

# Introduction to Alpine

Kevin Fotso



# Logging into Alpine(1)

## Logging into Alpine from the Shell App

### Note

Make sure you already have your XSEDE/ACCESS user name and password set up before proceeding and Duo 2-factor authentication set up for your ACCESS/XSEDE account

1. Visit <https://ondemand-rmacc.rc.colorado.edu> You will be redirected to CILogon. From there, make sure you select the ACCESS CI (XSEDE) as your identity provider and then click the "Log On" button.

### CILogon

Consent to Attribute Release

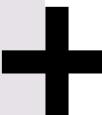
OOD RMACC requests access to the following information. If you do not approve this request, do not proceed.

- Your CILogon user identifier
- Your name
- Your email address
- Your username and affiliation from your identity provider

Select an Identity Provider

ACCESS CI (XSEDE) 

Remember this selection 



# Logging into Alpine(2)

The screenshot shows the OnDemand web interface for the University of Colorado Research Computing system. At the top, there is a navigation bar with icons for Files, Jobs, Clusters (which is currently selected and highlighted in black), Interactive Apps, and My Interactive Sessions. A red circle highlights the 'Clusters' dropdown menu, which is open to show two options: '\_Alpine Shell' and '\_Blanca Shell'. Below the navigation bar is the CU Boulder Research Computing logo, featuring a gold 'CU' monogram and the text 'Research Computing UNIVERSITY OF COLORADO BOULDER'. A main message states: 'OnDemand provides an integrated, single access point for all of your HPC resources.' Below this is a 'Message of the Day' section with the text: 'Welcome to the University of Colorado Research Computing.' Underneath, there is a 'Quick Links' section with several blue hyperlinks: 'CU Boulder RC Status', 'Research Computing User Guide', 'Research Computing at CU Boulder', 'RMACC @ Ask.Cyberinfrastructure', and 'Need help? Email ([rc-help@colorado.edu](mailto:rc-help@colorado.edu))'. A large black plus sign is positioned in the bottom right corner of the page.

OnDemand provides an integrated, single access point for all of your HPC resources.

## Message of the Day

---

Welcome to the University of Colorado Research Computing.

### Quick Links

[CU Boulder RC Status](#)  
[Research Computing User Guide](#)  
[Research Computing at CU Boulder](#)  
[RMACC @ Ask.Cyberinfrastructure](#)  
Need help? Email ([rc-help@colorado.edu](mailto:rc-help@colorado.edu))

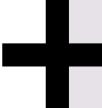
# ssh project

Worked with the Boulder team, to establish ssh over VPN.

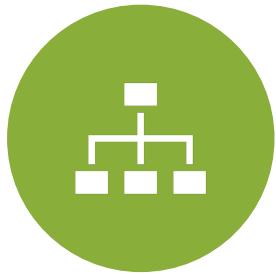
Successfully passed Beta mode and Beta plus mode.

Still working with new group of users

Not official yet but you can email us!



# Files transfer (Globus)



Creation of Globus Connect Personal endpoint.



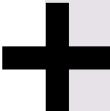
CU Boulder endpoint for Anschutz is CU Boulder Research Computing ACCESS.



Data Transfer node (dtn3) is 10 years old (1GB/sec)



Boulder is building 4 new DTN and upgrading configuration (2-4 weeks)



# Storage(1)

Home filesystem (2G). Backed up + for hosting config files.

Project filesystem (250G). Backed up -> for package installation

“cd .snapshot” to access those back ups

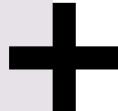
‘curc-quota’ or ‘du’ to check on space

# Storage(2)

Maintenance period:

```
[kfotso@xsede.org@login-ci1 ~]$ curc-quota
```

	Used	Avail	Quota	Limit
/home/kfotso@xsede.org	289M	1.8G	2.0G	
/projects/kfotso@xsede.org	144G	107G	250G	
/scratch/alpine1	7786G	20825G	28611G	



## Storage(3)

- Scratch space (10 TB)
- GPFS filesystem
- Very suitable for parallel application + heavy I/O
- Purged every 90 days

