SHARCNET™

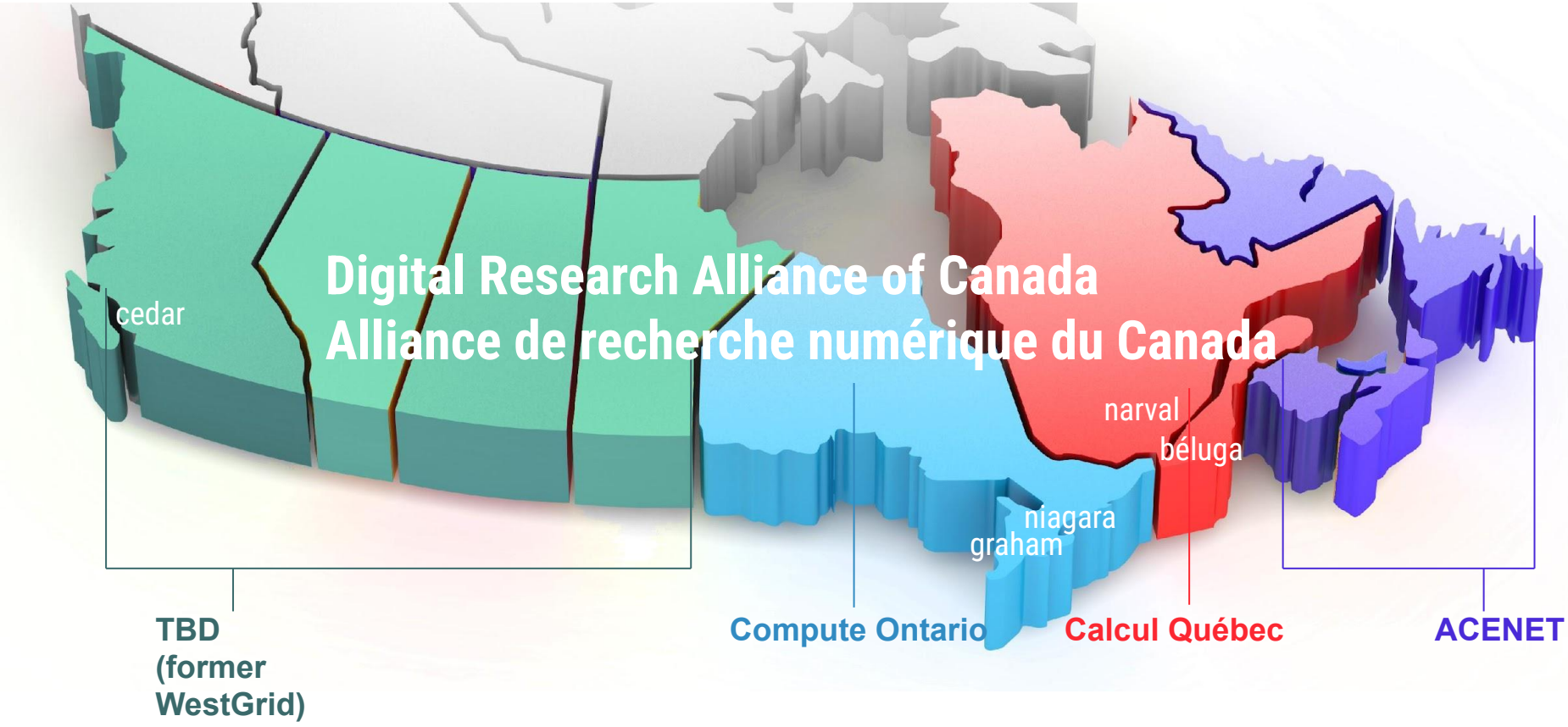
Digital Research Alliance of Canada

New User Seminar

SHARCNET™

Digital Research Alliance of Canada

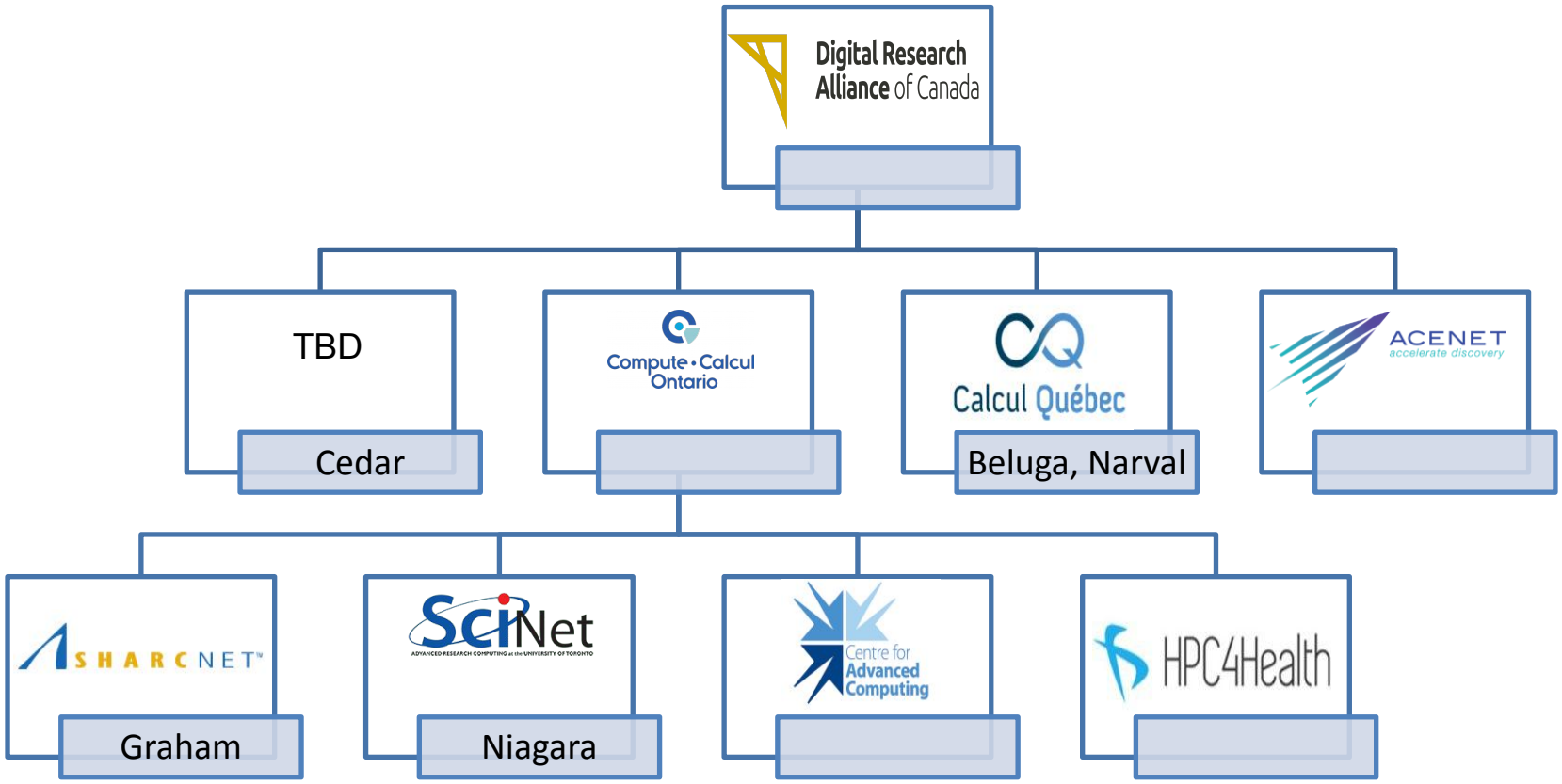
Single account...



