Get Started on S3DF - Cheat Sheet

As of this writing, access to S3DF is still by invitation. You can self-register following instructions here.

This page gives a very brief introduction to SLAC's new S3DF (SLAC Shared Scientific Data Facility) cluster to help you get started. Note that, as of April 15 2024, onboarding involves automatic creation of both Active Directory ("windows") and unix accounts. The AD account is only useful for cyber training and logging into the Service Now ticket web portal.

For what follows, we assume you already have a Unix account and your main intent is to run the Fermitools/Fermipy. During the transition, issues are discussed in the #s3df-migration slack channel. You can also join the #help-sdf channel if you wish to see SLAC-wide discussion of S3DF issues.

See the main S3DF documentation for detailed information about how to log in, use the SLURM batch system, and so on. Specify --account fermi:users.

Basically, **ssh to s3dflogin.slac.stanford.edu and from there ssh to fermi-devl (no .slac.stanford.edu; it is a load balancer, but there is only one node so far) to do actual interactive work.** The login nodes are not meant for doing analysis or accessing data. Of course, real computational intensive tasks are meant for the batch system and not the interactive nodes either. Send email to **s3df-help at slac.stanford.edu** for issues.

Account maintenance

Unix Password:

SLAC currently requires a password change every 6 months. You can use https://unix-password.slac.stanford.edu/ to do this.

Cyber Training

Cyber training comes up annually. If you have an Active Directory (aka Windows) account, just follow the links.

There are issues with the training system at the moment if you only have a unix account, so here is (hopefully) temporary advice on how to navigate it (note that if you got an email saying your training is coming due, the SLAC ID (SID) is embedded in the url in the email - that is the xxxxxxx in the instructions below - if your account has not been disabled, you can ssh to rhel6-64 and issue the command:

res list user <your unix account name>

which will give your SID (along with your account status).

if none of that works, ask your SLAC Point of Contact):

You need to go to the url below; DO NOT click on forgot password. Give it your system id (SID) number (xxxxxxx).

*Note: the interim training password is "*SLACtraining2005!"*. If it does not work, email slac-training, asking them to reset it. Then go back to the original link, enter SID and this password. Then do CS100.*

https://slactraining.csod.com/

Basically, always use the SID where "user name" is requested.

passwordless ssh to fermi-devl

You can modify your .ssh config to allow direct passwordless access from your device to fermi-devl, by adding this to your .ssh/config file on your end:

```
Host slac*
        User <you>

Host slacl
        Hostname s3dflogin.slac.stanford.edu

Host slacd
        Hostname fermi-devl
        ProxyJump slacl
```

and then add your e.g. `~/.ssh/id_rsa.pub` from from your device to `~/.ssh/authorized_keys` at SLAC, using:

ssh-copy-id <you>@s3dflogin.slac.stanford.edu

For those using the cvs server on centaurusa outside from slac, you have to add a proxyjump for centaurusa. Since the cvs server is written in all CVS/Root files of a cvs package, you have to use the following solution:

```
Host centaurusa.slac.stanford.edu
        Hostname centaurusa
        ProxyJump slacl
```

.bash_profile

.bashrc:

- a directory gets added to your home dir, called profile_d. It points back to the group equivalent in /sdf/group/fermi/sw/ and includes the contents of those conf files into your session's bashrc. Group-level settings go there, eg $LATCalibRoot.
- don't overwrite your .bash_profile or you'll lose the code that does this:

```
# SLAC S3DF - source all files under ~/.profile.d
if [[ -e ~/.profile.d && -n "$(ls -A ~/.profile.d/)" ]]; then
 source <(cat $(find -L ~/.profile.d -name '*.conf'))
fi
```

Disk space

- Your home directory is in weka (/sdf/home/<first letter of your userid>/<your userid>) with 30 GB of space. This space is backed up and is where code, etc., should go.
- We have group space at /sdf/group/fermi/:
  - some directories are under /sdf/data/fermi/, but we provide links into the group directory tree for easier access
  - includes shared software, including conda envs for Fermitools and containers for running rhel6 executables
  - Fermi-supplied user (i.e., on top of your home directory) space.
    - You can find it in /sdf/group/fermi/u/<you>. There is a symlink to it, called "fermi-user", in your home directory for convenience.
    - after gpfs is retired in late 2023, this is where your larger user space will be.
  - group space in /sdf/group/fermi/g/ - a one-time copy has been done of all the gpfs g/ directories, under /nfs/farm/g/glast/g/.
  - all of glast afs has been copied to /sdf/group/fermi/a/
  - the nfs u<xx> partitions were copied to /sdf/group/fermi/n/ (including u52 which contains the GlastRelease rhel6 builds)
- your user/group space on the old clusters is **not** directly accessible from s3df
  - We're still providing additional user space from the old cluster, available on request via the slac-helplist mailing list. It is **not** backed up. This space is natively gpfs. User directories are available under: /gpfs/slac/fermi/fs2/u/<your_dir>.
    - a read-only copy of all user directories on /nfs/farm/g/glast/u was made in mid-January 2024 and can be found at /sdf/data/fermi/gpfs-u/
  - During the transition, read-only mounts of afs and gpfs are available on the interactive nodes (not batch!).
    - afs is just the normal afs path, eg to your home directory (/afs/slac/u/ ...) - you may need to issue "aklog" to get an afs token.
    - gpfs is /fs/gpfs/slac/fermi/fs2/u/ ...
- Scratch space:
  - /sdf/scratch/<username_initial>/<username>: quota 100GB/per user. The space is visible on all interactive and batch nodes. Old data will be purged when overall space is needed, even if your usages is under the quota
  - /lscratch: On each batch node, this is a local space. It is shared by all users. You are encourage to create your own sub-dir when running your job, and clean up your space (to zero) at the end of your job. Debris left behind by jobs will be purged periodically. The size of the /lscratch are subject to change and please refer to the table in Slurm partition for info about their size.