# Petalibrary

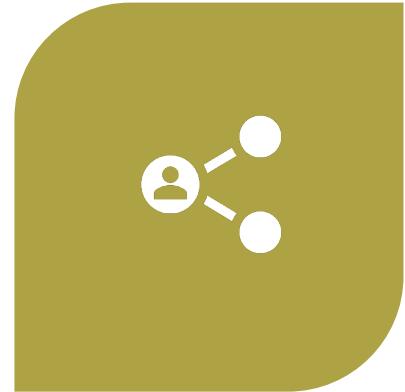
+

# Preliminary conditions (1)

---

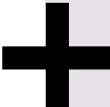


CREATION OF AN ACCESS  
GROUP

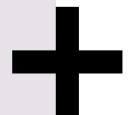
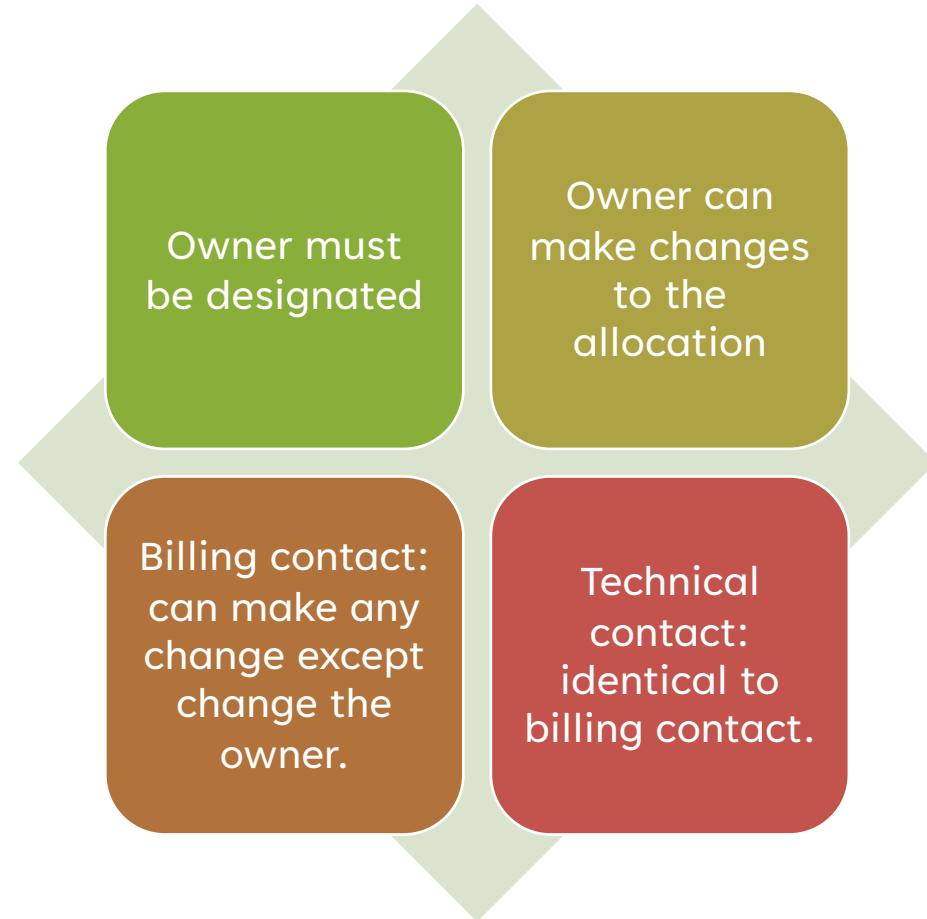


ALL MEMBER ADDED TO THE  
GROUP WILL NEED  
ACCESS/XSEDE ACCOUNTS.

/pl/active/<your\_allocation\_name>  
/pl/archive/<your\_allocation\_name>



# Preliminary conditions (2)



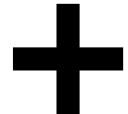


# Application

- Application submitted here:  
<https://www.colorado.edu/rcc/resources/petalibrary>
- The request form will need a speedtype: account# to which they plan to charge the allocation.

# Billing (1)

- On active storage: \$45/TB/yr.
- ZFS Raidz2 allow for frequent read/write + parity.
- **It is highly suggested that total TB size for the year is determined in advanced by the owner.**



## Billing (2)

- On archive storage: \$20/TB/yr.
- Tape-like storage for infrequently accessed data.
- Can be accessed from RC login node only and data transfer resources.
- Min size for any alloc is 1 TB



# Terms of usage

- Data will need to be in full compliance with term of service.
- No PHI data, no FERPA, no ITAR, no GDPR and data that comply with IRB requirements.
- More information here:  
<https://www.colorado.edu/rc/resources/petalibrary/tos>
- Data classification:  
<https://www.cu.edu/security/data-classification>.
- HIPAA identifiers:  
<https://www.dhcs.ca.gov/dataandstats/data/Pages>ListofHIPAAIdentifiers.aspx>



# Data redundancy (1)



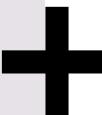
PL allocations are of single-copy nature.



All users should fill out the PL single copy acknowledgment.



Snapshots monitoring in place so that they are not missed unless on snapshot custom schedule



# Sort term back up solutions (1)

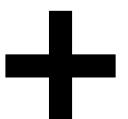
- Microsoft OneDrive: 5TB per person through A3 licensing.
- S3 buckets: customer paid & rates located here:  
<https://aws.amazon.com/s3/pricing/?p=pm&c=s3&z=4>
- Multiple PL allocations as though customer requested 2 allocations (X2 price).



# Sort+mid term back up solutions (2)

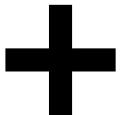
---

- [BETA]  
Replicated PetaLibrary active  
+archive allocation.
- Nothing is charged during  
beta period but will likely look  
like Multiple PL allocations.
- CURC **only takes data  
replication responsibility** for  
BETA

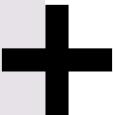


# Active+Archive replication

- Synchronization between active and archive happen every 15 min in theory but are not always guaranteed.
- Snapshots are also replicated from active to archive.
- The Boulder storage team maintains the second copy (e.g. archive) for you and you do not have direct access to it.
- \$65/TB/year



# Alpine computing



# Official Github pages:

- CU Anschutz HPC official Github page.

kf-cuanschutz / CU-Anschutz-HPC-documentation Public

Code Issues 2 Pull requests Actions Projects Security Insights Settings

main 1 branch 0 tags Go to file Add file ▾ Code ▾

**kf-cuanschutz Update README.md ...** a9741f6 2 weeks ago 63 commits

- Globus-local-entry-point-files Delete Overview\_of\_Petalibrary.pdf 2 weeks ago
- Office-hours-presentation-files Introduction to Petalibrary 2 weeks ago
- ALDEEx2-R-package-installation.md Fixed the hyperlink that was pointing to globus 3 months ago
- Alpine-cluster-maintenance.md replaced Preventative maintenance with FAQ 3 months ago
- Alpine-pipeline-opt-FAQ.md Added Job arrays upper limit 2 months ago
- EcholocateR.md small typo fix 3 months ago
- MATLAB-kernel-on-Jupyterlab.md Create MATLAB-kernel-on-Jupyterlab.md 2 weeks ago
- README.md Update README.md 2 weeks ago
- cellRangerRkit.md Few changes added (e.g. indentation on Part 1) 2 months ago