One can access all national supercomputers across the country, for free.

SHARCNET™

Alliance's Regional Partners



SHARCNET™

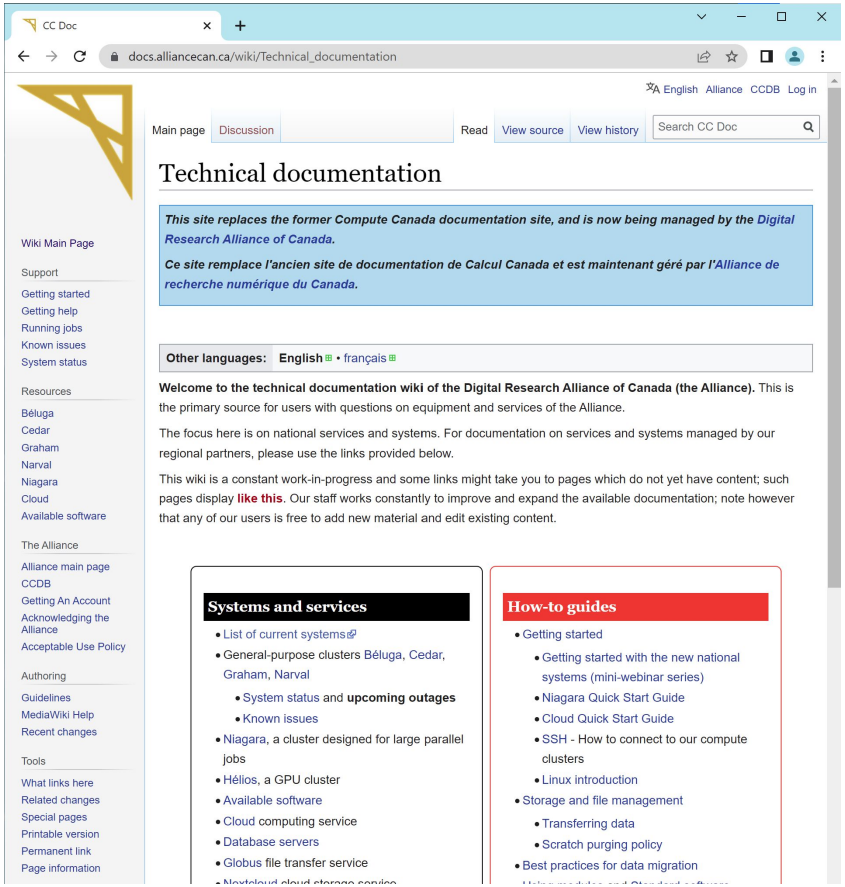
A consortium of 19 Ontario institutions providing advanced computing resources and support...

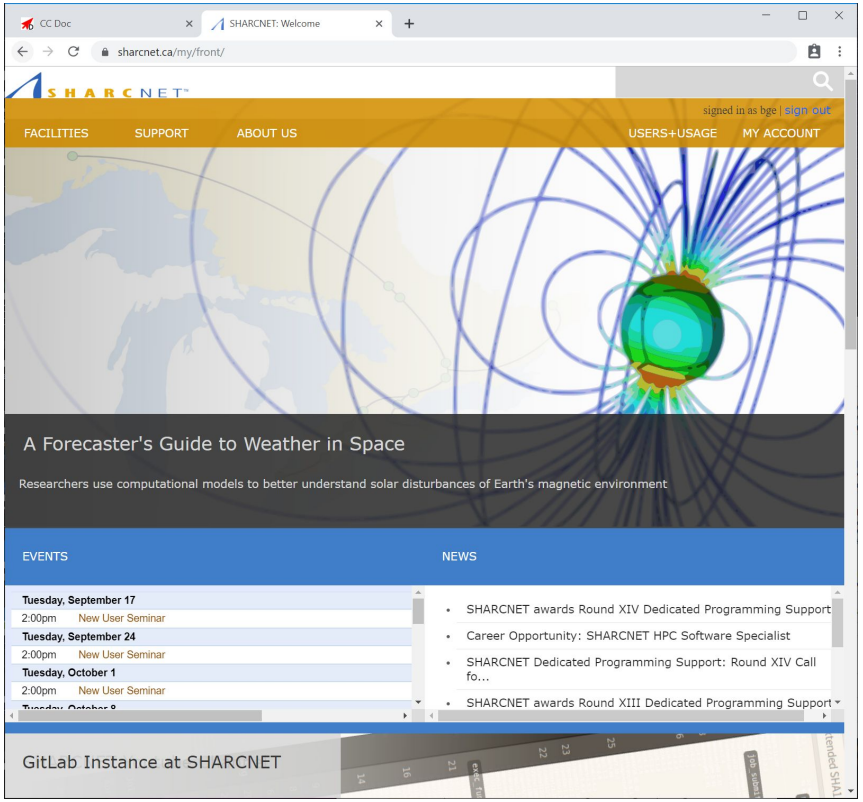
- Member of Alliance and Compute Ontario
- 3,000+ Canadian and international users

