

# 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.

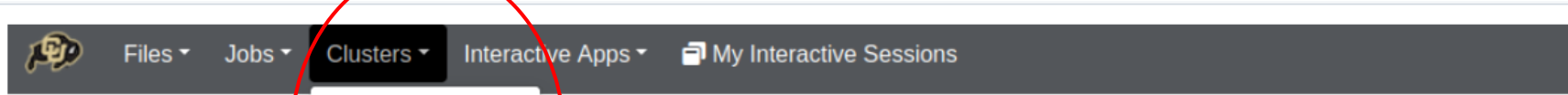
- Your CILogon user identifier
- Your name
- Your email address
- Your username and affiliation from your identity provider

### Select an Identity Provider

ACCESS CI (XSEDE) ⓘ

☐ Remember this selection ⓘ

# Logging into Alpine(2)



Research Computing  
UNIVERSITY OF COLORADO **BOULDER**

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.

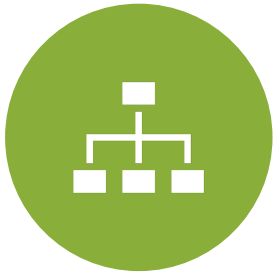
Successfully passed Beta mode and Beta plus mode.

Still working with new group of users

Not official yet



# 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)

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

# Petalibrary

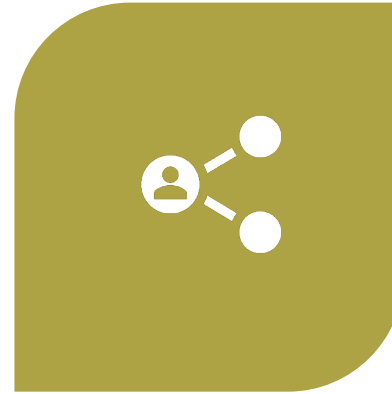




# Preliminary conditions (1)



CREATION OF AN ACCESS  
GROUP

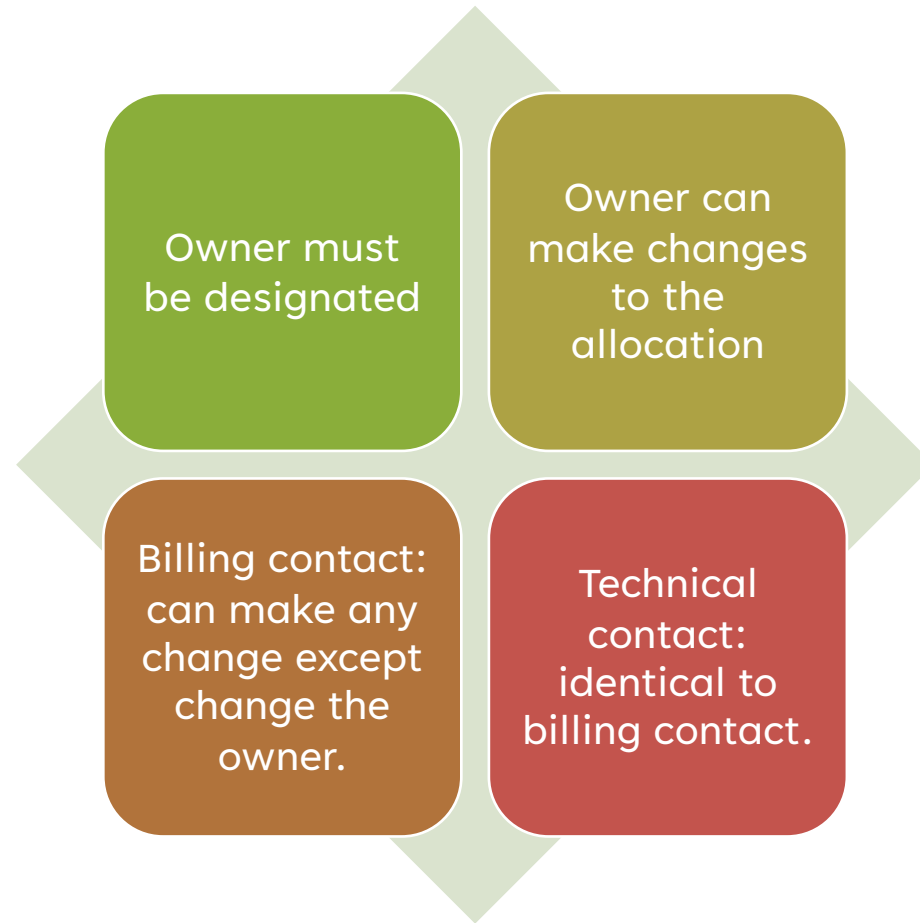


ALL MEMBER ADDED TO THE  
GROUP WILL NEED  
ACCESS/XSEDE ACCOUNTS.

```
/pl/active/<your_allocation_name>  
/pl/archive/<your_allocation_name>
```



# Preliminary conditions (2)





# Application

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





# Billing

- 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.



# 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>

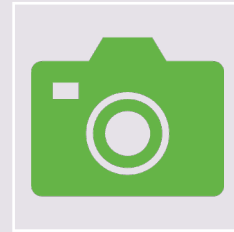
# Data redundancy (1)



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)

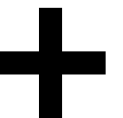
- Microsoft OneDrive: 5TB per person through A3 licensing.
- S3 buckets: customer paid & rates located here:  
<https://aws.amazon.com/s3/pricing/?p=pm&c=s3&z=4>
- Multiple PL allocations as though customer requested 2 allocations (X2 price).