**README.md**

CC BY NC ND

**CU Anschutz-HPC-documentation**

About All documentation associated with bioinformatics tutorial on the Alpine cluster

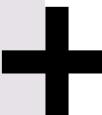
Readme Activity 6 stars 1 watching 0 forks

Releases No releases published Create a new release

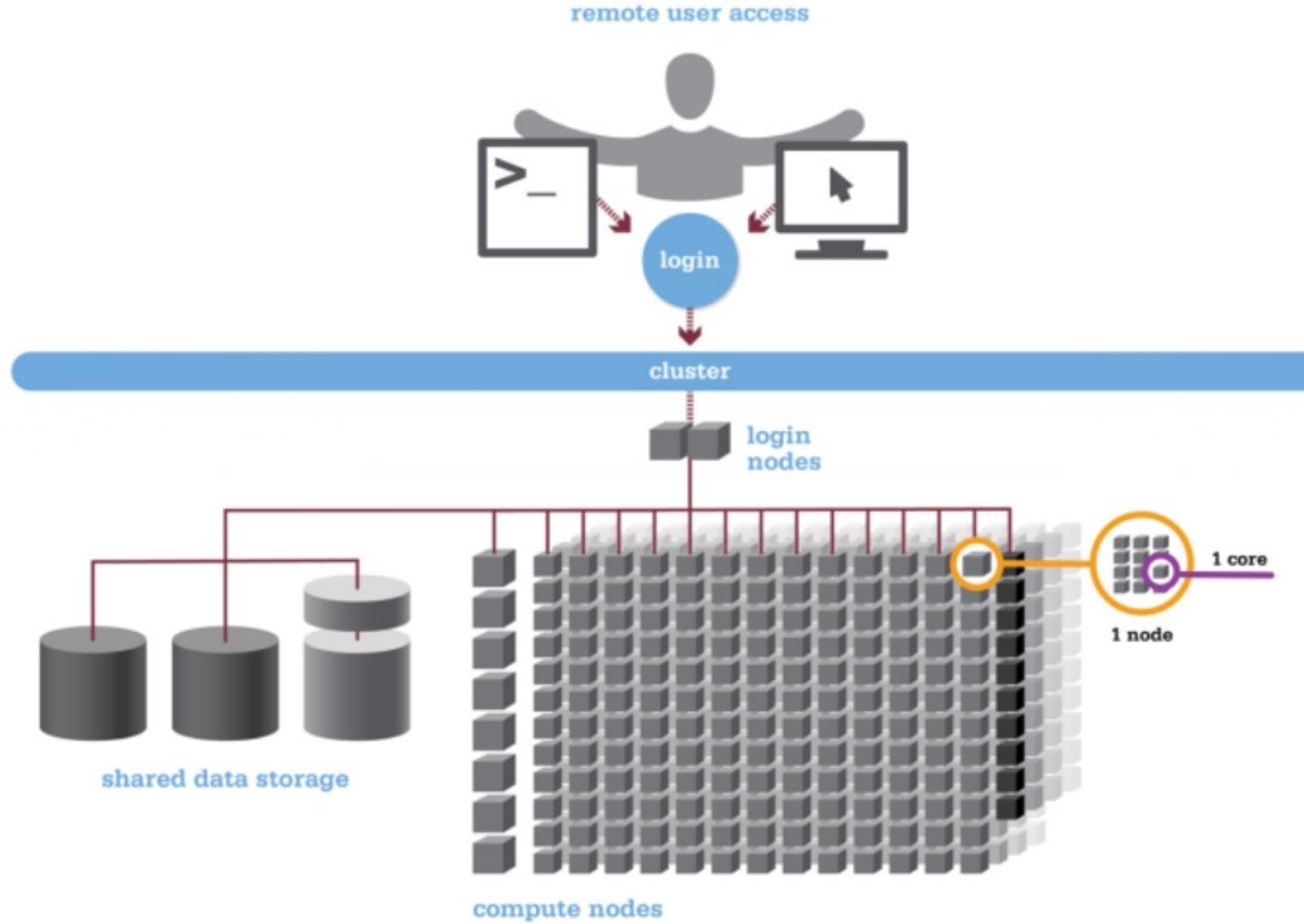
Packages No packages published Publish your first package

- CU Boulder curc doc:

<https://curc.readthedocs.io/en/latest/access/logging-in.html>



# Architecture of a supercomputer



- Login nodes. To log into the system, cd into directories, look at files etc ...
- Compute node. Dedicated to do the computation
- The slurm scheduler controls access to the compute nodes to avoid a tragedy of the commons ...

# Hardware (1)



317 compute nodes and 18,080 nodes officially.



184 CPU nodes (HDR IB interconnect)



12 high memory nodes (1TB)



8 NVIDIA A100 GPU and 8 AMD GPU MI100 nodes. (3 GPUs per node) + (2X25 Ethernet interconnect)



NVIDIA GPU tend to be more busy but AMD GPU are popular.

## Hardware (2)

GPU debug nodes are now available with --qos=atesting\_a100.

1 hour and up to 2 GPUs.