Shared
Hierarchical
Academic
Research
Computing
NETwork

- ~50,000 CPU cores
- 500+ GPUs
- 100 Gb/s between nat'l centres

Where to look for information and get help





https://docs.alliancecan.ca/

https://www.sharcnet.ca/
http://youtube.sharcnet.ca

SHARCNET™

Digital Research Alliance of Canada

Where to look for information and get help

https://training.sharcnet.ca

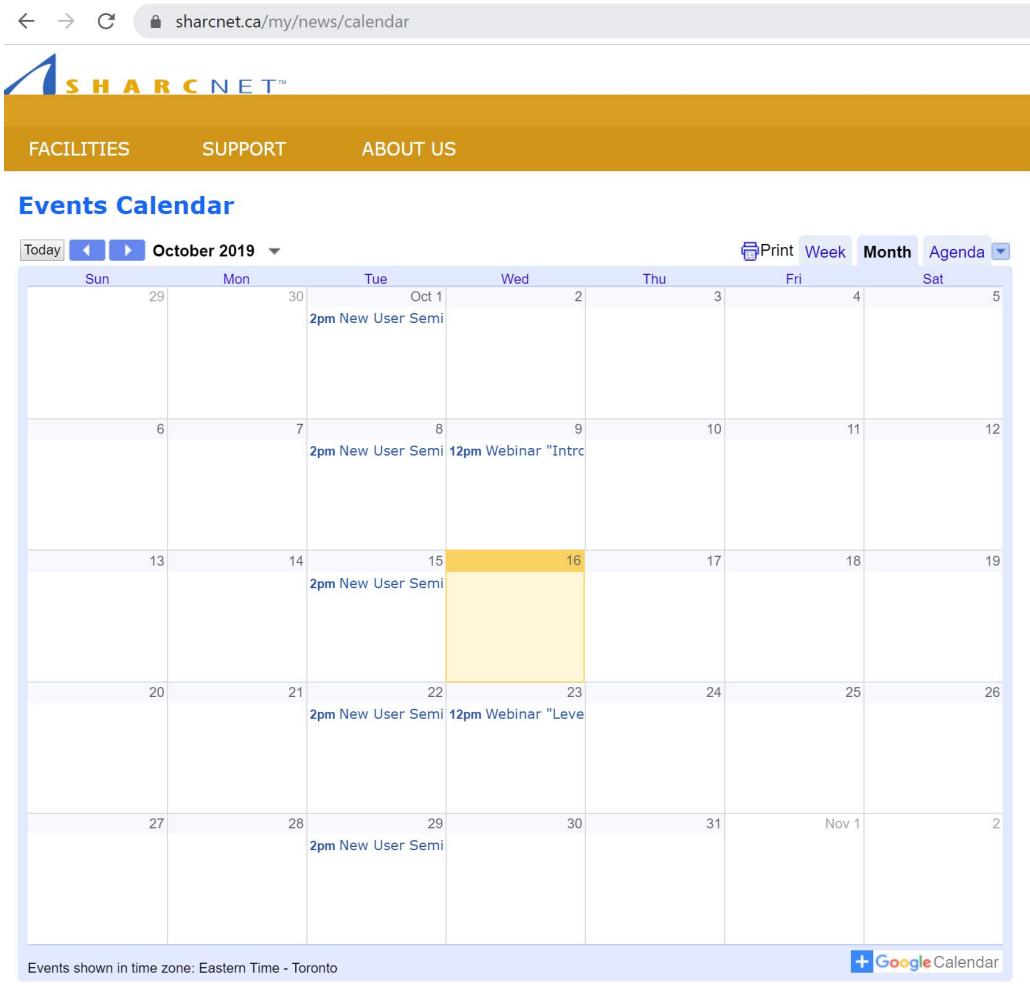
Getting Help (no login required)

Self-Paced Courses (login required)

Where to look for information and get help

Online events

- **NEW**: online self-paced courses
- New user seminar every Tuesday at 2pm.
- Weekly CO colloquia at noon on Wednesdays.
- Recorded seminars on our youtube channel:
  http://youtube.sharcnet.ca
- To subscribe to our Events mailing list, send an email to events+subscribe@sharcnet.ca
- Google for “digital alliance training” to find the calendar for all Alliance events



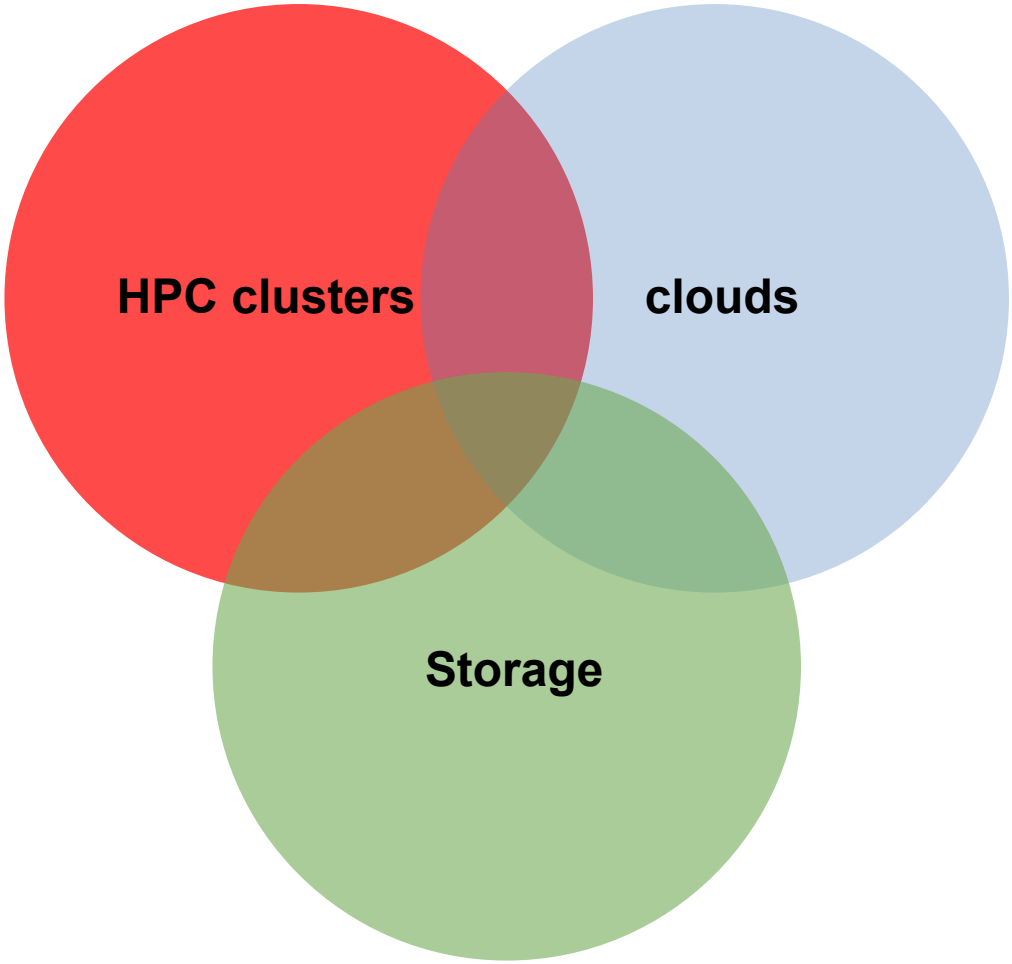
Where to look for information and get help

Interactive help

- Ticketing system via support@tech.alliancecan.ca or help@sharcnet.ca
- E-mail us - check staff contact info on https://www.sharcnet.ca/
- Phone us
- Office visit*

Use of systems
Installation of software
Access to commercial software and site licence
Debugging and optimizing code
Programming
RAC applications
...

The facilities and resources



Supercomputing at SHARCNET and beyond

Clusters across the country

- cedar.alliancecan.ca (**94,528c**)
- graham.alliancecan.ca (**41,548c**)
- niagara.alliancecan.ca (**80,640c**)
- beluga.alliancecan.ca (**39,120c**)
- narval.alliancecan.ca (**80,912c**)

Cloud services

- arbutus.cloud.alliancecan.ca
- cedar.cloud.alliancecan.ca
- graham.cloud.alliancecan.ca
- east.cloud.alliancecan.ca

GPUs

- P100, V100, A100, T4

Storage space

- /home 50G, backed up.
- /project 1T per group, up to 40T by request; backed up.
- /scratch 20T per user, up to 200T by request; old files are removed in 60 days.
- /nearline (tapes)