# Sort 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

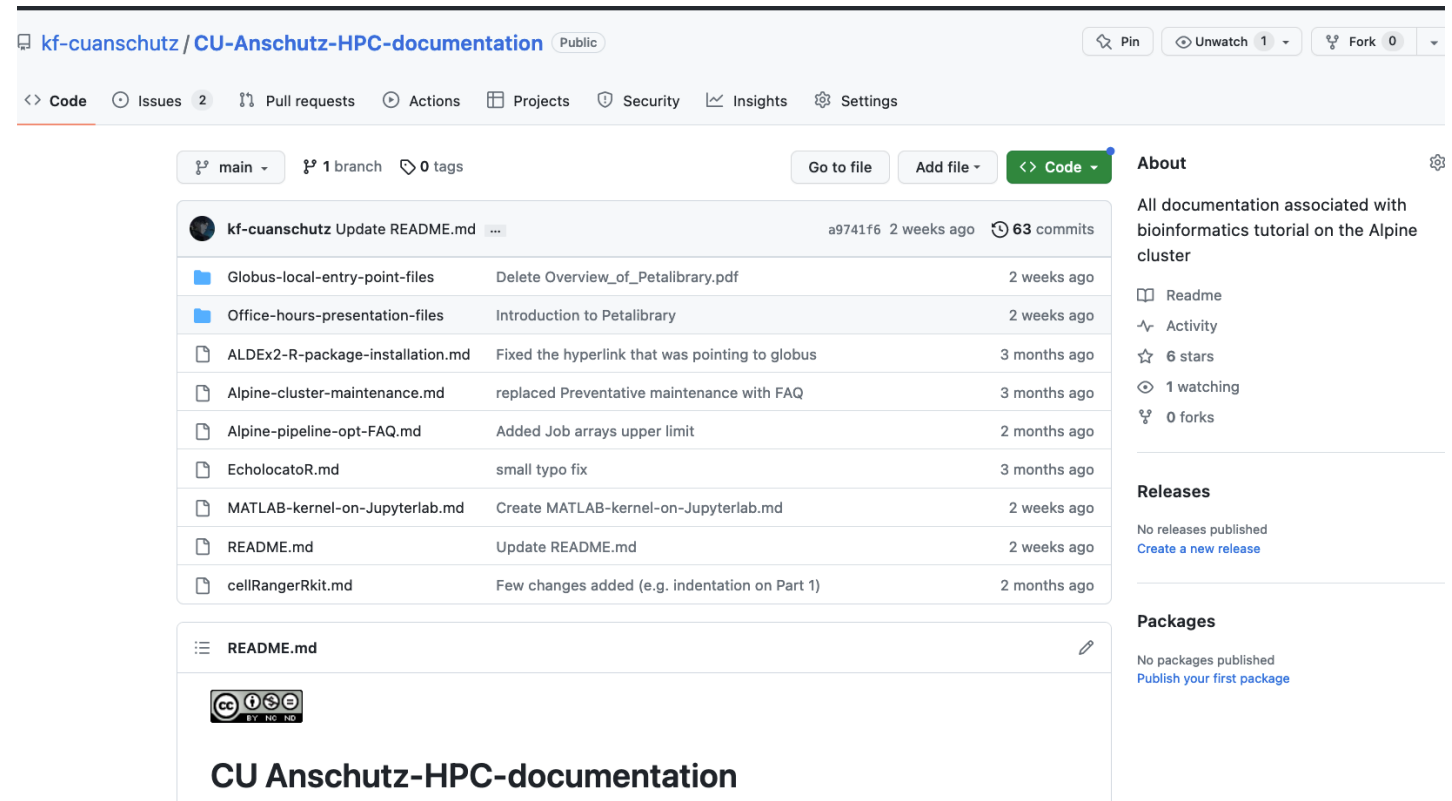


# Alpine computing



# Official Github pages:

– CU Anschutz HPC official Github page.



The screenshot shows the GitHub repository page for 'kf-cuanschutz / CU-Anschutz-HPC-documentation'. The repository is public and has 63 commits. The file list includes:

File	Description	Time
Globus-local-entry-point-files	Delete Overview_of_Petalibrary.pdf	2 weeks ago
Office-hours-presentation-files	Introduction to Petalibrary	2 weeks ago
ALDEx2-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
EcholocatoR.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

The README.md file is selected, showing a Creative Commons BY-NC-ND license and the title 'CU Anschutz-HPC-documentation'.

– CU Boulder curc doc:

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



# 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`.

1 hour and up to 2 GPUs.

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

# Scheduler Slurm

- **acompile --ntasks=1 --time=00:30:00** to build packages and do some 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



