University of Colorado **Boulder**

CURC Cheat Sheet

Guide to CURC Resources

Accessing CURC

Get a Research Computing account

- https://bit.ly/3LR8XYi

Logging in from a terminal

```
ssh <username>@login.rc.colorado.edu
```

Windows/Mac Clients:

PuTTY: SSH client for Windows
WinSCP: SCP client for Windows
FileZilla: FTP client for Linux, Windows, Mac
Xming/XQuartz: X11 server for Windows/Mac

Logging in from the CURC Web Portal, OnDemand

- Visit: ondemand.rc.colorado.edu
- Web portal to view, edit, down/upload files, manage and create job templates, and access interactive applications

Accessing different CURC clusters

```
module load slurm/summit   # Summit
module load slurm/blanca   # Blanca
module load slurm/alpine   # Alpine
```

Accessing Compile Nodes/Jobs

```
ssh scompile   # Summit compile node
acompile       # Start Alpine compile job
```

CURC Resources

High Performance Computing (HPC) Clusters

Summit	2nd-Gen CURC Cluster
Blanca	CURC Condo Cluster
Alpine	Next-Gen CURC cluster

Storage

- Cluster Directories

/home (2GB)	Backed up daily
/projects (250GB)	Backed up daily
/scratch/summit (10TB)	No back up, 90 day purge
/scratch/alpine (10TB)	No back up, 90 day purge
/rc_scratch (10TB)	No back up, 90 day purge

- PetaLibrary
  - Paid storage service that supports storage, archive, and sharing of research data
  - Backed up with ZFS Snapshots, snapshots do count against your storage quota

Cloud

- AWS
  - Research Computing Cloud Team managed Amazon Web Servies (AWS) accounts
- CUmulus
  - CURC on-premise cloud service

Monitoring Tools

Slurmtools: A module that contains a collection of functions to assess recent usage statistics.

```
module load slurmtools
```

XDMOD: A web portal for viewing metrics at the system, partition, and user-levels.

- Visit: xdmod.rc.colorado.edu

Curc-Quota: CLI utility to check the storage capacity of your directories on the CURC system.

```
curc-quota
```

Data Transfers

Transferring files between CURC and your system

```
scp source <uname>@login.rc.colorado.edu:destination
scp <uname>@login.rc.colorado.edu:source destination
```

Other transfer options

- Globus: Browser application (**Recommended**)
- Rsync: CLI sync utility
- Sftp: CLI interactive utility
- Rclone: CLI cloud transfer utility

For more details, visit curc.readthedocs.io

SLURM

SLURM is an open-source cluster management and job scheduling system for Linux clusters

SLURM Scheduling

sbatch <file>	Submits a job script <file>
sinteractive	Submits interactive job
squeue -u <user>	Show job queue for <user>.
scancel <jobid>	Deletes the job with <jobid>
scontrol hold <jobid>	Hold job with <jobid>
scontrol release <jobid>	Release job with <jobid>
sinfo	Cluster status
salloc	Request new resource allocation
srun	Launch parallel job step
sacct	Display job accounting information

#SBATCH Directives

--nodes=<count>	Number of nodes
--tasks-per-node=<count>	Processes per node
--ntasks=<count>	Total processes
--cpus-per-task=<count>	CPU cores per process
--nodelist=<nodes>	Preferred nodes
--exclude=<nodes>	Nodes to avoid
--time=<min>	Time limit; either min or dd-hh:mm:ss
--mem=<count>	RAM per node; e.g. 10G
--output=<file>	Standard output; defaults to slurm-jobid.out if omitted
--error=<file>	Write standard error to file
--array=<arrayspec>	Define job array
--gres=gpu:<type>:<count>	Number of GPUs
--mail-user=<email>	Email for job alerts
--mail-type=<type>	Email alert types: BEGIN, END, FAIL, REQUEUE, ALL
--account=<account>	Account to charge
--depend=<state>:<jobid>	Job dependency. state = after,
--job-name=<name>	afterok, afterany, afternotok
--constrain=<attribute>	Job name
--partition=<name>	Request CPU type: e.g., westmere-ex, sandybridge, ivybridge, haswell Submit to partition: e.g., shas, smem, sgpu, amilan-ucb

SLURM (Cont.)

SLURM Environment Variables

SLURM_JOBID	Job ID
SLURM_SUBMIT_DIR	Job submission directory
SLURM_SUBMIT_HOST	Name of host from which job was submitted
SLURM_JOB_NODELIST	Names of nodes allocated to job
SLURM_ARRAY_TASK_ID	Task id within job array
SLURM_JOB_CPUS_PER_NODE	CPU cores per node allocated to job
SLURM_NNODES	Number of nodes allocated to job

Software

Research Computing uses a module system called LMOD to load most software into a user's environment. Most software is not accessible by default and must be loaded in.

LMOD Commands

module avail	Shows all available module
module load <module>	Loads <module> in the environment, specify version with <module>/version**
module spider <string>	Searches for module names matching <string>
module keyword <string>	Searches for modules containing <string> in name or description
module list	List currently loaded modules
module unload <module>	Removes <module> from environment
module purge	Removes all modules from environment
module save <collection>	Saves currently loaded modules to collections
module savelist	Returns all saved module collections
module describe	Get modules in a saved collection

Important notes:

- Modules should be loaded in job scripts, interactive jobs, or on compile nodes only, ***not*** on one of the login nodes
- **It is highly recommended that modules are loaded using their versions, e.g., module load cmake/3.14.1
- A module's dependencies must be loaded before the module can be loaded.