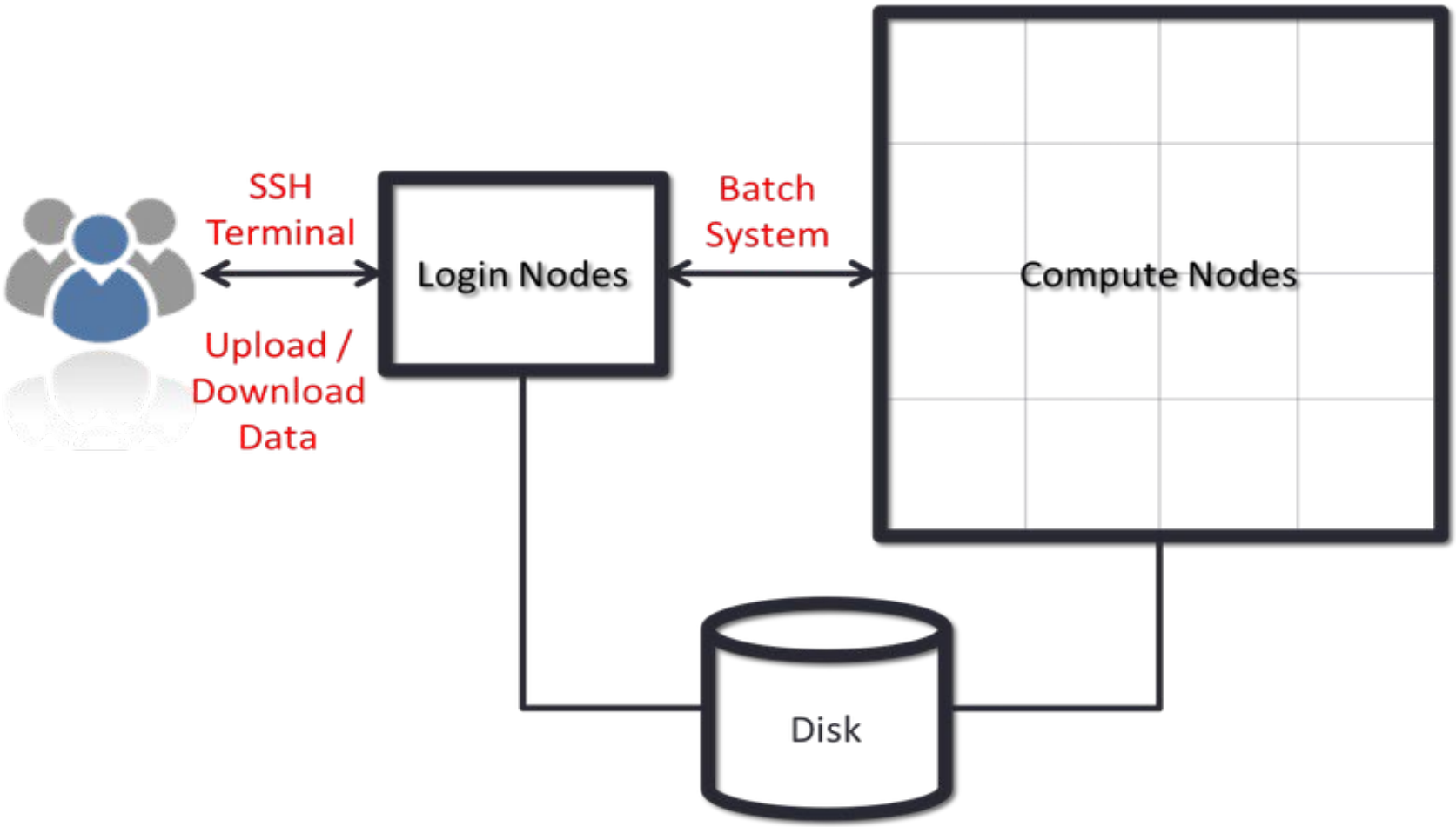
Supercomputing at SHARCNET and beyond

Cluster computing environment

- OS: 64-bit Linux (CentOS)
- Languages supported: Python, R, C/C++, Fortran, Matlab, Java, Julia, CUDA, etc.
- Access to a variety of software packages
- Parallel development support:
  - MPI, Chapel: Distributed memory systems (cross compute nodes) and shared memory system (single node)
  - OpenMP, Pthreads: Multithreading, within a single node
  - CUDA, OpenACC, OpenCL: GPUs and other accelerators on chip
  - C++: Language support for multithreading (since C++-11 standard)
  - Fortran: Language support for parallel programming (since 2003 standard)
  - Julia: Parallel processing constructs, shared and distributed objects
- Data science support:
  - R, Python, Julia, Spark, DASK, etc.
- You must learn how to run programmes in batch via job scheduler slurm

Training courses are available at https://training.sharcnet.ca/

Login nodes vs. compute nodes



Connecting to clusters via SSH

For Windows users, use SSH via one of the following

- PowerShell
- Windows Subsystem for Linux (WSL)
- MobaXterm

For Linux and Mac users

- Install OpenSSH client, if SSH client not already installed
- Open a terminal window, run ssh

To connect to graham cluster, run either of the following

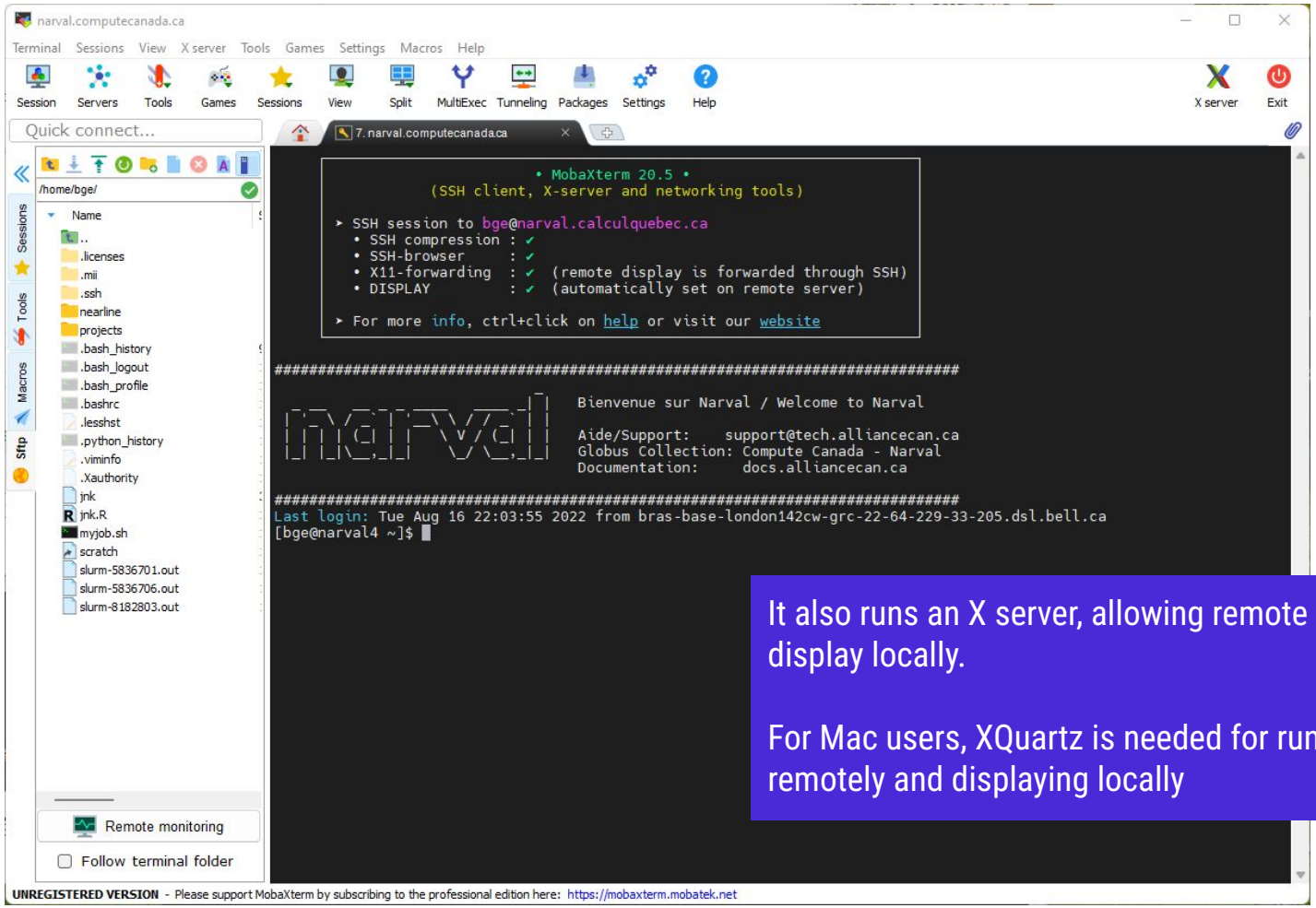
ssh ***username***@graham.alliancecan.ca
ssh graham.alliancecan.ca -l **username**