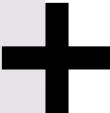
Users are now limited to up to 2/3 of the GPU partition (not per node)

# Debug nodes for NVIDIA A100 and AMD MI100

- sinteractive --partition=atesting\_a100 --qos=testing --time=1:00:00 --gres=gpu:1 --ntasks=1
- sinteractive --partition=atesting\_mi100 --qos=testing --time=1:00:00 --gres=gpu:1 --ntasks=1

# Jupyterlab with GPUs

- Requires registration to ssh
- ssh forwarding



# Username Aliasing

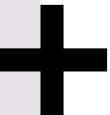
---

- `/home/foo@xsede.org` → `/home/.xsede.org/foo/`
- `/projects/foo@xsede.org` → `/projects/.xsede.org/foo/`
- `/scratch/alpine/foo@xsede.org` → `/scratch/alpine/.xsede.org/foo/`



# Scheduler Slurm

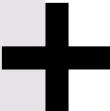
- **acompile --ntasks=1 --time=00:30:00** to build packages and do some [REDACTED] testing.
- **sinteractive --ntasks-per-node=2 --nodes=2 --partition=atesting** to test pipelines
- NVIDIA gpu partitions are aa100, amc and atesting\_a100.
- AMD gpu partitions are ami100 and atesting\_mi100



```
[kfotso@xsede.org@login-ci1 kfotso@xsede.org]$ sinfo --partition=aa100
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
aa100          up 1-00:00:00      1  resv  c3gpu-a9-u33-1
aa100          up 1-00:00:00      7  mix   c3gpu-a9-u31-1,c3gpu-a9-u35-1,c3gpu-c2-u
aa100          up 1-00:00:00      4  alloc  c3gpu-a9-u29-1,c3gpu-c2-u[7,13,15]
```

# sinfo

Can be used to get  
information about a node.



# Slurm example

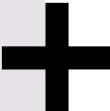
```
#!/bin/bash

#SBATCH --partition=amilan
#SBATCH --job-name=example-job
#SBATCH --output=example-job.%j.out
#SBATCH --time=01:00:00
#SBATCH --qos=normal
#SBATCH --nodes=1
#SBATCH --ntasks=4
#SBATCH --mail-type=ALL
#SBATCH --mail-user=youridentikey@colorado.edu

module purge
module load anaconda
conda activate custom-env

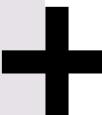
python myscript.py
```

- Partition is the type of node
- Qos is the quality of service
- ntasks are the number of cores
- Sbatch slurm script



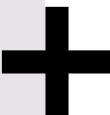
# Slurm cheatsheet (1)

Slurm script command	Description
<code>#!/bin/bash</code>	Sets the shell that the job will be executed on the compute node
<code>#SBATCH --ntasks=1</code> <code>#SBATCH --n1</code>	Requests for 1 processors on task, usually 1 cpu as 1 cpu per task is default.
<code>#SBATCH --time=0-05:00</code> <code>#SBATCH -t 0-05:00</code>	Sets the maximum runtime of 5 hours for your job
<code>#SBATCH --mail-user= &lt;email&gt;</code>	Sets the email address for sending notifications about your job state.
<code>#SBATCH --mail-type=BEGIN</code> <code>#SBATCH --mail-type=END</code> <code>#SBATCH --mail-type=FAIL</code> <code>#SBATCH --mail-type=REQUEUE</code> <code>#SBATCH --mail-type=ALL</code>	Sets the scheduling system to send you email when the job enters the following states: BEGIN,END,FAIL,REQUEUE,ALL
<code>#SBATCH --job-name=my-named-job</code>	Sets the Jobs name



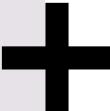
# Slurm cheatsheet(2)

Slurm script command	Description
#SBATCH --ntasks=X	Requests for X tasks. When cpus-per-task=1 (and this is the default) this requests X cores. When not otherwise constraint these CPUs may be running on any node
#SBATCH --nodes=X	Request that a minimum of X nodes be allocated to this job
#SBATCH --nodes=X-Y	Request that a minimum of X nodes and a maximum of Y nodes be allocated to this job
#SBATCH --cpus-per-task=X	Request that a minimum of X CPUs per task be allocated to this job
#SBATCH --tasks-per-node=X	Requests minimum of X task be allocated per node



# Slurm cheatsheet(3)

Slurm script commands	Description of effects
<code>#SBATCH --ntasks=1</code> <code>#SBATCH --cpus-per-task=1</code>	<b>Requests 1 CPU (Serial)</b> cpus-per-task is set to 1 by default and may be omitted.
<code>#SBATCH --cpus-per-task=X</code> <code>#SBATCH --ntasks=1</code> <code>#SBATCH --nodes=1</code>	<b>Requests for X CPUs in 1 task on 1 node (OpenMP)</b> Both ntasks and nodes are set to 1 by default and may be omitted
<code>#SBATCH --ntasks=X</code> <code>#SBATCH --tasks-per-node=X</code> <code>#SBATCH --cpus-per-task=1</code>	<b>Requests for X CPUs and tasks on 1 node</b> cpus-per-task is set to 1 by default and may be omitted.
<code>#SBATCH --ntasks=X</code> <code>#SBATCH --nodes=1</code> <code>#SBATCH --cpus-per-task=1</code>	<b>Requests for X CPUs and tasks on 1 node</b> cpus-per-task is set to 1 by default and may be omitted.



