

# GACRC Sapelo2 Cluster New User Training

---

Georgia Advanced Computing Resource Center (GACRC)

Enterprise Information Technology Services(EITS)

The University of Georgia

# Outline

---

- GACRC
- Sapelo2 Cluster
  - Cluster Diagram and Overview
  - Five Directories
  - Five Computational Partitions
  - Software Environment
- Batch Job Submission Workflow
- Useful Commands: `sq --me`, `sacct-gacrc -X`, `interact`
- GACRC Wiki and User Support
- Appendices

# GACRC

- A high-performance-computing (HPC) center at the UGA
- Provide to the UGA research and education community an advanced computing environment:
  - HPC computing and networking infrastructure located at the Boyd Data Center
  - Comprehensive collection of scientific, engineering and business applications
  - Consulting and training services

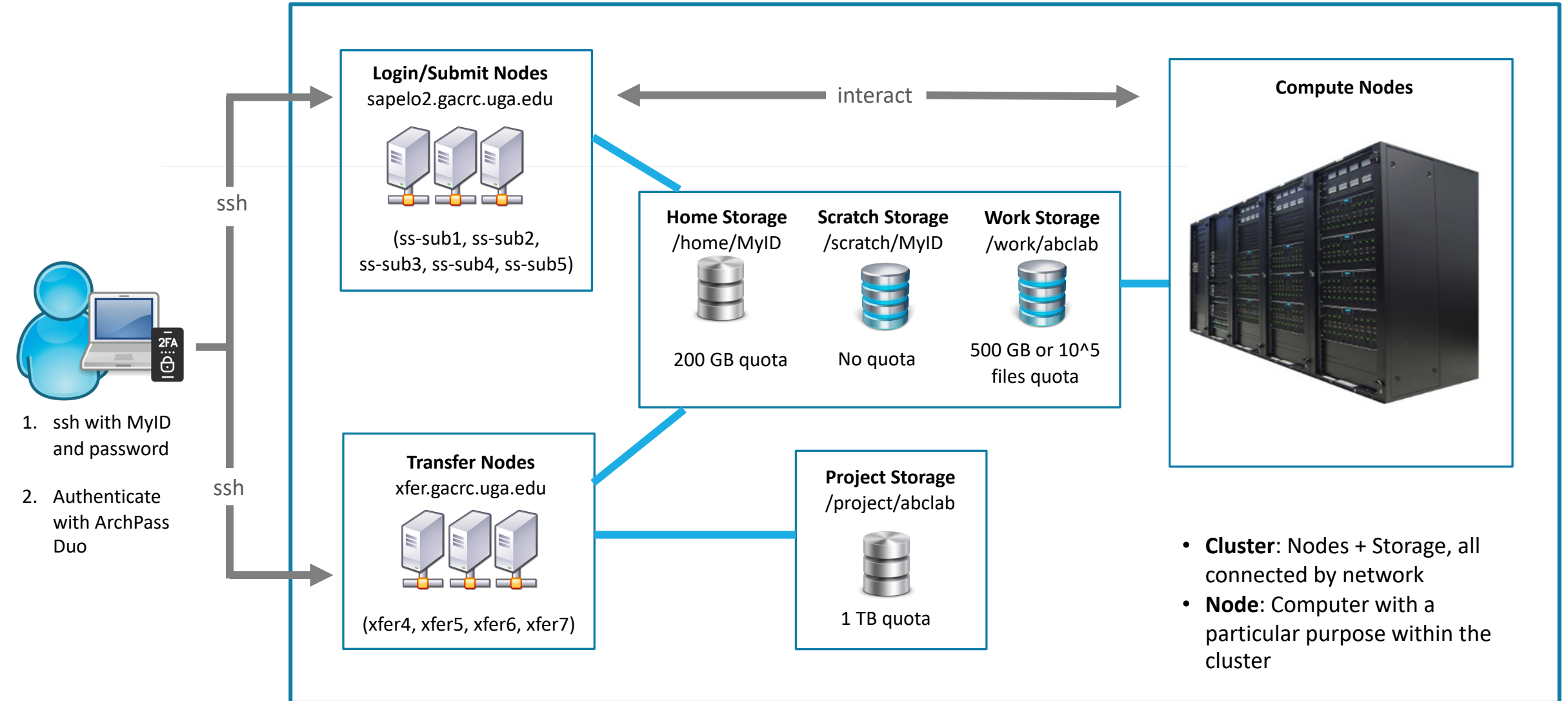
Wiki: <http://wiki.gacrc.uga.edu>

Help and Support: <http://help.gacrc.uga.edu>

Web Site: <http://gacrc.uga.edu>

Kaltura Channel: <https://kaltura.uga.edu/channel/GACRC/176125031>

# Sapelo2 Cluster



**Note:** You need to connect to the **UGA network using VPN** when accessing from outside of the **UGA main campus**.

UGA VPN: [https://eits.uga.edu/access\\_and\\_security/infosec/tools/vpn/](https://eits.uga.edu/access_and_security/infosec/tools/vpn/)



# Five Directories [https://wiki.gacrc.uga.edu/wiki/Disk\\_Storage](https://wiki.gacrc.uga.edu/wiki/Disk_Storage)

Directory	Name	Quota	Accessible from	Intended Use	Backed-up	Important Notes
/home/MyID	Home	200GB	Login Transfer Compute	Static data, e.g. 1. Scripts, source codes 2. Local software	Yes	Not for storing data of your jobs!
/scratch/MyID	Scratch	No Limit	Login Transfer Compute	Temporary files needed for currently running jobs	No	Clean up when your job finishes! Subject to "30-day purge" policy
/work/abclab	Work	500GB 10 <sup>5</sup> files	Login Transfer Compute	Input files needed for repeated jobs	No	Clean up when your job finishes! Group sharing is possible
/project/abclab	Project	1TB (initial)	Transfer	Temporary data parking	Yes	Group sharing is possible
/lscratch	Local Scratch	200GB - 800GB	Compute	Jobs with heavy disk I/O operations	No	Clean up when job exits from node!

# Scratch File System 30-Day Purge Policy

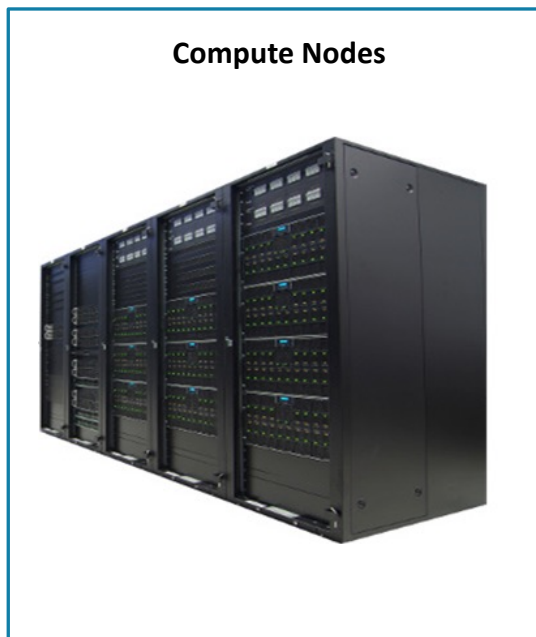
[https://wiki.gacrc.uga.edu/wiki/Disk\\_Storage#Scratch\\_file\\_system](https://wiki.gacrc.uga.edu/wiki/Disk_Storage#Scratch_file_system)

Any file that is not accessed or modified by a compute job in a time period **no longer than 30 days** will be automatically deleted off the /scratch file system.