To transfer files between your computer and remote systems, use **scp** or **sftp** command, or GUI, e.g.MobaXterm, or Globus (web based for large files)

MFA user enrollment is mandatory.

Connecting to clusters via SSH - other options

Windows users may MobaXterm to transfer files



It also runs an X server, allowing remote GUI apps to display locally.

For Mac users, XQuartz is needed for running GUI apps remotely and displaying locally

Multi Factor Authentication (MFA)

Multifactor authentication on all clusters effective April 15, 2024.

Full details: https://docs.alliancecan.ca/wiki/Multifactor_authentication

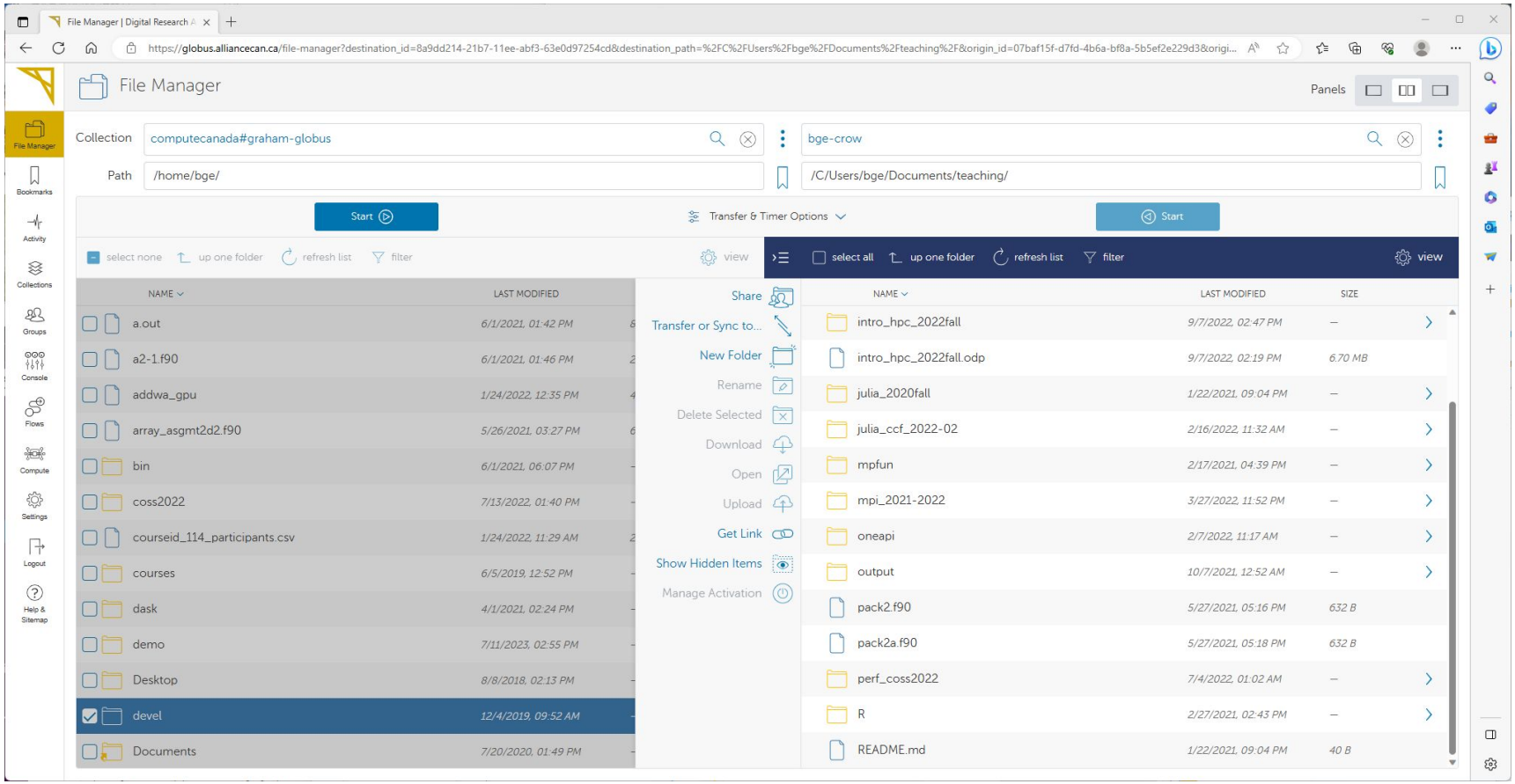
Two common second factors used:

- Duo Mobile authentication application for Android or IOS (free if you already have a phone/tablet)
- Hardware USB key called YubiKey (not free)