# Get information about jobs

```
[kfotso@xsede.org@login-ci1 ~]$ squeue -l --me
```

JOBID	PARTITION	NAME	USER	STATE
2158225	acompile	acompile	kfotso@x	RUNNING

TIME	TIME_LIMI	NODES	NODELIST(REASON)
0:16	3:00	1	c3cpu-c11-u21-2

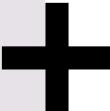
# Monitor resources

```
[kfotso@xsede.org@login-ci1 ~]$ module load slurmtools
[kfotso@xsede.org@login-ci1 ~]$ jobstats $USER 2
job stats for user kfotso@xsede.org over past 2 days
jobid      jobname  partition   qos       account    cpus state start-date-time elapsed    wait
-----
2064187    sinterac atesting_+ testing    amc-gener+  48  TIMEOUT 2023-06-20T23:32:52 01:00:04  0 hrs
2071952    vep_loft aa100        normal    amc-gener+  64  COMPLETE 2023-06-21T13:47:55 00:26:42  3 hrs
```

- Allows to get information about a past jobs

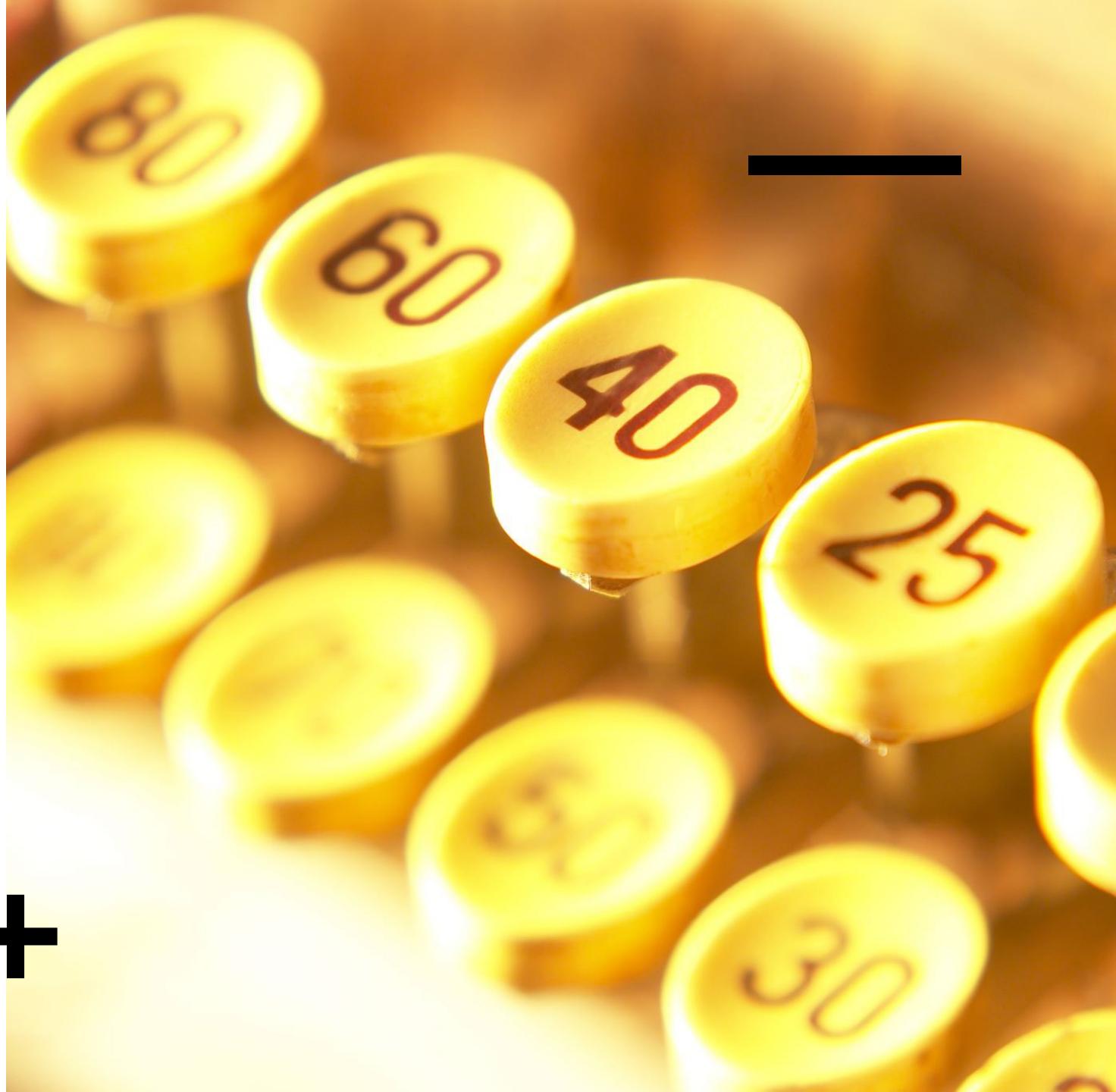
```
[kfotso@xsede.org@login-ci1 ~]$ seff 1451164
Job ID: 1451164
Cluster: alpine
User/Group: kfotso@xsede.org/kfotsopgrp@xsede.org
State: COMPLETED (exit code 0)
Nodes: 1
Cores per node: 48
CPU Utilized: 26-03:21:39
CPU Efficiency: 94.06% of 27-19:00:48 core-walltime
Job Wall-clock time: 13:53:46
Memory Utilized: 412.77 GB
Memory Efficiency: 41.28% of 999.98 GB
```