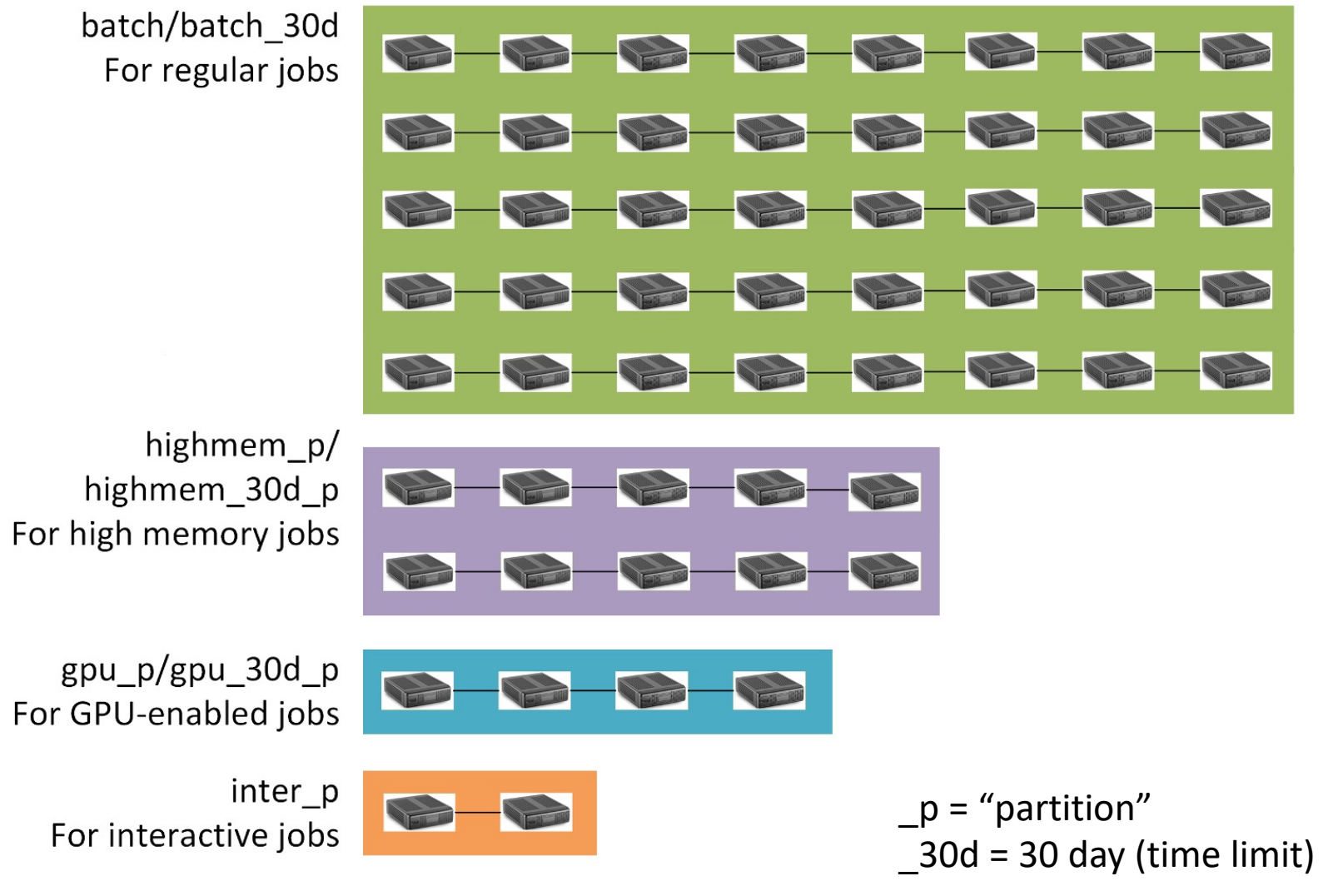
Measures circumventing this policy will be monitored and actively discouraged.

- You are suggested to copy files from /scratch to **/project** or **outside of GACRC**
- You should first move all unnecessary files and folders to **/scratch/trash/\$USER**
- The fastest way to save old files is to copy them to /project area, using the **fpsync** utility on xfer.gacrc.uga.edu
- When you archive data using **tar** on /scratch, please **do not use z option** (compression option). After you archive data with tar, you can use gzip to compress it.

# Computational Partitions



Compute nodes are divided into groups called **partitions**. A **partition** is a collection of compute nodes for a particular computing need.



# Computational Partitions [https://wiki.gacrc.uga.edu/wiki/Job\\_Submission\\_partitions\\_on\\_Sapelo2](https://wiki.gacrc.uga.edu/wiki/Job_Submission_partitions_on_Sapelo2) <https://wiki.gacrc.uga.edu/wiki/Systems#Sapelo2>

Type	Partition	Time limit	Max jobs Running	Max jobs Submit	Notes
Regular	batch	7 days	250	10,000	Regular nodes
	highmem_p		15	100	For running high memory jobs
	hugemem_p		4	4	For running huge memory jobs
	gpu_p		18	20	For running GPU-enabled jobs
Long-term	batch_30d	30 days	1	2	30-day partition counterparts
	highmem_30d_p		4	4	
	hugemem_30d_p		2	2	
	gpu_30d_p				
Interactive	inter_p	2 days	3	20	Regular nodes, for interactive jobs.
Buy-in	name_p	variable			Partitions that target different groups' buy-in nodes. The <b>name</b> string is specific to each group.

Partition	Total Nodes	Max Mem(GB) /Single-node job	Cores /Node	Processor Type	GPU Cards /Node
batch batch_30d	119	500	128	AMD EPYC Milan	N/A
	4	250	64		
	2	120	64		
	123		64	AMD EPYC Rome	
	64		32	AMD EPYC Naples	
	42	180	32	Intel Xeon Skylake	
highmem_p highmem_30d_p	18	500	32	AMD EPYC Naples	N/A
	4	990	64		
	5		32	AMD EPYC Milan	
	2		128		
	4		28	Intel Xeon Broadwell	
hugemem_p, hugemem_30d_p	2	2000	32	AMD EPYC Rome	N/A
gpu_p gpu_30d_p	4	180	32	Intel Xeon Skylake	1 NVIDIA P100
	2	120	16	Intel Xeon	8 NVIDIA K40m
	5	1000	64	AMD PEYC Milan	4 NVIDIA A100
Total			1000	1000	1000

# Software Environment

<https://wiki.gacrc.uga.edu/wiki/Software>

- Approximately 900 software modules are installed (as of March 2021)
- Most modules are compiled EasyBuild (EB) toolchains GCC-8.3.0 or foss-2019b.
- Name format: **Name/Version-Toolchain**, e.g., **Python/3.8.2-GCCcore-8.3.0** (case-sensitive)
- Module commands:
  - **ml spider *pattern*** : Search module names matching a *pattern*
  - **ml *moduleName*** : Load a module into your working environment
    - **DO NOT LOAD/USE MODULES ON THE LOGIN/SUBMIT NODES!** (ss-sub1, ss-sub2, ss-sub3, etc...)
  - **ml av** : List all available software modules installed on cluster
  - **ml** : List modules currently loaded
  - **ml -*moduleName*** : Remove a module from working environment
  - **ml purge** : Remove all modules from working environment

# Important Tip using Software

[https://wiki.gacrc.uga.edu/wiki/Available\\_Toolchains\\_and\\_Toolchain\\_Compatibility](https://wiki.gacrc.uga.edu/wiki/Available_Toolchains_and_Toolchain_Compatibility)

- When you load more than one software modules, **toolchain compatibility** is the most important thing you need to pay attention to
- If you load more than one module and some toolchains are incompatible, your job will end up with failing dependencies or Lmod errors, such as:

Lmod has detected the following error:

These module(s) exist but cannot be loaded as requested



ml Python/3.7.4-GCCcore-8.3.0  
ml Perl/5.28.0-GCCcore-7.3.0



ml Python/3.7.4-GCCcore-8.3.0  
ml Perl/5.30.0-GCCcore-8.3.0



ml Beast/2.6.3-foss-2019b  
ml Perl/5.28.0-GCCcore-7.3.0



ml Beast/2.6.3-foss-2019b  
ml Perl/5.30.0-GCCcore-8.3.0

# Job Submission Workflow

[https://wiki.gacrc.uga.edu/wiki/Running\\_Jobs\\_on\\_Sapelo2](https://wiki.gacrc.uga.edu/wiki/Running_Jobs_on_Sapelo2)

---

1. Log on to Login node using MyID and password, and two-factor authentication with Archpass Duo:  
`ssh MyID@sapelo2.gacrc.uga.edu`
2. On Login node, change directory to your scratch space: `cd /scratch/MyID`
3. Create a working subdirectory for a job : `mkdir workDir`
4. Change directory to workDir : `cd workDir`
5. Transfer data from local computer to workDir : use **Globus** to transfer data to the cluster  
Transfer data on cluster to workDir : use **Globus** or log on to Transfer node and then use `cp` or `mv`
6. Make a job submission script in workDir : `nano sub.sh`
7. Submit a job from workDir : `sbatch sub.sh`
8. Check job status : `squeue --me` or Cancel a job : `scancel jobID`



## Step 1: Log on to Login node - Mac/Linux using ssh

<https://wiki.gacrc.uga.edu/wiki/Connecting>

---

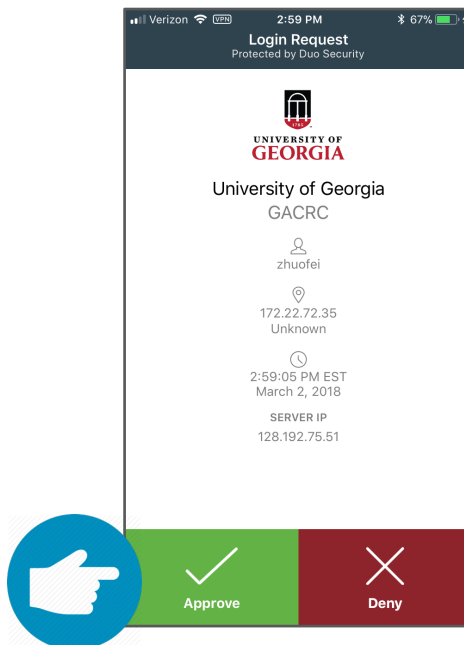
1. Open **Terminal** utility
2. Type command line: `ssh MyID@sapelo2.gacrc.uga.edu`
3. You will be prompted for your **MyID password**
4. Sapelo2 access requires ID verification using two-factor authentication with Archpass

Duo. If you are not enrolled in Archpass Duo, please refer to

[https://eits.uga.edu/access\\_and\\_security/infosec/tools/archpass\\_duo/](https://eits.uga.edu/access_and_security/infosec/tools/archpass_duo/) on how to enroll

## Step 1 (Cont.): Mac/Linux using ssh

Use Terminal utility on  
Mac or Linux!



```
[zhuofei@localhost ~]$  
[zhuofei@localhost ~]$ ssh zhuofei@sapelo2.gacrc.uga.edu ← Log on  
Password: ← Input MyID password!
```

.....  
Enter a passcode or select one of the following options:

1. Duo Push to XXX-XXX-5758
2. Phone call to XXX-XXX-5758
3. Phone call to XXX-XXX-1925
4. SMS passcodes to XXX-XXX-5758

Passcode or option (1-5): 1 ← Select Duo authentication option!

