from LMOD R!!!

III) Miniforge R

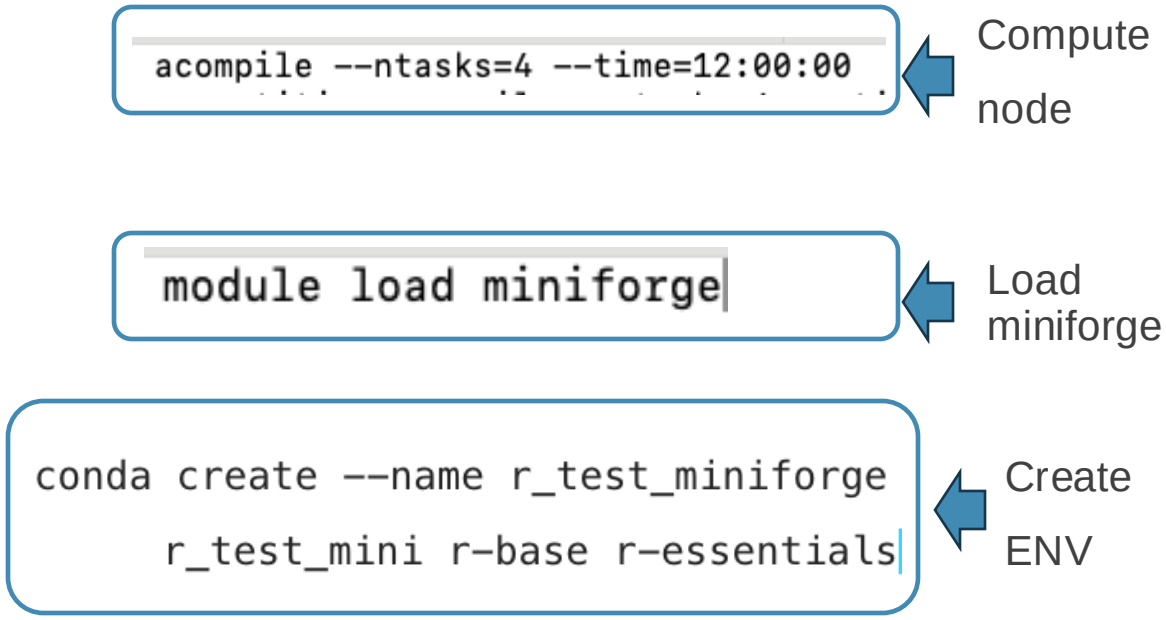
Use miniforge/anaconda R if:

- You are a frequent miniconda/anaconda user.
- If you have followed this guide already:

  https://github.com/kf-cuanschutz/CU-Anschutz-HPC-documentation/blob/main/miniforge_migration.md

- If you plan to share the environment you are using with the rest of your lab on Alpine. For example, one can export a miniforge environment as yml.

- Consider this r-irkernel guide if you plan on using Jupyterlab: https://curc.readthedocs.io/en/latest/open_ondemand/jupyter_session.html



III) Miniforge R

Use miniforge/anaconda R if:

- You are a frequent miniconda/anaconda user.
- If you have followed this guide already:

  https://github.com/kf-cuanschutz/CU-Anschutz-HPC-documentation/blob/main/miniforge_migration.md

- If you plan to share the environment you are using with the rest of your lab on Alpine. For example, one can export a miniforge environment as yml.

- Consider this r-irkernel guide if you plan on using Jupyterlab: https://curc.readthedocs.io/en/latest/open_ondemand/jupyter_session.html

```
acompile --ntasks=4 --time=12:00:00
```

Compute node

```
module load miniforge
```

Load miniforge

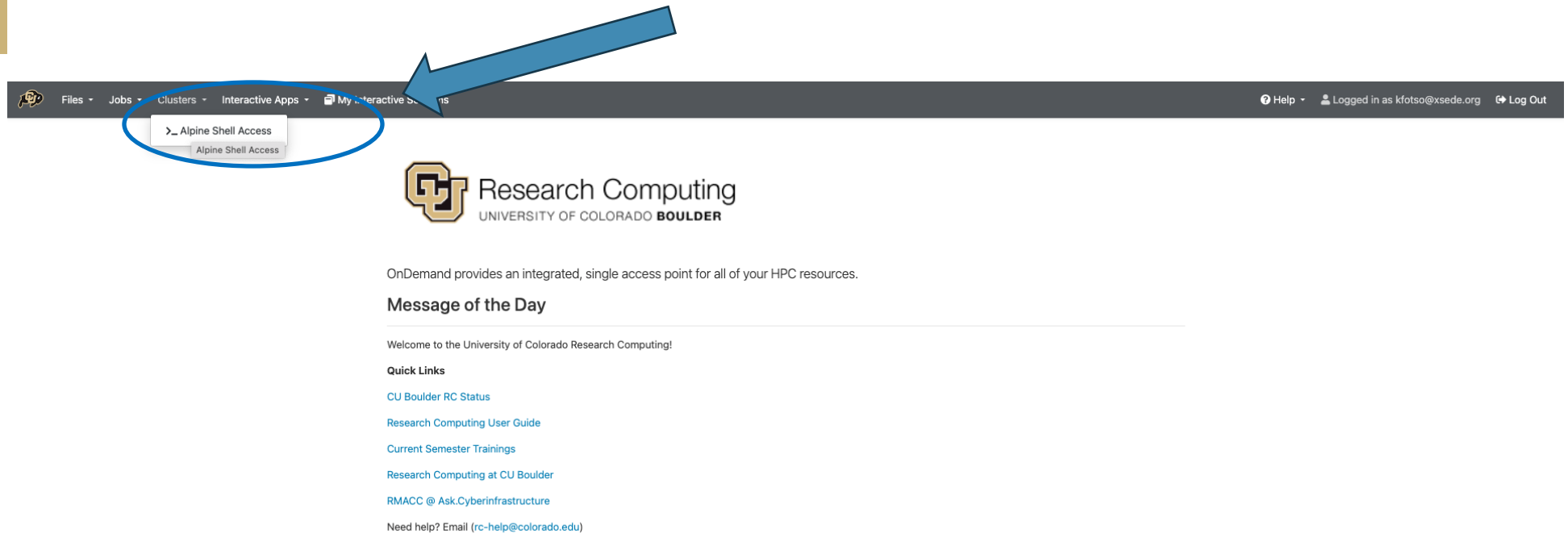
```
conda create --name r_test_miniforge
     r_test_mini r-base r-essentials
```