- To get more computational information about the job efficiency



# Slurm Quality of service (qos)

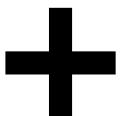
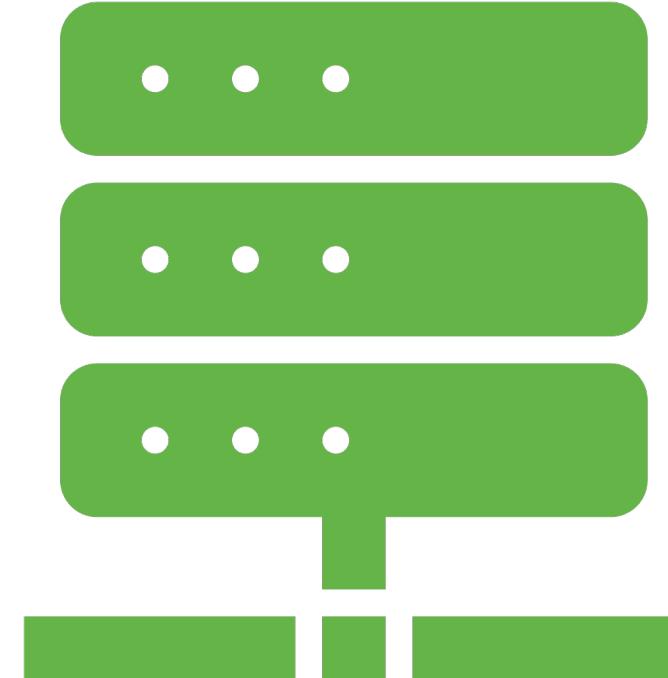
- Used to modify or constrain characteristics that a job can have.
- **--qos=normal** corresponds to a walltime of 24 hours and is the default.
- **--qos=long** corresponds to a walltime of up to 7 days
- **--qos=mem** corresponds to high memory jobs only (up to 1TB)



# Fairshare overview

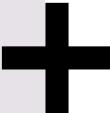
---

- Difference between the portion of computing resource that has been promised and the amount of resources that has been consumed.
- Level fairshare of 1 indicates average priority compared to other users in that account (amc-general)
- **module load slurmtools; levels \$USER**



# Job priority calculation formula

```
Job_priority =  
    site_factor +  
    (PriorityWeightAge) * (age_factor) +  
    (PriorityWeightAssoc) * (assoc_factor) +  
    (PriorityWeightFairshare) * (fair-share_factor) +  
    (PriorityWeightJobSize) * (job_size_factor) +  
    (PriorityWeightPartition) * (partition_factor) +  
    (PriorityWeightQOS) * (QOS_factor) +  
    SUM(TRES_weight_cpu * TRES_factor_cpu,  
        TRES_weight_<type> * TRES_factor_<type>,  
        ...)  
    - nice_factor
```

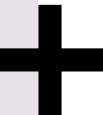


# Check fairshare

Host: login-ci1.rc.int.colorado.edu

```
[kfotso@xsede.org@login-ci1 ~]$ levelfs $USER
LevelFS for user kfotso@xsede.org and institution amc:
Account          LevelFS_User      LevelFS_Inst
-----
amc-general      0.194275        4.750220
[kfotso@xsede.org@login-ci1 ~]$
```

- 0.19 means that my priority will be low
- On the other hand 4.75 means that priority for the institution is high



# Service Units (SU)

- It is the number of core hours used.

```
[kfotso@xsede.org@login-ci1 ~]$ suuser $USER 10
SU used by user kfotso@xsede.org in the last 10 days:
Cluster|Account|Login|Proper Name|TRES Name|Used|
alpine|amc-general|kfotso@xsede.org|Kevin Fotso|billing|8393|
```

- suacct to get the number of core hours used by institution

```
Host: login-ci1.rc.int.colorado.edu
[kfotso@xsede.org@login-ci1 ~]$ suacct amc-general 180
SU used by account (allocation) amc-general in the last 180 days:
Cluster|Account|Login|Proper Name|TRES Name|Used
alpine|amc-general|||billing|1806360
alpine| amc-general|acozart@xsede.org|Abigail Cozart|billing|573
alpine| amc-general|agillen@xsede.org|Austin Gillen|billing|40320
alpine| amc-general|agray@xsede.org|Alyx Gray|billing|22
```

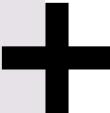
# Package availability (1)

Some packages that have been built and accessible through Imod.

Adding new packages through Imod takes a lot of round of approval so it is recommended to build them locally.

Solutions: (cmake+make), Anaconda, pip, containers, spack etc ...

Submit a ticket at [rc-help@Colorado.edu](mailto:rc-help@Colorado.edu) so that I can build it for you locally.



# List of officially available bio modules

- “acompile”
- “module avail”

`alphafold/2.2.0`  
`alphafold/2.3.1 (D)`  
`bamtools/2.5.2`  
`bbtools/39.01`

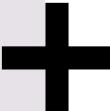
`bcftools/1.16`  
`bedtools/2.29.1`  
`bowtie2/2.5.0`  
`bwa/0.7.17`