Success. Logging you in...

Last login: Tue Sep 15 11:22:42 2020 from 128.192.75.65

```
zhuofei@ss-sub1 ~$ ← I am on login node ss-sub1!
```

## Step 1 (Cont.): Windows using PuTTY

---

1. Download and install PuTTY: <https://www.putty.org/>

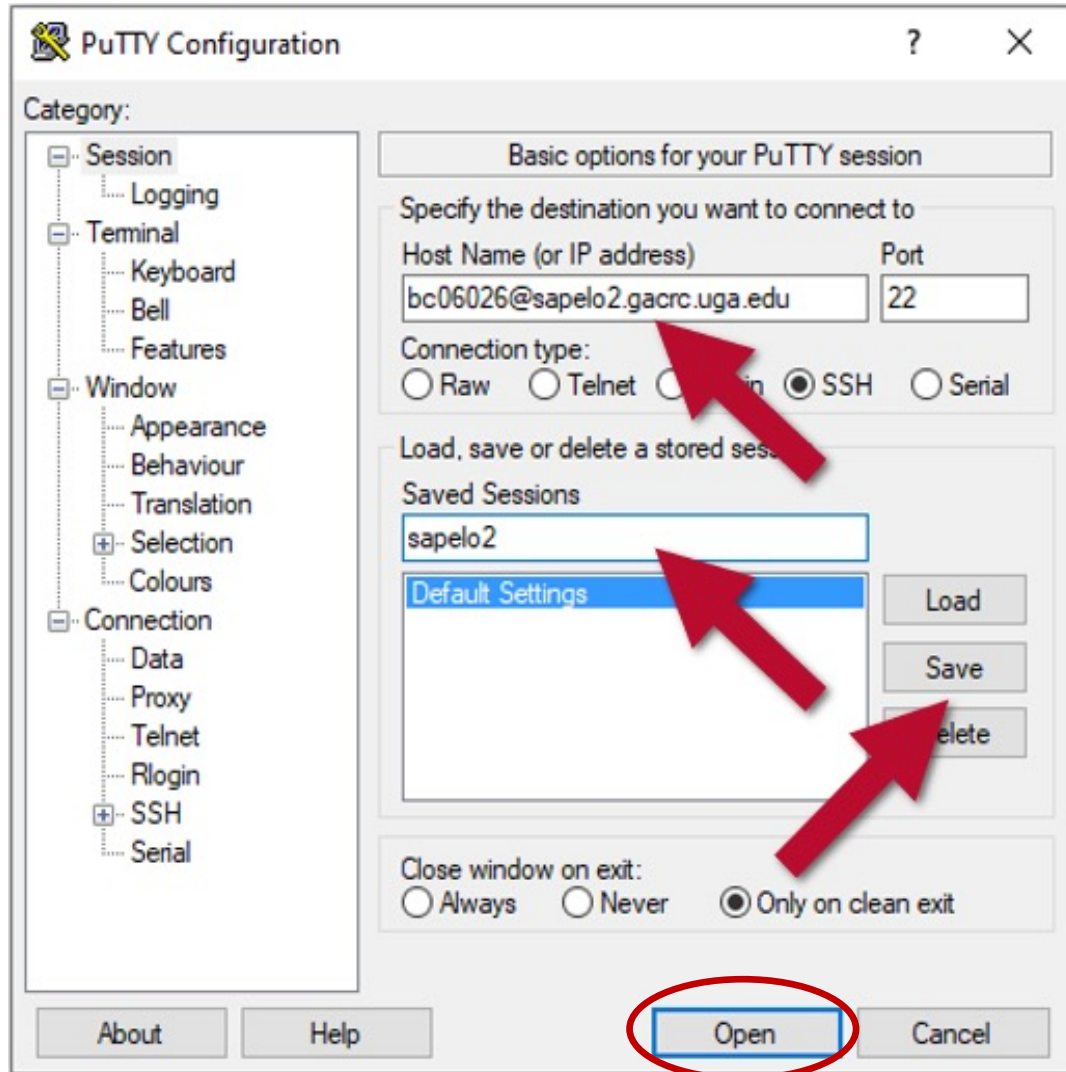
2. Detailed downloading and installation instructions:

[https://wiki.gacrc.uga.edu/wiki/How\\_to\\_Install\\_and\\_Configure\\_PuTTY](https://wiki.gacrc.uga.edu/wiki/How_to_Install_and_Configure_PuTTY)

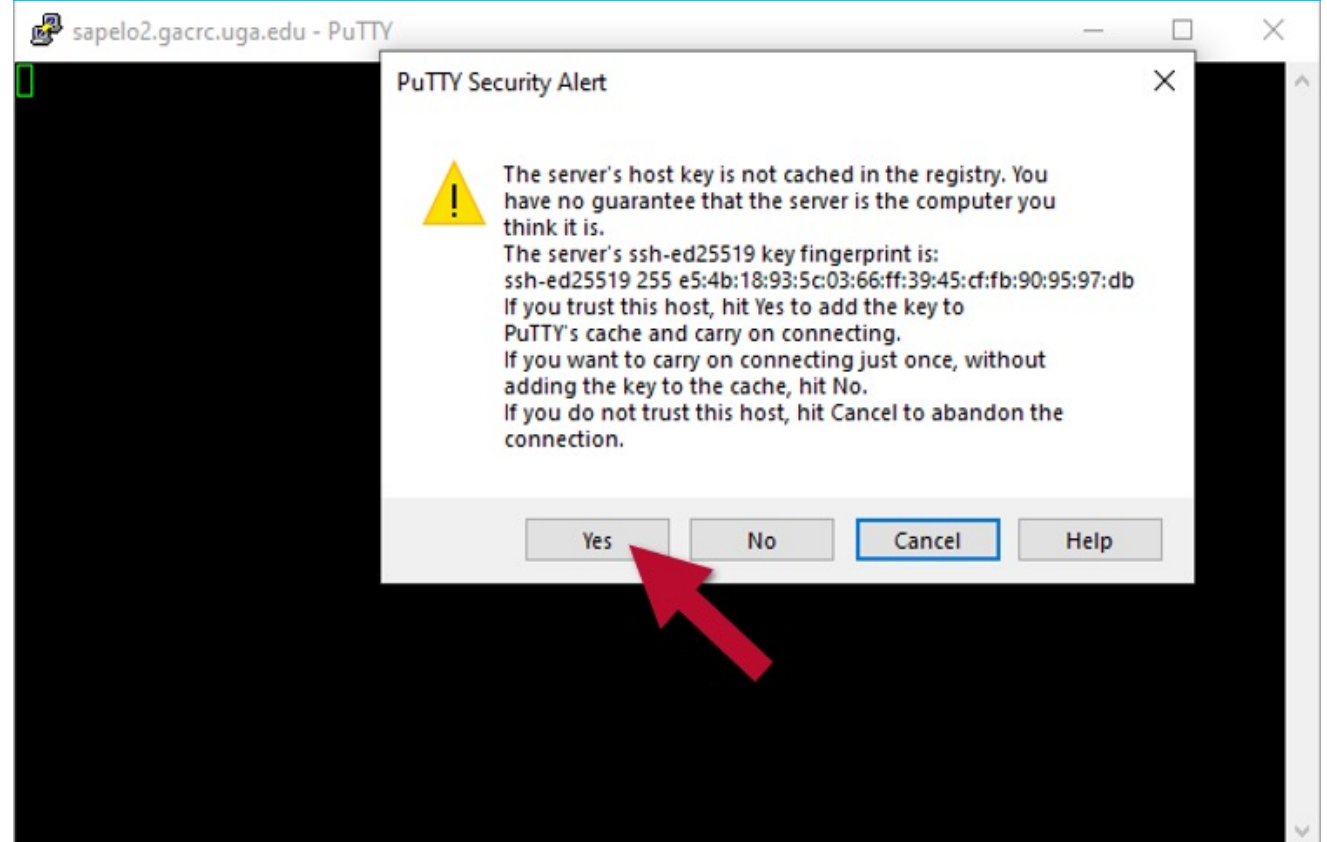
3. Detailed configuring and usage instructions:

[https://wiki.gacrc.uga.edu/wiki/How\\_to\\_Install\\_and\\_Configure\\_PuTTY#Configuring\\_PuTTY](https://wiki.gacrc.uga.edu/wiki/How_to_Install_and_Configure_PuTTY#Configuring_PuTTY)