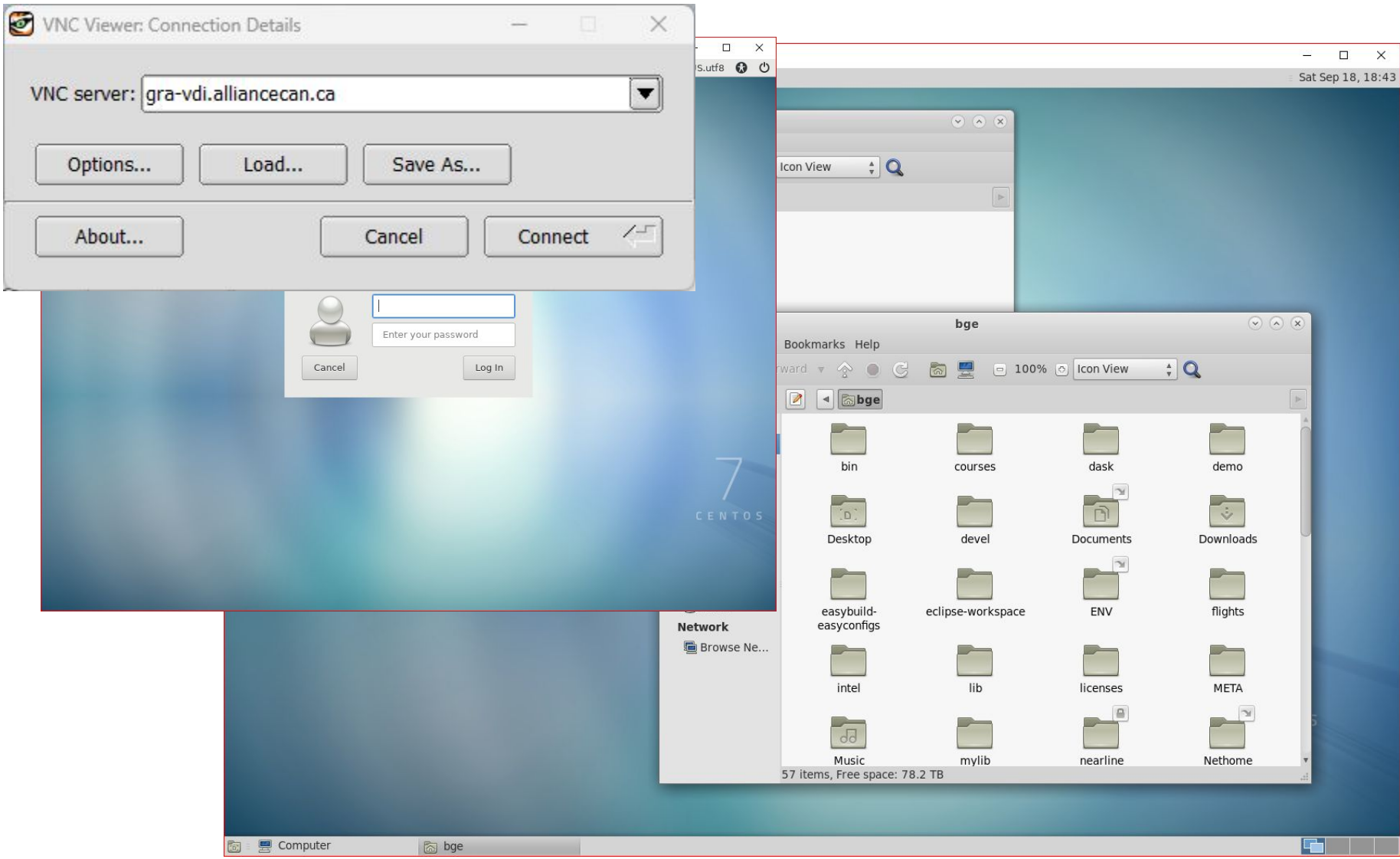
Normally, with MFA you need to use the second factor every time you execute ssh command, or initiate a file copy (scp, sftp, rsync etc.). You can use the **ControlMaster** mechanism to reduce the number of times you have to do this with macOS, Linux, and under Windows if WSL is used. (See the above MFA link for details.)

Transferring large amounts of files using Globus

Check https://docs.alliancecan.ca/wiki/Globus. Go to https://globus.alliancecan.ca/ and follow the instructions



Connecting to a GUI desktop



Accessing and managing files

Everyone has the access to the following file systems

- /**home**: 50G, 0.5m files; backed up regularly
- /**project**: 1T (extendable to 40T) per group, 0.5m files; backed up.
- /**scratch**: 20T per user, 1m files, up to 100T; 2 months of life.
- **nearline**: to store files not currently in use, but may be needed later. 5000 files (approx. 10T) per group. NOT available on compute nodes.

*NB: Please **DO NOT** store everything, remove the files no longer in use to save space.*

Tips for project space

- Do not preserve file permissions when copying files to Project! If you do, you will likely get a “Not enough of disk space” error on Project.
- Specifically, **don’t do** the following when project is the destination, and source is not in project:

  $ cp -p

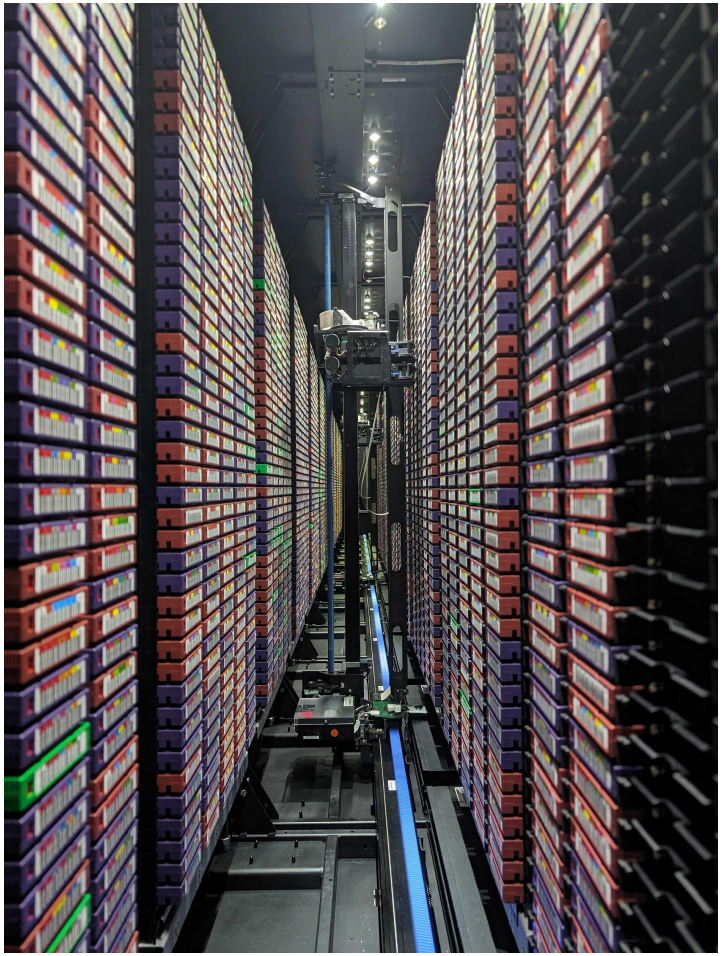
  $ mv

  $ rsync -a

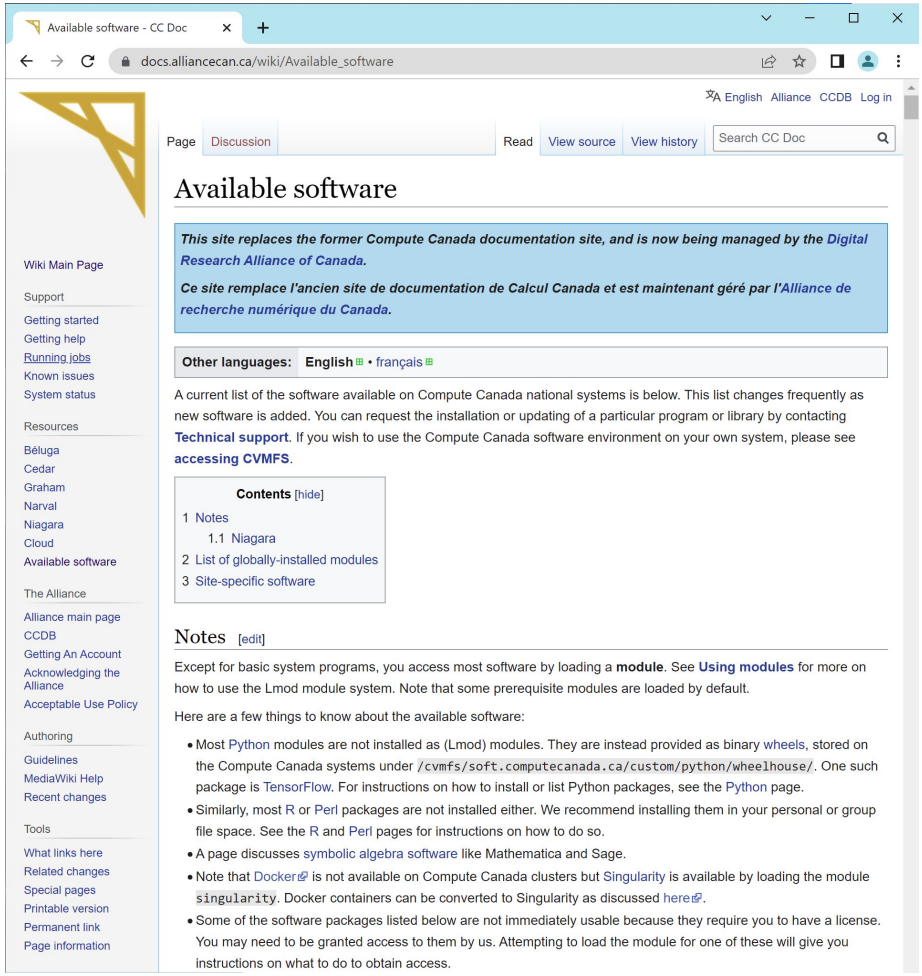
Tips for nearline space

- Quota limit is an “ingest” limit. After it has been consumed, you can add more files up to your file count limit.
- Find your file count by typing:

  find /nearline/YOUR_ACCOUNT/ | **wc** -l
- Don’t edit files in nearline! If it is on tape, your editor might hang.
- File size, some guideline:
  - At least 10G to 20G per file. Ideal file size is 100G to 500G. Up to 2T per file is acceptable.



What software packages are available?



Available software - CC Doc

docs.alliancecan.ca/wiki/Available_software

English Alliance CCDB Log in

Page Discussion Read View source View history Search CC Doc

Available software

This site replaces the former Compute Canada documentation site, and is now being managed by the Digital Research Alliance of Canada.

Ce site remplace l'ancien site de documentation de Calcul Canada et est maintenant géré par l'Alliance de recherche numérique du Canada.

Other languages: **English** • français

A current list of the software available on Compute Canada national systems is below. This list changes frequently as new software is added. You can request the installation or updating of a particular program or library by contacting **Technical support**. If you wish to use the Compute Canada software environment on your own system, please see **accessing CVMFS**.

Contents [hide]

1 Notes
  1.1 Niagara
2 List of globally-installed modules
3 Site-specific software

Notes [edit]

Except for basic system programs, you access most software by loading a **module**. See **Using modules** for more on how to use the Lmod module system. Note that some prerequisite modules are loaded by default.

Here are a few things to know about the available software:

- Most Python modules are not installed as (Lmod) modules. They are instead provided as binary wheels, stored on the Compute Canada systems under `/cvmfs/soft.computecanada.ca/custom/python/wheelhouse/`. One such package is TensorFlow. For instructions on how to install or list Python packages, see the Python page.
- Similarly, most R or Perl packages are not installed either. We recommend installing them in your personal or group file space. See the R and Perl pages for instructions on how to do so.
- A page discusses symbolic algebra software like Mathematica and Sage.
- Note that Docker is not available on Compute Canada clusters but Singularity is available by loading the module `singularity`. Docker containers can be converted to Singularity as discussed here.
- Some of the software packages listed below are not immediately usable because they require you to have a license. You may need to be granted access to them by us. Attempting to load the module for one of these will give you instructions on what to do to obtain access.

Wiki Main Page

Support
Getting started
Getting help
Running jobs
Known issues
System status

Resources
Béluga
Cedar
Graham
Narval
Niagara
Cloud
Available software

The Alliance
Alliance main page
CCDB
Getting An Account
Acknowledging the Alliance
Acceptable Use Policy

Authoring
Guidelines
MediaWiki Help
Recent changes

Tools
What links here
Related changes
Special pages
Printable version
Permanent link
Page information

Check for software while on a cluster

$ **module avail**

$ **module spider** *keyword*

Running jobs using a slurm script - *myjob*.sh

Submitting a serial job

With a slurm submit script:

```
#!/bin/bash
#SBATCH --time=00-01:00:00 # DD-HH:MM
#SBATCH --account=def-user
module load python/3.6
python simple_job.py 7 output
```

To see what account groups you have access to, use command **sshare -U** or **salloc** by itself

sbatch *myjob*.sh

Submitting a series of jobs

With a slurm submit script:

```
#!/bin/bash
#SBATCH --time=01:00
#SBATCH --account=def-user
#SBATCH --array=1-200

python simple_job.py $SLURM_ARRAY_TASK_ID output
```

META package

https://docs.alliancecan.ca/wiki/META:_A_package_for_job_farming

(Google for “meta digital alliance”)

A more universal and convenient way to manage a large number of similar jobs.

Check https://docs.alliancecan.ca/ for Running jobs.

Running jobs using a slurm script - *myjob*.sh

Submitting a threaded job

With a slurm submit script:

```
#!/bin/bash
#SBATCH --account=def-user
#SBATCH --time=0-03:00
#SBATCH --cpus-per-task=32
#SBATCH --ntasks=1
#SBATCH --mem=20G

export \
 OMP_NUM_THREADS=$SLURM_CPUS_PER_TASK

./myprog.exe
```

Submitting a parallel job

With a slurm submit script:

```
#!/bin/bash
#SBATCH --account=def-user
#SBATCH --time=5-00:00
#SBATCH --ntasks=100
#SBATCH --mem-per-cpu=4G

srun ./mympiprog.exe
```

sbatch *myjob*.sh

Check https://docs.alliancecan.ca/ for Running jobs.

Running jobs using a slurm script - *myjob*.sh

Submitting a GPU job

With a slurm submit script:

```
#!/bin/bash
#SBATCH --account=def-user
#SBATCH --time=0-03:00
#SBATCH --gpus-per-node=v100:2
#SBATCH --mem=20G

./myprog.exe
```

[*type*:]*number*

Submitting hybrid MPI-threaded job

With a slurm submit script:

```
#!/bin/bash
#SBATCH --account=def-user
#SBATCH --time=0-03:00
#SBATCH –ntasks=16
#SBATCB --cpus-per-task=4
#SBATCH --mem=20G

export
OMP_NUM_THREADS=$SLURM_CPUS_PER_TASK
srun –cpus-per-task=$SLURM_CPUS_PER_TASK
./myprog
```

☜ MPI ranks

☜ threads

sbatch *myjob*.sh

Use options –ntasks, --nodes, --ntasks-per-node and –cpus-per-task to configure how you want to distribute MPI ranks.