`cellranger/7.1.0`  
`cutadapt/4.2`  
`fastqc/0.11.9`  
`gatk/4.3.0.0`

`htslib/1.16`  
`multiqc/1.14`  
`nextflow/22.10.6`  
`nextflow/23.04 (D)`

`picard/2.27.5`  
`plink2/2.00a2.3`  
`qiime2/2023.5`  
`samtools/1.16.1`

`sra-toolkit/3.0.0`  
`star/2.7.10b`  
`trimmmomatic/0.39`

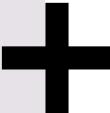


# Anaconda

---

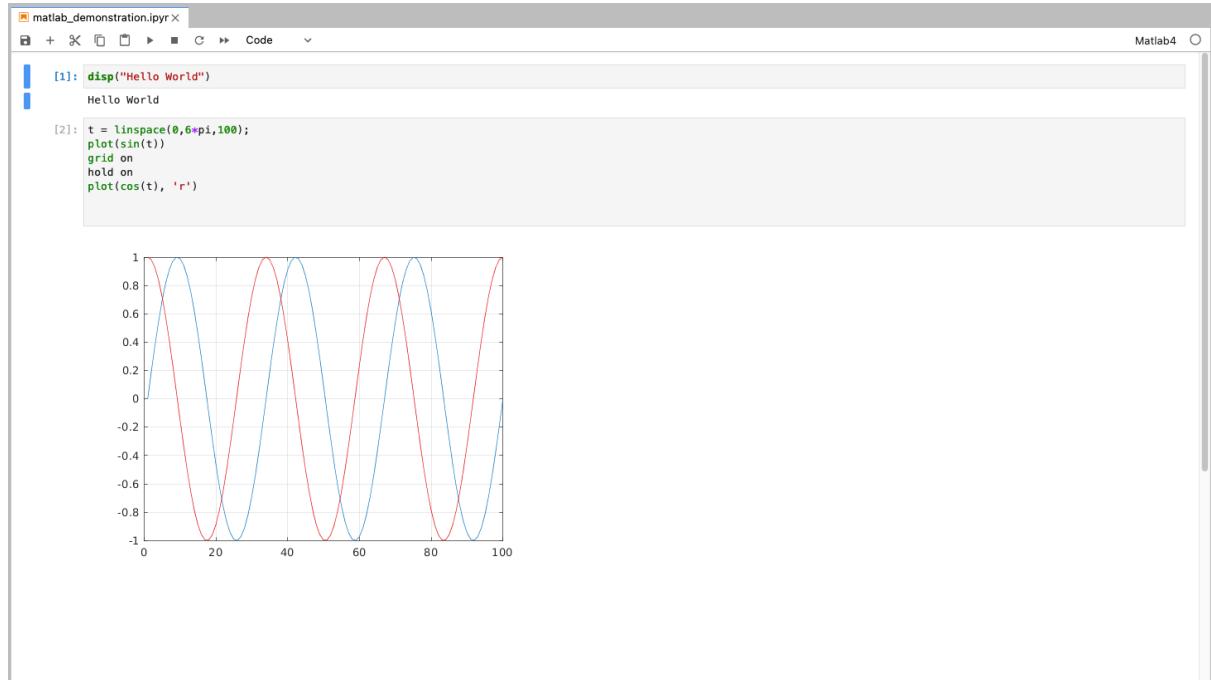
- Make sure to edit your `~/.condarc` file otherwise you might fill your home directory !!!
- In general, always make sure to check your cache in `$HOME` to make sure you are not running out of space.

```
pkgs_dirs:  
  - /projects/$USER/.conda_pkgs  
envs_dirs:  
  - /projects/$USER/software/anaconda/envs
```



# Ondemand Jupyterlab with MATLAB kernel

- Instructions here:  
<https://github.com/kf-cuanschutz/CU-Anschutz-HPC-documentation/blob/main/MATLAB-kernel-on-Jupyterlab.md>
- Possibility to use MATLAB with ssh (if registered)
- MATLAB Ondemand coming soon.



# Rstudio with Ondemand

The screenshot shows the 'Interactive Apps' section of the RStudio with Ondemand interface. The 'RStudio Server (Custom)' option is selected in the dropdown menu. The main panel displays the configuration for launching an RStudio Server session on the Alpine cluster.

**Servers**

- JupyterHub 1 (Presets)
- JupyterHub 2 (Custom)
- RStudio Server (Custom) **(Selected)**
- RStudio Server (Presets)
- JupyterHub 1 (Presets)
- JupyterHub 2 (Custom)
- RStudio Server (Custom) **(Selected)**
- RStudio Server (Presets)

**RStudio Server (Custom)**

This app will launch [RStudio Server](#), an IDE for R on Alpine.

Before utilizing this application, please see the [RStudio section of the CURC documentation](#). This documentation includes important information regarding quitting an RStudio session. For more information on possible settings for this application, see [Running Custom Interactive applications](#) in our documentation.

**RStudio Version**

Rstudio 2023.03.0, R 4.2.2

**Cluster**

Alpine

**Account**

amc-general

**Partition**

ahub

**Number of cores**

4

**Memory [GiB]**

4

**QoS Name**

# Other apps

- VS code with Ondemand coming soon

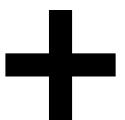


# Package availability for ML (2)

- Cuda 11.2, Cuda 11.3 and Cuda 11.4 on Alpine.
- Only cudnn 8.1 and 8.2 on Alpine.
- Can be problematic for DL build with GPU compatibility

GPU

Version	Python version	Compiler	Build tools	cuDNN	CUDA
tensorflow-2.13.0	3.8-3.11	Clang 16.0.0	Bazel 5.3.0	8.6	11.8
tensorflow-2.12.0	3.8-3.11	GCC 9.3.1	Bazel 5.3.0	8.6	11.8
tensorflow-2.11.0	3.7-3.10	GCC 9.3.1	Bazel 5.3.0	8.1	11.2
tensorflow-2.10.0	3.7-3.10	GCC 9.3.1	Bazel 5.1.1	8.1	11.2
tensorflow-2.9.0	3.7-3.10	GCC 9.3.1	Bazel 5.0.0	8.1	11.2
tensorflow-2.8.0	3.7-3.10	GCC 7.3.1	Bazel 4.2.1	8.1	11.2
tensorflow-2.7.0	3.7-3.9	GCC 7.3.1	Bazel 3.7.2	8.1	11.2
tensorflow-2.6.0	3.6-3.9	GCC 7.3.1	Bazel 3.7.2	8.1	11.2
tensorflow-2.5.0	3.6-3.9	GCC 7.3.1	Bazel 3.7.2	8.1	11.2
tensorflow-2.4.0	3.6-3.8	GCC 7.3.1	Bazel 3.1.0	8.0	11.0
tensorflow-2.3.0	3.5-3.8	GCC 7.3.1	Bazel 3.1.0	7.6	10.1