Create ENV

Packages installed under:
/projects/$USER/software/anaconda/envs

How to monitor GPU jobs?

- Remember to be inside an Alpine shell!



How to monitor GPU jobs?

```
sede.org@login-ci1 ~]$ squeue --me --long
1 09:46:17 2025
    JOBID PARTITION     NAME     USER    STATE       TIME TIME_LIMI  NODES NODELIST(REASON)
 12400310 atesting_ sinterac kfotso@x  RUNNING       0:54   1:00:00      1 c3gpu-c2-u13
sede.org@login-ci1 ~]$ ssh c3gpu-c2-u13
```

- Use ”squeue” to get the address of the gpu node

  where your job is running.

How to monitor GPU jobs?

```
sede.org@login-ci1 ~]$ squeue --me --long
1 09:46:17 2025
    JOBID PARTITION     NAME     USER    STATE       TIME TIME_LIMI  NODES NODELIST(REASON)
 12400310 atesting_ sinterac kfotso@x  RUNNING       0:54   1:00:00      1 c3gpu-c2-u13
sede.org@login-ci1 ~]$ ssh c3gpu-c2-u13
```

- Use the "**ssh**" command to access that node.
- "**ssh**" is a network protocol that means secure shell.

How to monitor GPU jobs?

```
3 ~]$ watch -n 1 nvidia-smi -l
3 ~]$
```

- Watch is a command that allows to constantly **refresh/update** the screen and the command being used

How to monitor GPU jobs?

```
3 ~]$ watch -n 1 nvidia-smi -l
3 ~]$
```

- Our main command to monitor gpu jobs is nvidia-smi.

How to monitor GPU jobs?

```
Every 1.0s: nvidia-smi -l

Fri Mar 21 09:47:51 2025
+-----------------------------------------------------------------------------------------+
| NVIDIA-SMI 550.90.12              Driver Version: 550.90.12      CUDA Version: 12.4     |
|-----------------------------------------+------------------------+----------------------+
| GPU  Name                 Persistence-M | Bus-Id          Disp.A | Volatile Uncorr. ECC |
| Fan  Temp   Perf          Pwr:Usage/Cap |           Memory-Usage | GPU-Util  Compute M. |
|                                         |                        |               MIG M. |
|=========================================+========================+======================|
|   0  NVIDIA A100-PCIE-40GB          On  |   00000000:21:00.0 Off |                   On |
| N/A   31C    P0             52W /  250W |      75MiB /  40960MiB |     N/A      Default |
|                                         |                        |              Enabled |
+-----------------------------------------+------------------------+----------------------+
|   1  NVIDIA A100-PCIE-40GB          On  |   00000000:81:00.0 Off |                   On |
| N/A   28C    P0             34W /  250W |      75MiB /  40960MiB |     N/A      Default |
|                                         |                        |              Enabled |
+-----------------------------------------+------------------------+----------------------+
|   2  NVIDIA A100-PCIE-40GB          On  |   00000000:E2:00.0 Off |                   On |
| N/A   29C    P0             34W /  250W |      75MiB /  40960MiB |     N/A      Default |
|                                         |                        |              Enabled |
+-----------------------------------------+------------------------+----------------------+

+-----------------------------------------------------------------------------------------+
| MIG devices:                                                                            |
+------------------+--------------------------------+-----------+-----------------------+
| GPU  GI  CI  MIG |                     Memory-Usage |        Vol|      Shared           |
|      ID  ID  Dev |                       BAR1-Usage | SM     Unc| CE ENC DEC OFA JPG    |
|                  |                                |        ECC|                       |
|==================+================================+===========+=======================|
|  0    1   0   0  |              38MiB / 19968MiB | 42      0 |  3   0    2    0    0 |
|                  |                 0MiB / 32767MiB |           |                       |
+------------------+--------------------------------+-----------+-----------------------+
|  0    2   0   1  |              38MiB / 19968MiB | 42      0 |  3   0    2    0    0 |
|                  |                 0MiB / 32767MiB |           |                       |
+------------------+--------------------------------+-----------+-----------------------+
|  1    1   0   0  |              38MiB / 19968MiB | 42      0 |  3   0    2    0    0 |
|                  |                 0MiB / 32767MiB |           |                       |
+------------------+--------------------------------+-----------+-----------------------+
```

- GPU memory utilization
- Consider installing and using **nvitop** for a more detailed analysis, including for the kernel:

  https://github.com/XuehaiPan/nvitop

Need further assistance or resources?

- Visit our github page that has many useful tutorials and workshops geared towards CU Anschutz users: https://github.com/kf-cuanschutz/CU-Anschutz-HPC-documentation

- Visit the Alpine CURC documentation: https://curc.readthedocs.io/en/latest/getting_started/navigating_docs.html

- Email us at rc-help@colorado.edu to open a ticket


University of Colorado Anschutz Medical Campus
THANK YOU