# 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

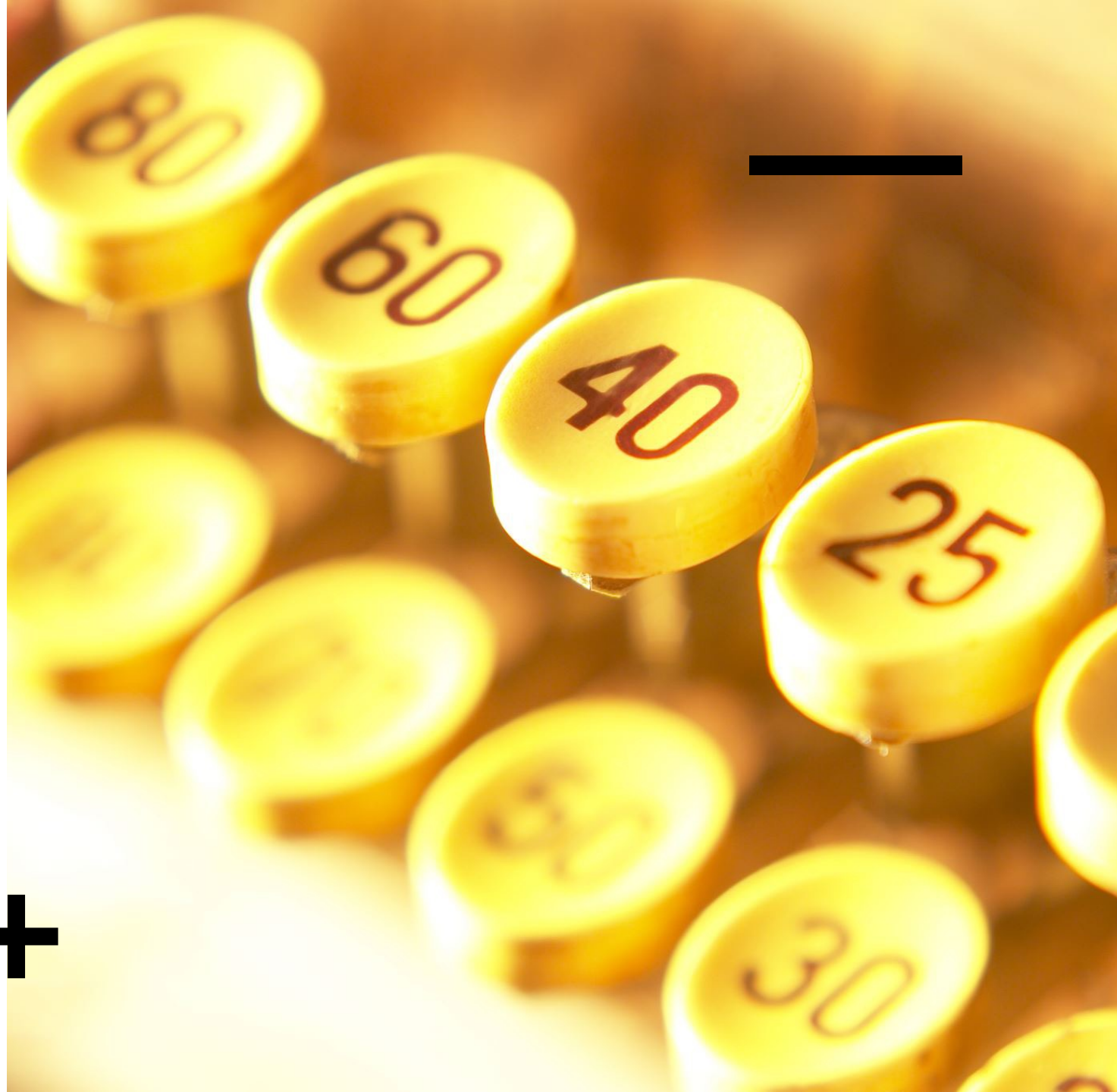
python myscript.py
```





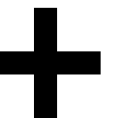
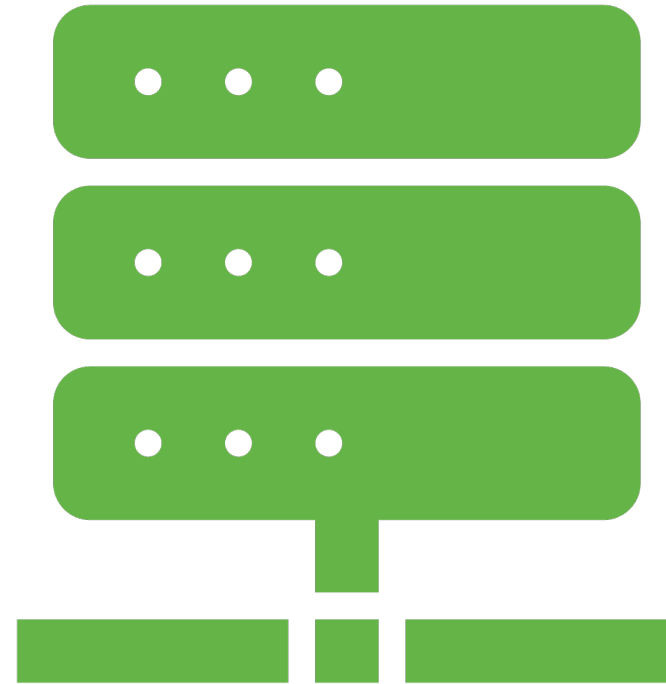
# Slurm Quality of service (qos)

- 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 slurmttools; 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
```



# 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.

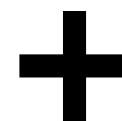


# Package availability for ML (2)

- 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







Questions?