Handy urls

- **Fermi s3df accounts status**
- **fermi-user and groups usage/quotas**
- **fermi-devl server metrics**
- **slurm**
- **weka filesystem**

Access to SDF files

If you were working on SDF, note that S3DF is completely separate (aside from the account name). Even though path names might look similar, they are on different file systems. You can still access all your SDF files by prepending "/fs/ddn/" to the paths you were used to.

Software and Containers

Fermitools and other analysis software (e.g., 3ML) are available via shared Conda installation, so you don't need to install Conda yourself. See Fermitools/Conda Shared Installation at SLAC. If you do want your own Conda, you shouldn't install it in your home directory due to quota limits; put it in your Fermi-supplied user space. Follow the S3DF documentation instructions to install Conda and set a prefix path for the Conda installation that will put it and any environments you create in your group-provided space. However, you should use a prefix to your personal space, e.g., /sdf/group/fermi/u/$USER/miniconda3, instead of the path in their example.

You can also run a RHEL6 Singularity container (for apps that are not portable to RHEL/Centos7). See Using RHEL6 Singularity Container.

Slurm Batch Usage

For generic advice on running in batch, see Running on SLAC Central Linux. Note that the actual batch system has changed and we have not updated the doc to reflect that. This is advice on copying data to local scratch, etc. **If you find you cannot submit jobs to the fermi:users repo, ask for access in the #s3df-migration slack channel.**

- LSB_JOBID -> SLURM_JOB_ID
- scratch space during job execution:
  - at job start, a directory is automatically created on the scratch of the worker: ${LSCRATCH} = /lscratch/${USER} /slurm_job_id_${SLURM_JOB_ID}
  - once all of a user's jobs on a node are completed/exited, their corresponding LSCRATCH directory on that host is deleted.

You need to specify an account and "repo" on your slurm submissions. The repos allow subdivision of our allocation to different uses. There are 4 repos available under the fermi account. The format is "–-account fermi:<repo>" where repo is one of:

- default (jobs are pre-emptible - if "paying jobs" need slots, pre-emptible jobs will be killed)
- L1
- other-pipelines
- users

L1 and other-pipelines are restricted to known pipelines. Non-default repos have quality of service (qos) defaulting to normal (non-pre-emptible).

At time of writing, there is no accounting yet. When that is enabled, we'll have to decide how to split up our allocation into the various repos.

S3DF Slurm organizes the different hardware resource type under Slurm partitions. Slurm doesn't have the concept of batch queue. Users can specify the resource their job needs (because, for example a 12-core CPU request can be satisfied by different types of CPUs). The following is an example script that submits a job to Slurm:

```
#!/bin/bash
#SBATCH --account=fermi:users
##SBATCH --partition=milano
#SBATCH --job-name=my_first_job
#SBATCH --output=output-%j.txt
#SBATCH --error=output-%j.txt
#SBATCH --ntasks=1
#SBATCH --cpus-per-task=2
#SBATCH --mem-per-cpu=4g
#SBATCH --time=0-00:10:00
hostname
```

Note that the specifying "--gpus a100:1" option is preferred over the specifying "–partition=ampere" (the latter is not needed). If GPU is not requested, you job will not have access to a GPU even if it is landed on an ampere node.

Using cron

There is now a dedicated machine for cron: **sdfcron001**. Cron has been disabled on all other nodes. See:

https://s3df.slac.stanford.edu/public/doc/#/service-compute?id=s3df-cron-tasks

You might want to keep a backup of your crontab file:

crontab -l > ~/crontab.backup

Then to re-add the jobs back in:

crontab ~/crontab.backup

command line xrootd access

xrootd CLI commands are available via cvmfs:

module load osg/client-latest

which xrdcp

module unload osg/client-latest #when done with xrootd

Oracle access

Oracle drivers etc have been installed in /sdf/group/fermi/sw/oracle/. The setup.sh file in the driver-version directory sets up everything needed to issue sqlplus commands from the command line.

Using Datacat & Pipeline-II

/sdf/home/g/glast/a/datacat/prod/datacat

/sdf/home/g/glast/a/pipeline-II/prod/pipeline

cvs access

For now, we are leaving the live cvs repo on nfs. The cvs client has been installed on the iana nodes.

Set:

CVSROOT=:ext:$<USER>@centaurusa.slac.stanford.edu:/nfs/slac/g/glast/ground/cvs

Calibrations

Calibrations go to

$LATCalibRoot=/sdf/group/fermi/ground/releases/calibrations/

Write access is controlled by the glast-calibs permissions group.

the environment variable is set in the group profile. (Note: /sdf/data/fermi/a/ground/releases/calibrations is historical and not to be used)

unix group managment

If you manage any of the unix groups from the old NFS cluster (eg glast-catalog, glast-skywatch etc), maintenance is still only available from the rhel6-64 machines, using the ypgroup command. This will change once the legacy filesystems go away.