# Step 1 (Cont.): Windows using PuTTY

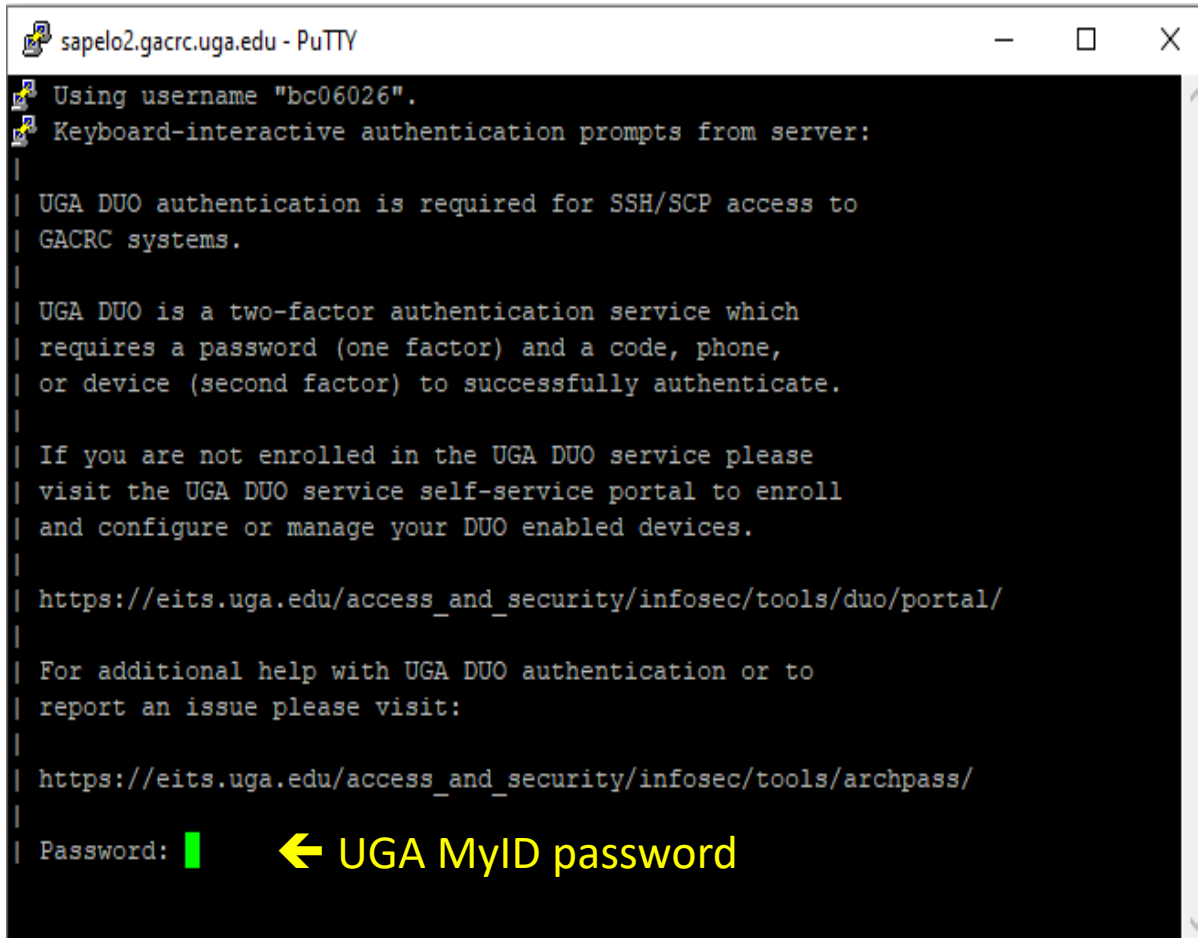


The first time you connect to a login node, PuTTY will give you this security alert window. Please click "Yes" or "Accept"



# Step 1 (Cont.): Windows using PuTTY

Next you will enter your UGA MyID password and initiate DUO authentication procedure:



```
sapelo2.gacrc.uga.edu - PuTTY
Using username "bc06026".
Keyboard-interactive authentication prompts from server:

UGA DUO authentication is required for SSH/SCP access to
GACRC systems.

UGA DUO is a two-factor authentication service which
requires a password (one factor) and a code, phone,
or device (second factor) to successfully authenticate.

If you are not enrolled in the UGA DUO service please
visit the UGA DUO service self-service portal to enroll
and configure or manage your DUO enabled devices.

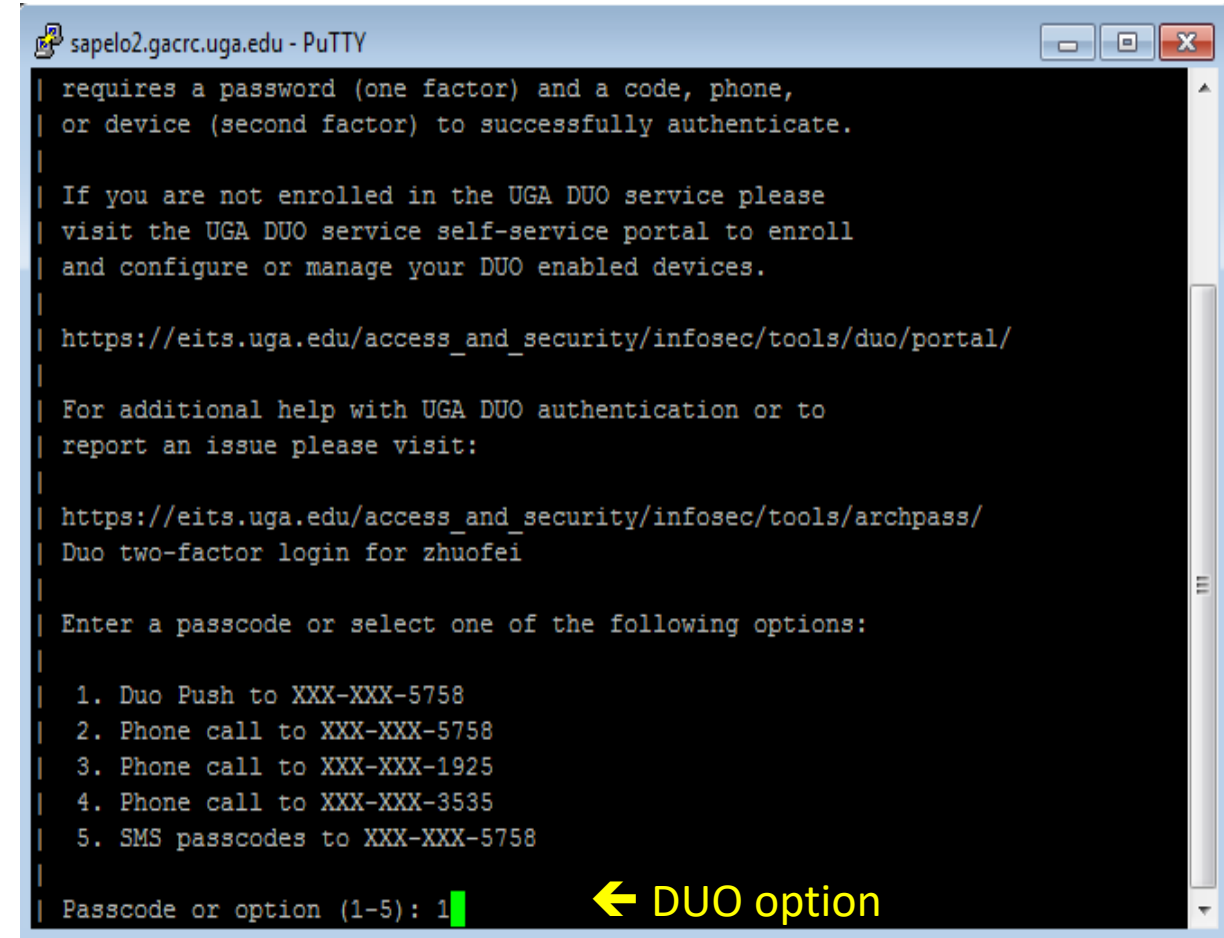
https://eits.uga.edu/access_and_security/infosec/tools/duo/portal/

For additional help with UGA DUO authentication or to
report an issue please visit:

https://eits.uga.edu/access_and_security/infosec/tools/archpass/

Password: █
```

← UGA MyID password



```
sapelo2.gacrc.uga.edu - PuTTY
requires a password (one factor) and a code, phone,
or device (second factor) to successfully authenticate.

If you are not enrolled in the UGA DUO service please
visit the UGA DUO service self-service portal to enroll
and configure or manage your DUO enabled devices.

https://eits.uga.edu/access_and_security/infosec/tools/duo/portal/

For additional help with UGA DUO authentication or to
report an issue please visit:

https://eits.uga.edu/access_and_security/infosec/tools/archpass/
Duo two-factor login for zhuofei

Enter a passcode or select one of the following options:

1. Duo Push to XXX-XXX-5758
2. Phone call to XXX-XXX-5758
3. Phone call to XXX-XXX-1925
4. Phone call to XXX-XXX-3535
5. SMS passcodes to XXX-XXX-5758

Passcode or option (1-5): 1 █
```

← DUO option

## Step 2: On Login node change directory to scratch

---

- Once you logged on, your current directory will be your home directory

```
zhuofei@ss-sub1 ~$ pwd  
/home/zhuofei
```

← this is my home directory!