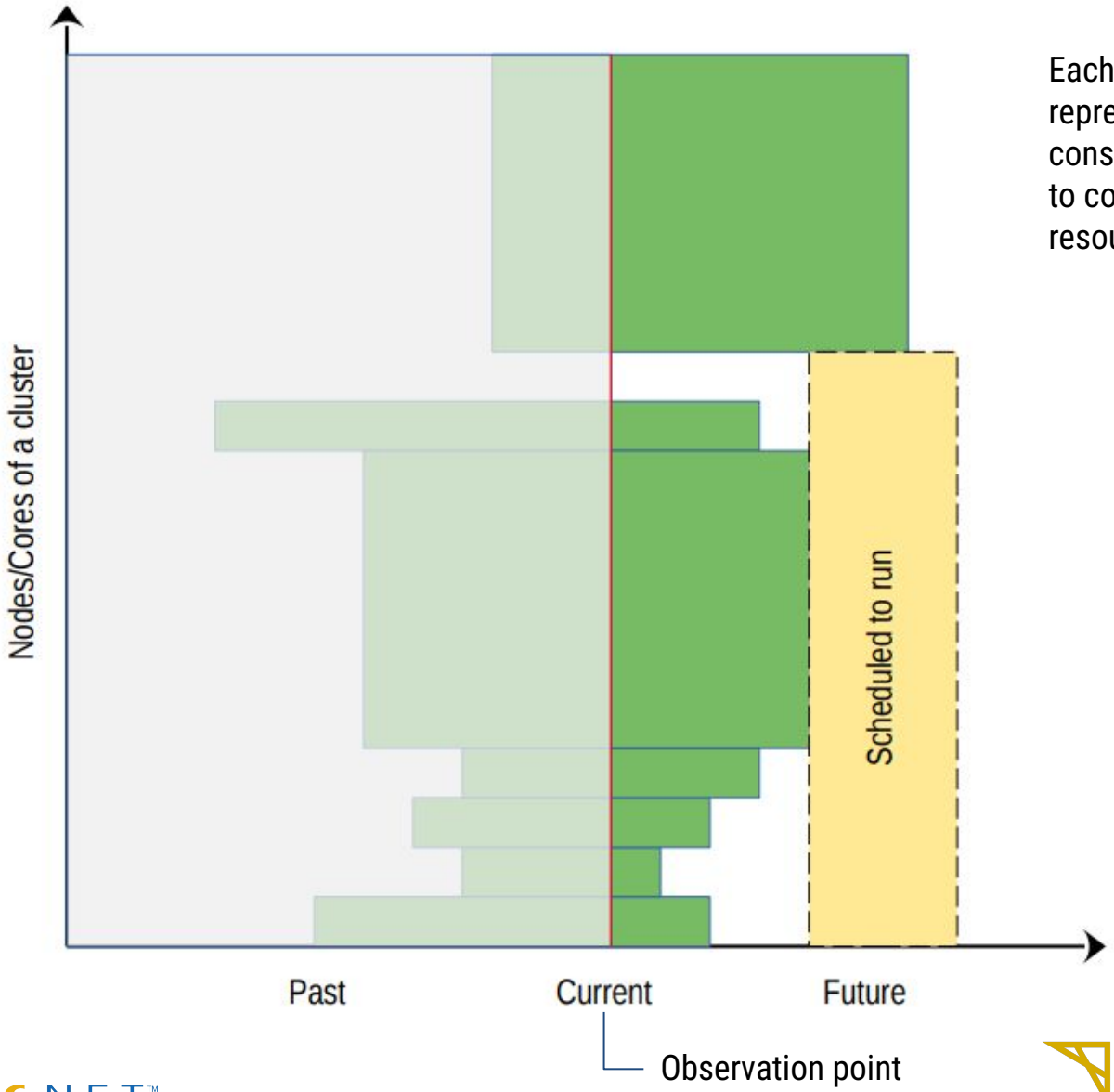
Check https://docs.alliancecan.ca/ for Running jobs.

Viewing your jobs

Commonly used slurm commands

- squeue -u *username* *To get the status of submitted jobs*
- sacct *To get the info of past jobs*
- scancel *jobID* *To cancel job jobID*
- seff *jobID* *To output a finished job report*
- salloc *To run jobs interactively*
- sbatch *To submit a (batch) job to run*
- sshare *To show group fair share level*

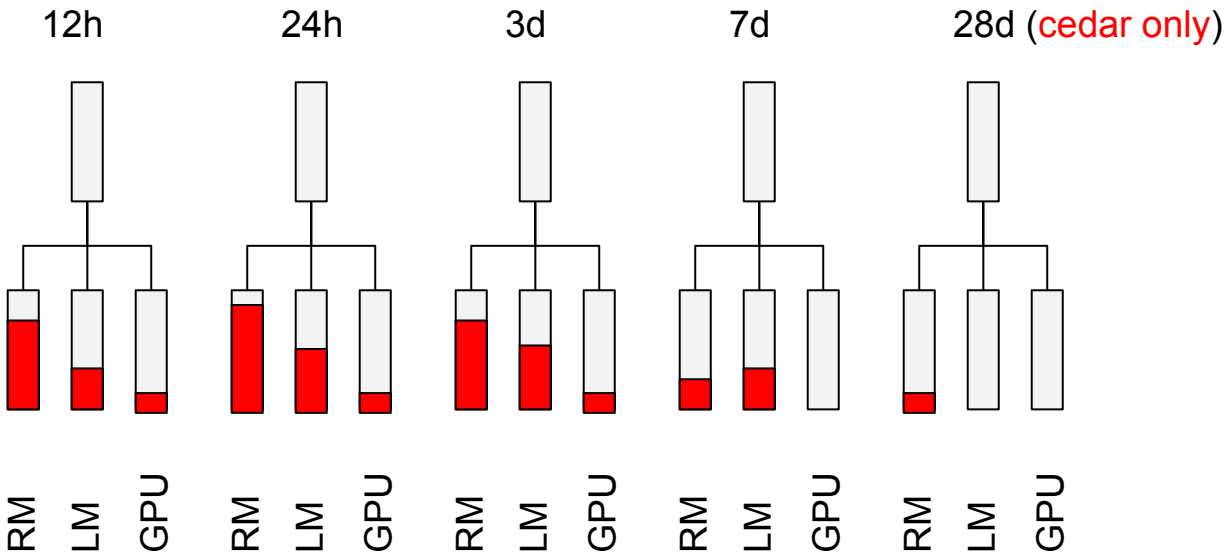
How scheduler works - *jobs may wait hours to days!*



Each block represents a job consuming (green) or to consume (yellow) resources

Why my jobs don't start

On entry, each of your jobs is "placed in one of the partitions" by its attributes (cores, memory, runtime, etc.) you specified



You may use command **partition-stats** to see the system status.

Further information

Please go to SHARCNET training website at

https://training.sharcnet.ca/

What can be done about wait times

- Requesting more resources (runtime, CPU cores, memory) than what the job process requires can result in a longer queue times.

  Tip: request only what the job needs, with a bit of leeway for time and memory.

- The recent usage of an account is calculated independently on each of the Alliance general purpose systems (Graham, Beluga, and Cedar) and the availability of the resources varies across systems.

  Tip: use multiple systems when appropriate.

- More resources are available to full-node jobs. If your job can efficiently use multiples of 32 cpu cores (graham) it gains access to a larger set of nodes if it is submitted as a full-node job.

  Tip: use --nodes=N and --ntasks-per-node=32 sbatch arguments for full-node jobs.

- Less than 20% of all resources are available via default accounts.

  Tip: If a project needs more than the default level usage, a larger target share of the system can be obtained through the annual Resources Allocation Competition (RAC)

Common mistakes to avoid

- Do not run significant programs on login nodes, nor run programs directly on compute nodes.
- Do not specify a job run time blindly (say, 7 days), or more memory than needed for your program
- Try not to create millions of tiny files, or large amounts (> GB) of uncompressed (eg. ASCII) output
- Do not let your jobs access (read/write) files frequently (more than 10 files per second from all of your running jobs).

Q&A