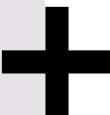
# Containers (1)



- Singularity only and it needs to be built offline and then imported back to the cluster.
- Can be built either from a definition file or converted from a docker image.
- e.g. **`sudo singularity -v build splice_conda_v7.sif splice_conda.def`**

# Containers (2)

- module load singularity
- export ALPINE\_SCRATCH=/gpfs/alpine1/scratch/\$USER
- export SINGULARITY\_TMPDIR=\$ALPINE\_SCRATCH/singularity/tmp
- export SINGULARITY\_CACHEDIR=\$ALPINE\_SCRATCH/singularity/cache  
mkdir -pv \$SINGULARITY\_CACHEDIR \$SINGULARITY\_TMPDIR



# Containers (4)

geertvandeweyer Merge pull request #2 from matthiasblum/master ... 12c3079 on Mar 23 210 commits

docker	small fix to dockerfile	8 months ago
examples	Version 1.3.1	3 years ago
spliceai	Set default port	4 months ago
tests	Changes to support running predictions in batches for better GPU ...	2 years ago
.gitignore	Version 1.3	4 years ago
COPYRIGHT	Added NOTICE file	4 years ago
LICENSE	Update LICENSE	2 years ago
NOTICE	Added NOTICE file	4 years ago
README.md	Add -P option to README	4 months ago
setup.py	Changes to support running predictions in batches for better GPU ...	2 years ago

Readme View license Activity 3 stars 0 watching 136 forks Report repository

Releases No releases published

Packages No packages published

Languages Python 96.4% Dockerfile 3.6%

**SpliceAI: A deep learning-based tool to identify splice variants**

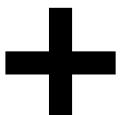
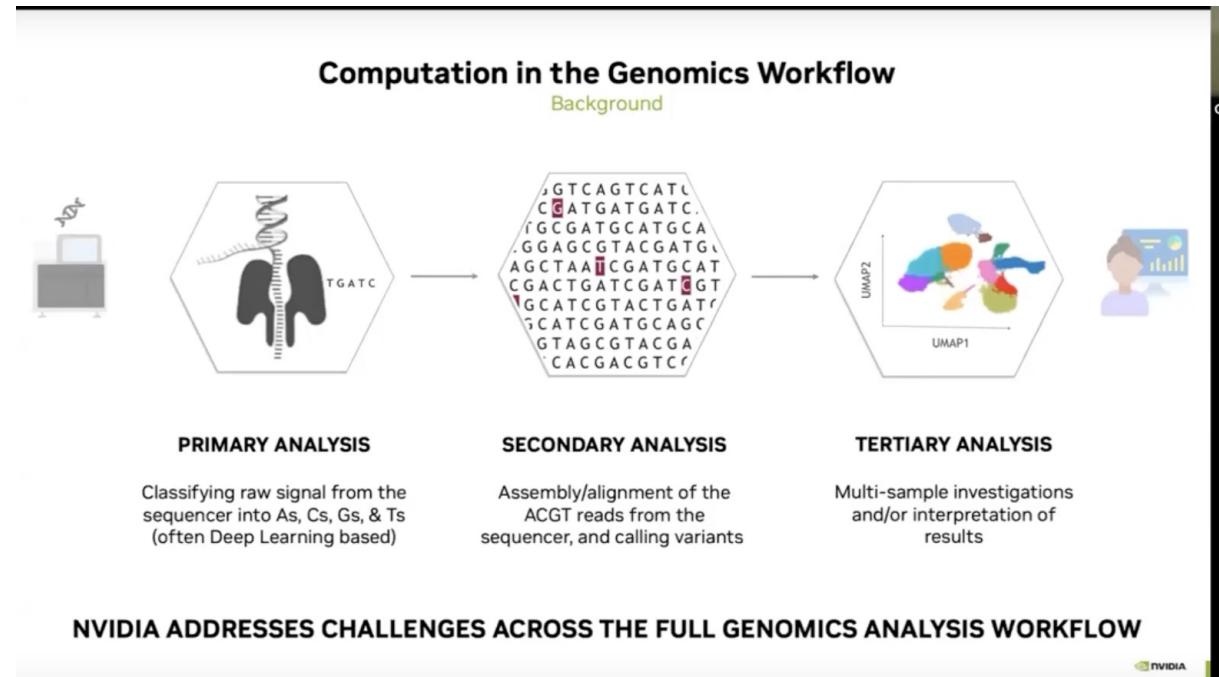
release v1.3.1 license GPLv3 downloads 152k

This package annotates genetic variants with their predicted effect on splicing, as described in Jaganathan et al, Cell 2019 in press. The annotations for all possible substitutions, 1 base insertions, and 1-4 base deletions within genes are available [here](#) for download. These annotations are free for academic and not-for-profit use; other use requires a commercial license from Illumina, Inc.

- If you need help with building containers for Alpine let us know!
- We have built a SpliceAI container for another lab and they are now heavily using GPUs.

# NVIDIA Clara Parafabrics

- Tested containers coming during the Fall.
- Nextflow, GATK etc ... on GPU
- Link to the NVIDIA series:  
[https://www.youtube.com/  
watch?v=lJuEoRXg6Jw](https://www.youtube.com/watch?v=lJuEoRXg6Jw)



A 3D rendering of Sonic the Hedgehog, a blue anthropomorphic hedgehog with green eyes and white gloves and shoes. He is shown in a dynamic pose, running towards the right while looking back over his shoulder with a wide, smiling mouth. A large, semi-transparent dark grey rectangular box covers the bottom third of the image, containing the text "Questions?".

Questions?