- Use `cd` command to change your current directory to /scratch/MyID

```
zhuofei@ss-sub1 ~$ cd /scratch/zhuofei/  
zhuofei@ss-sub1 zhuofei$ pwd  
/scratch/zhuofei
```

← this is my scratch space!

- Use `ls` command to take a look in /scratch/MyID

```
zhuofei@ss-sub1 zhuofei$ ls  
user_test
```

## Step 3 - 4: Create and cd to a working subdirectory

---

- Use `mkdir` command to make a subdirectory in `/scratch/MyID`

```
zhuofei@ss-sub1 zhuofei$ mkdir workDir
zhuofei@ss-sub1 zhuofei$ ls
user_test  workDir
```

- Use `cd` command to change your current directory to `/scratch/MyID/workDir`

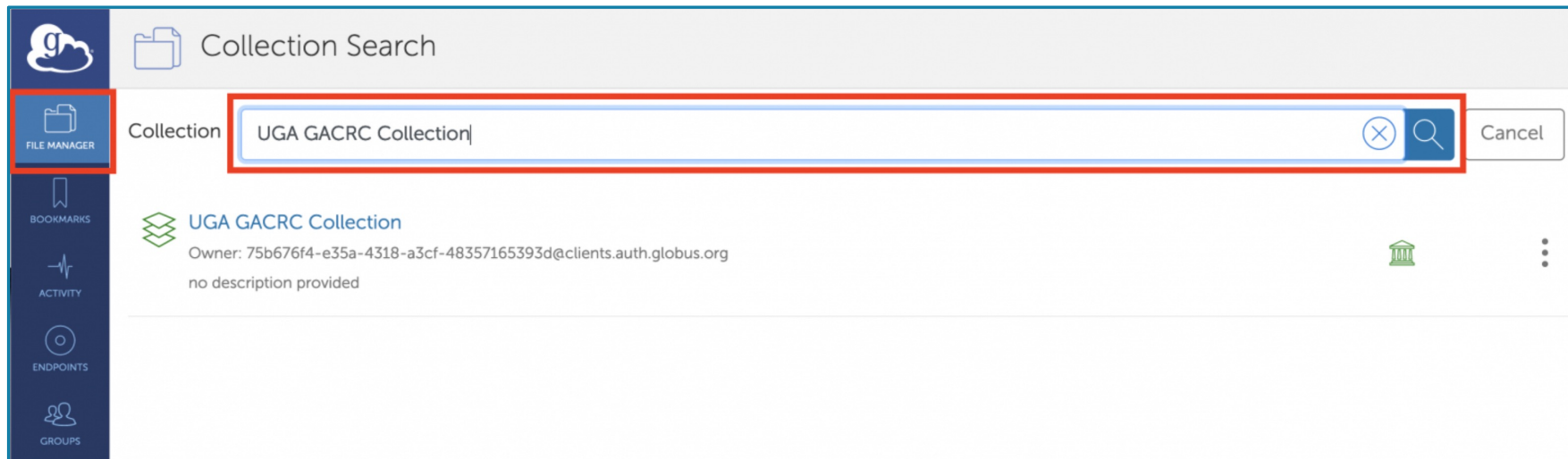
```
zhuofei@ss-sub1 zhuofei$ cd workDir
zhuofei@ss-sub1 workDir$ pwd
/scratch/zhuofei/workDir
zhuofei@ss-sub1 workDir$ ls
```

← it is empty!

# Step 5: Transfer Data to the Cluster using Globus

<https://wiki.gacrc.uga.edu/wiki/Globus>

1. Create a Globus Identity Account at globus.org following the instructions at [https://wiki.gacrc.uga.edu/wiki/Globus#Getting\\_Started](https://wiki.gacrc.uga.edu/wiki/Globus#Getting_Started)
2. From the File Manager tab at globus.org, search for the “UGA GACRC Collection”





# Step 5 (Cont.): Transfer Data to the Cluster using Globus

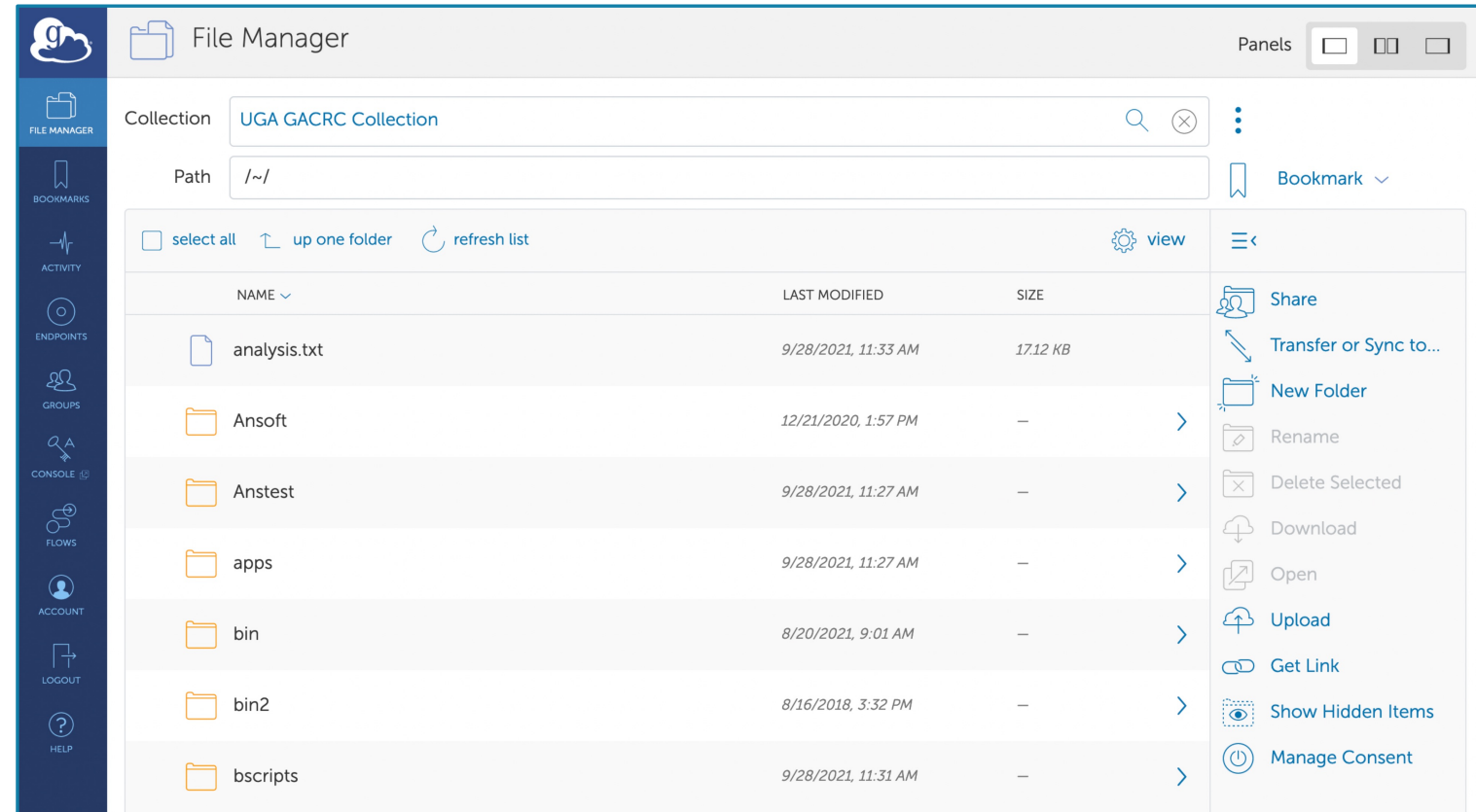
<https://wiki.gacrc.uga.edu/wiki/Globus>

When you open the UGA GACRC Collection in Globus, by default you will be in your home directory. You can change to any of your or your lab's other directories by typing a path in the Path bar.

Use the **Upload/Download** options on the right for transferring **small** or a **few** files.

Use the **Transfer or Sync** option for **many** or **large** files (requires Globus Connect Personal on your computer:

[https://wiki.gacrc.uga.edu/wiki/Globus\\_Connect\\_Personal](https://wiki.gacrc.uga.edu/wiki/Globus_Connect_Personal))

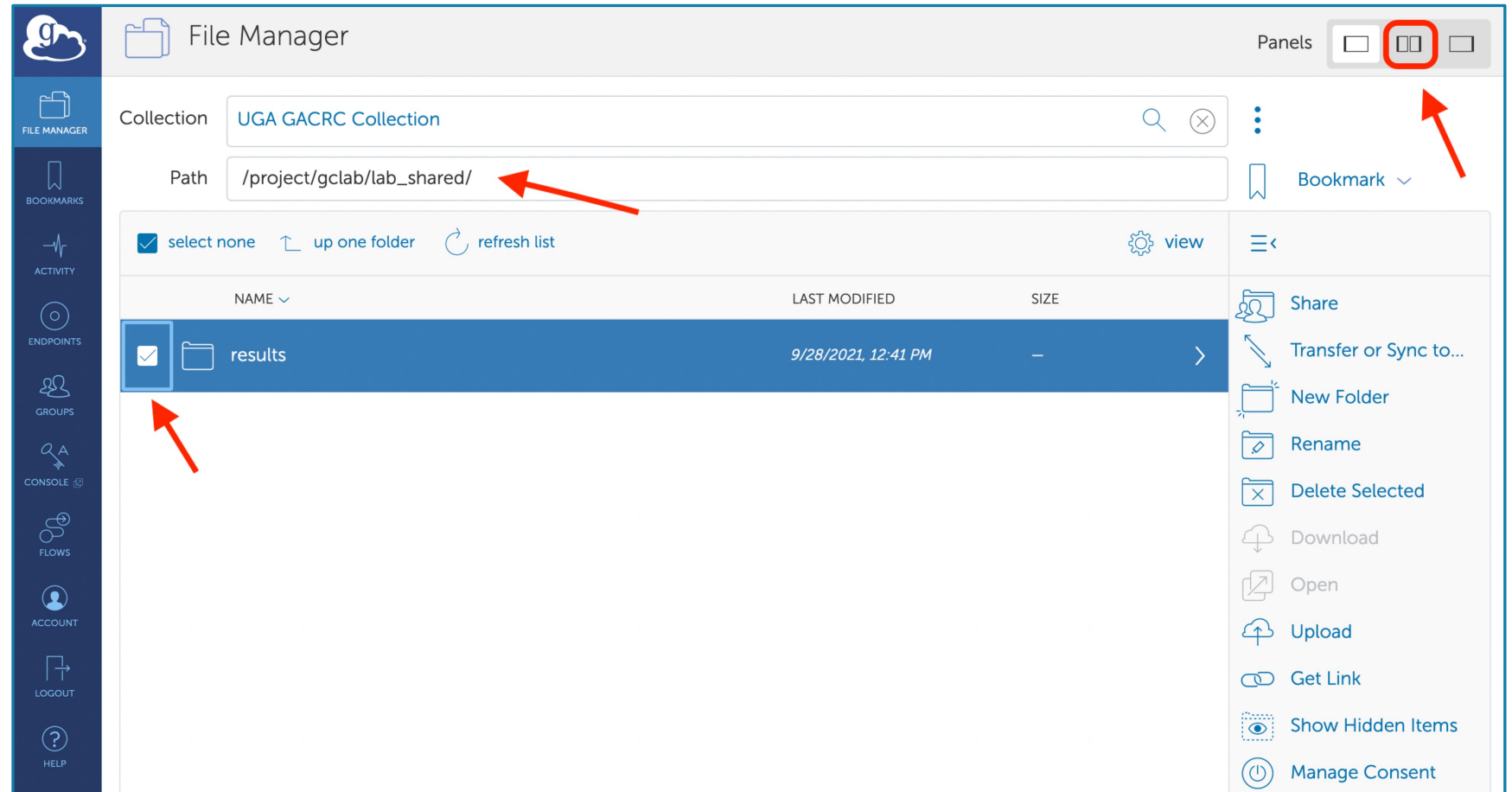


# Step 5 (Cont.): Transfer Data to the Cluster using Globus

<https://wiki.gacrc.uga.edu/wiki/Globus>

## Example 1:

Using the Transfer or Sync option from between one's personal computer and a /project directory on the cluster

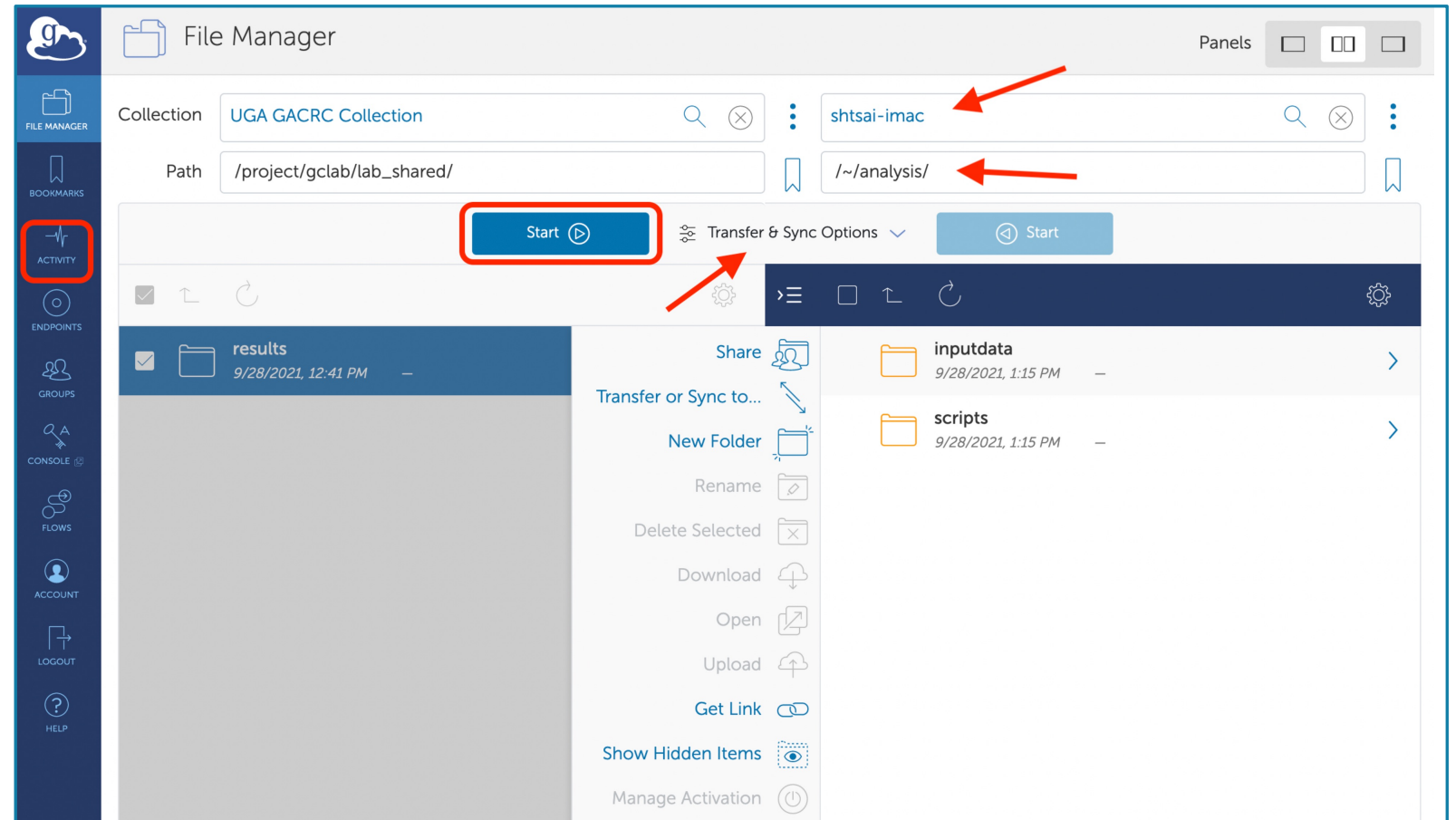


# Step 5 (Cont.): Transfer Data to the Cluster using Globus

<https://wiki.gacrc.uga.edu/wiki/Globus>

## Example 1 (Cont.):

Using the Transfer or Sync option from between one's personal computer and a /project directory on the cluster

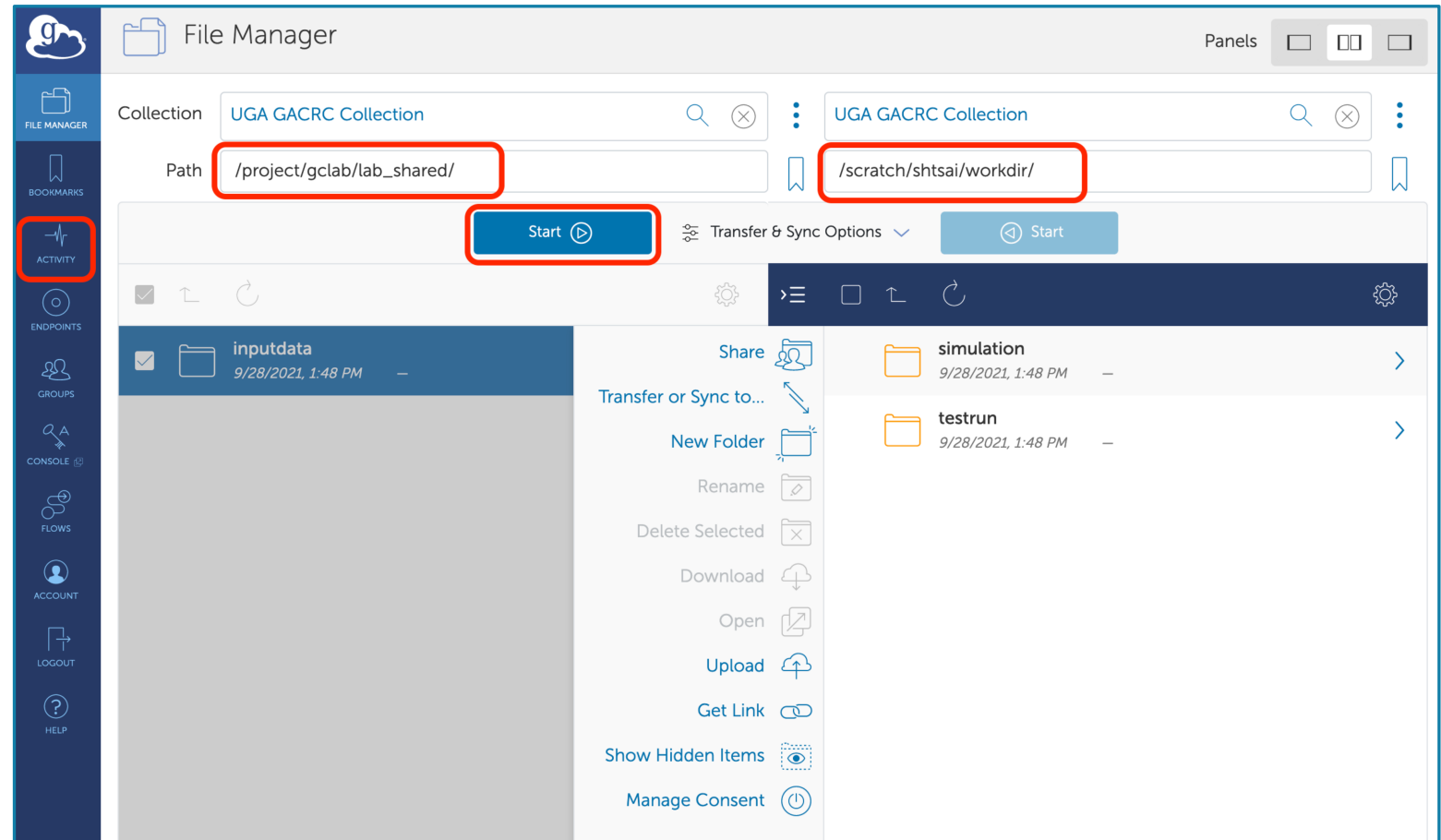


# Step 5 (Cont.): Transfer Data on the Cluster to workDir using Globus

<https://wiki.gacrc.uga.edu/wiki/Globus>

## Example 2:

Using the Transfer or Sync option between two places on the cluster



## Step 6: Make a job submission script in workDir

[https://wiki.gacrc.uga.edu/wiki/Sample\\_batch\\_job\\_submission\\_scripts\\_on\\_Sapelo2](https://wiki.gacrc.uga.edu/wiki/Sample_batch_job_submission_scripts_on_Sapelo2)

```
$ nano sub.sh
```

nano is a simple text editor on Linux. You are welcome to use other editors like vim or emacs.

Ctrl-x to save file and quit from nano



```
#!/bin/bash
#SBATCH --job-name=testBowtie2      # Job name (testBowtie2)
#SBATCH --partition=batch           # Partition name (batch, highmem_p, or gpu_p)
#SBATCH --ntasks=1                  # 1 task (process) for below commands
#SBATCH --cpus-per-task=1           # CPU core count per task, by default 1 CPU core per task
#SBATCH --mem=4G                    # Memory per node (4GB); by default using M as unit
#SBATCH --time=1:00:00              # Time limit hrs:min:sec or days-hours:minutes:seconds
#SBATCH --output=%x_%j.out          # Standard output log, e.g., testBowtie2_12345.out
#SBATCH --mail-user=username@uga.edu # Where to send mail
#SBATCH --mail-type=END,FAIL        # Mail events (BEGIN, END, FAIL, ALL)

ml Bowtie2/2.4.1-GCC-8.3.0          # Load software module and run bowtie2 below

bowtie2 -x index/lambda_virus -U myreads.fq
```

```
^G Get Help      ^O WriteOut      ^R Read File     ^Y Prev Page     ^K Cut Text
^X Exit          ^J Justify       ^W Where Is      ^V Next Page     ^U UnCut Text
```

To run the demo example, please copy  
these files into your working dir:

```
cp -r /usr/local/training/Sapelo2/* .
```

# Submission scripts do not have to be complex!

[https://wiki.gacrc.uga.edu/wiki/Sample\\_batch\\_job\\_submission\\_scripts\\_on\\_Sapelo2](https://wiki.gacrc.uga.edu/wiki/Sample_batch_job_submission_scripts_on_Sapelo2)

```
#!/bin/bash

#SBATCH --partition=batch
#SBATCH --ntasks=1
#SBATCH --mem=4G
#SBATCH --time=1:00:00

m1 Bowtie2/2.4.1-GCC-8.3.0
=
bowtie2 -x index/lambda_virus -U myreads.fq
```

The only required Slurm headers are:

- --partition
- --ntasks
- --mem
- --time

(Slurm will let you know if you forgot one when you try to submit your submission script)

Some default values if not specified:

- --cpus-per-task=1
- --nodes=1

## Step 7: Submit a job from workDir using **sbatch**

[https://wiki.gacrc.uga.edu/wiki/Running\\_Jobs\\_on\\_Sapelo2#How\\_to\\_submit\\_a\\_job\\_to\\_the\\_batch\\_partition](https://wiki.gacrc.uga.edu/wiki/Running_Jobs_on_Sapelo2#How_to_submit_a_job_to_the_batch_partition)

```
zhuofei@ss-sub1 workDir$ pwd
/scratch/zhuofei/workDir
zhuofei@ss-sub1 workDir$ ls
index  myreads.fq  sub.sh
zhuofei@ss-sub1 workDir$ sbatch sub.sh
Submitted batch job 32860
```

**sub.sh** is a job submission script to

1. specify computing resources:
2. load software using **ml *moduleName***
3. run any Linux commands you want to run
4. run the software



## Step 8: Check pending/running job status using `squeue/sq --me`

[https://wiki.gacrc.uga.edu/wiki/Monitoring\\_Jobs\\_on\\_Sapelo2](https://wiki.gacrc.uga.edu/wiki/Monitoring_Jobs_on_Sapelo2)

```
zhuofei@ss-sub1 workdir$ squeue --me
```

JOBID	PARTITION	NAME	USER	NODES	TIME	ST	NODELIST (REASON)
4618668	highmem_p	test-job	zhuofei	1	1:16	R	d4-11
4618666	batch	Bowtie2-2cpu	zhuofei	1	1:19	R	c1-23
4618665	batch	testBowtie2	zhuofei	1	5:23	R	c1-23

```
zhuofei@ss-sub1 workdir$ sq --me
```

JOBID	TIME	TIME_LIMIT	NAME	PARTITION	USER	NODES	CPUS	MIN_MEMORY	PRIORITY	STATE	NODELIST (REASON)
4618668	1:18	4:00:00	test-job	highmem_p	zhuofei	1	1	200G	5993	RUNNING	d4-11
4618666	1:21	1:00:00	Bowtie2-2cpu	batch	zhuofei	1	2	4G	5991	RUNNING	c1-23
4618665	5:25	2:00:00	testBowtie2	batch	zhuofei	1	1	4G	5991	RUNNING	c1-23

## sq --help output

[https://wiki.gacrc.uga.edu/wiki/Monitoring\\_Jobs\\_on\\_Sapelo2](https://wiki.gacrc.uga.edu/wiki/Monitoring_Jobs_on_Sapelo2)

```
zhuofei@ss-sub1 workDir$ sq --help
```

```
Usage: sq [OPTIONS]
```

```
Descriptions: sq - preformatted wrapper for squeue.  See man squeue for more information.
```

-j	Displays squeue output for a given job
--me	Displays squeue output for the user executing this command
-p	Displays squeue output for a given partition
-u	Displays squeue output for a given user
-T	Displays submit and start time columns
-h, --help	Displays this help output

## Step8 (Cont.): Cancel job using **scancel**

[https://wiki.gacrc.uga.edu/wiki/Running\\_Jobs\\_on\\_Sapelo2#How\\_to\\_cancel\\_.28delete.29\\_a\\_running\\_or\\_pending\\_job](https://wiki.gacrc.uga.edu/wiki/Running_Jobs_on_Sapelo2#How_to_cancel_.28delete.29_a_running_or_pending_job)

```
zhuofei@ss-sub1 workDir$ queue --me -l
Tue Sep 15 15:08:27 2020
JOBID PARTITION      NAME      USER      STATE      TIME  TIME_LIMI  NODES  NODELIST (REASON)
32869      batch  testBowt  zhuofei    RUNNING    0:01    1:00:00      1  c5-19
32868      batch  testBowt  zhuofei    RUNNING    0:05    1:00:00      1  c5-19
32867      batch  testBowt  zhuofei    RUNNING    0:10    1:00:00      1  c5-19
zhuofei@ss-sub1 workDir$
zhuofei@ss-sub1 workDir$ scancel 32867
zhuofei@ss-sub1 workDir$ queue --me -l
Tue Sep 15 15:08:45 2020
JOBID PARTITION      NAME      USER      STATE      TIME  TIME_LIMI  NODES  NODELIST (REASON)
32867      batch  testBowt  zhuofei    COMPLETI    0:26    1:00:00      1  c5-19
32869      batch  testBowt  zhuofei    RUNNING    0:19    1:00:00      1  c5-19
32868      batch  testBowt  zhuofei    RUNNING    0:23    1:00:00      1  c5-19
zhuofei@ss-sub1 workDir$ queue --me -l
Tue Sep 15 15:08:50 2020
JOBID PARTITION      NAME      USER      STATE      TIME  TIME_LIMI  NODES  NODELIST (REASON)
32869      batch  testBowt  zhuofei    RUNNING    0:19    1:00:00      1  c5-19
32868      batch  testBowt  zhuofei    RUNNING    0:23    1:00:00      1  c5-19
```

## Step 8 (Cont.): Check running or finished job status using `sacct -X` / `sacct-gacrc -X`

[https://wiki.gacrc.uga.edu/wiki/Monitoring\\_Jobs\\_on\\_Sapelo2](https://wiki.gacrc.uga.edu/wiki/Monitoring_Jobs_on_Sapelo2)

```
zhuofei@ss-sub1 workdir$ sacct -X
```

JobID	JobName	Partition	Account	AllocCPUS	State	ExitCode
4618312	interact	inter_p	gacrc-ins+	1	COMPLETED	0:0
4618665	testBowti+	batch	gacrc-ins+	1	COMPLETED	0:0
4618666	Bowtie2-2+	batch	gacrc-ins+	2	COMPLETED	0:0
4618668	test-job	highmem_p	gacrc-ins+	1	COMPLETED	0:0

```
zhuofei@ss-sub1 workdir$ sacct-gacrc -X
```

JobID	JobName	User	Partition	NNode	NCPUS	ReqMem	CPUTime	Elapsed	Timelimit	State	ExitCode	NodeList
4618312	interact	bc06026	inter_p	1	1	2Gn	00:07:01	00:07:01	12:00:00	COMPLETED	0:0	c4-20
4618665	testBowtie2	bc06026	batch	1	1	4Gn	00:23:12	00:23:12	01:00:00	COMPLETED	0:0	c1-23
4618666	Bowtie2-2cpu	bc06026	batch	1	2	4Gn	00:20:24	00:10:12	01:00:00	COMPLETED	0:0	c1-23
4618668	test-job	bc06026	highmem_p	1	1	200Gn	00:09:46	00:09:46	04:00:00	COMPLETED	0:0	d4-11

# sacct-gacrc --help output

[https://wiki.gacrc.uga.edu/wiki/Monitoring\\_Jobs\\_on\\_Sapelo2](https://wiki.gacrc.uga.edu/wiki/Monitoring_Jobs_on_Sapelo2)

```
zhuofei@ss-sub1 workDir$ sacct-gacrc --help
```

```
Usage: sacct-gacrc [OPTIONS]
```

```
Description: preformatted wrapper for sacct.  See man sacct for more information.
```

-E, --endtime	Display information about jobs up to a date, in the format of yyyy-mm-dd (default: now)
-j, --jobs	Display information about a particular job or jobs (comma-separated list if more than one job)
-r, --partition	Display information about jobs from a particular partition
-S, --starttime	Display information about jobs starting from a date in the format of yyyy-mm-dd (default: Midnight of today)
-u, --user	Display information about a particular user's job(s) (default: current user)
-X, --allocations	Only show one line per job (do not display job steps)
--debug	Display the sacct command being executed
-h, --help	Display this help output

## Step 8 (Cont.): Check resource usage of finished jobs using **seff**

[https://wiki.gacrc.uga.edu/wiki/Monitoring\\_Jobs\\_on\\_Sapelo2](https://wiki.gacrc.uga.edu/wiki/Monitoring_Jobs_on_Sapelo2)

```
zhuofei@ss-sub1 workDir$ seff 37259
```

Job ID: 37259

Cluster: tc2

User/Group: zhuofei/gacrc-instruction

State: **COMPLETED** (exit code 0)

Cores: 1

**CPU Utilized: 00:09:45**

**CPU Efficiency: 99.66% of 00:09:47 core-walltime**

**Job Wall-clock time: 00:09:47**

**Memory Utilized: 197.34 MB**

**Memory Efficiency: 4.82% of 4.00 GB**

# Obtain Job Details

[https://wiki.gacrc.uga.edu/wiki/Running\\_Jobs\\_on\\_Sapelo2#How\\_to\\_check\\_resource\\_utilization\\_of\\_a\\_running\\_or\\_finished\\_job](https://wiki.gacrc.uga.edu/wiki/Running_Jobs_on_Sapelo2#How_to_check_resource_utilization_of_a_running_or_finished_job)

---

Option 1: `squeue` or `sq --me` for details of a pending or running jobs

Option 2: `sacct -X` or `sacct-gacrc -X` for details of running or finished jobs

Option 3: `seff` for details of computing resource usage of a finished job

Option 4: Email notification from finished jobs (completed, canceled, or crashed), if using:

```
#SBATCH --mail-user=username@uga.edu
```

```
#SBATCH --mail-type=ALL
```

# Interactive jobs

[https://wiki.gacrc.uga.edu/wiki/Running\\_Jobs\\_on\\_Sapelo2#How to open an interactive session](https://wiki.gacrc.uga.edu/wiki/Running_Jobs_on_Sapelo2#How_to_open_an_interactive_session)

[https://wiki.gacrc.uga.edu/wiki/Running\\_Jobs\\_on\\_Sapelo2#How to run an interactive job with Graphical User Interface capabilities](https://wiki.gacrc.uga.edu/wiki/Running_Jobs_on_Sapelo2#How_to_run_an_interactive_job_with_Graphical_User_Interface_capabilities)

Description	Command
Start an interactive session	<code>interact</code>
Start an interactive session with X forwarding	<code>interact --x11</code>

<code>interact</code>	<code>srun --pty --cpus-per-task=1 --job-name=interact --ntasks=1 --nodes=1 --partition=inter_p --time=12:00:00 --mem=2GB /bin/bash -l</code>
<code>interact --x11</code>	<code>srun --pty --cpus-per-task=1 --job-name=interact --ntasks=1 --nodes=1 --partition=inter_p --time=12:00:00 --mem=2GB --x11 /bin/bash -l</code>



# interact --help output

[https://wiki.gacrc.uga.edu/wiki/Running\\_Jobs\\_on\\_Sapelo2](https://wiki.gacrc.uga.edu/wiki/Running_Jobs_on_Sapelo2)

```
zhuofei@ss-sub1 workDir$ interact --help
```

Usage: interact [OPTIONS]

Description: Start an interactive job

-c, --cpus-per-task	CPU cores per task (default: 1)
-J, --job-name	Job name (default: interact)
-n, --ntasks	Number of tasks (default: 1)
-N, --nodes	Number of nodes (default: 1)
-p, --partition	Partition for interactive job (default: inter_p)
-q, --qos	Request a quality of service for the job.
-t, --time	Maximum run time for interactive job (default: 12:00:00)
-w, --nodelist	List of node name(s) on which your job should run
--constraint	Job constraints
--gres	Generic consumable resources
--mem	Memory per node (default 2GB)
--shell	Absolute path to the shell to be used in your interactive job (default: /bin/bash)
--wckey	Wckey to be used with job
--x11	Start an interactive job with X Forwarding
-h, --help	Display this help output

GACRC Wiki <http://wiki.gacrc.uga.edu>

Kaltura channel <https://kaltura.uga.edu/channel/GACRC/176125031>

---

System: <https://wiki.gacrc.uga.edu/wiki/Systems#Sapelo2>

Connection: [https://wiki.gacrc.uga.edu/wiki/Connecting#Connecting\\_to\\_Sapelo2](https://wiki.gacrc.uga.edu/wiki/Connecting#Connecting_to_Sapelo2)

Software: [https://wiki.gacrc.uga.edu/wiki/Software\\_on\\_Sapelo2](https://wiki.gacrc.uga.edu/wiki/Software_on_Sapelo2)

Running Jobs: [https://wiki.gacrc.uga.edu/wiki/Running\\_Jobs\\_on\\_Sapelo2](https://wiki.gacrc.uga.edu/wiki/Running_Jobs_on_Sapelo2)

Monitoring Jobs: [https://wiki.gacrc.uga.edu/wiki/Monitoring\\_Jobs\\_on\\_Sapelo2](https://wiki.gacrc.uga.edu/wiki/Monitoring_Jobs_on_Sapelo2)

Sample scripts : [https://wiki.gacrc.uga.edu/wiki/Sample\\_batch\\_job\\_submission\\_scripts\\_on\\_Sapelo2](https://wiki.gacrc.uga.edu/wiki/Sample_batch_job_submission_scripts_on_Sapelo2)

Transferring Files: <https://wiki.gacrc.uga.edu/wiki/Globus>

Linux Commands: [https://wiki.gacrc.uga.edu/wiki/Command\\_List](https://wiki.gacrc.uga.edu/wiki/Command_List)

Open OnDemand: <https://wiki.gacrc.uga.edu/wiki/OnDemand>

Training: <https://wiki.gacrc.uga.edu/wiki/Training>

# GACRC Help and Support

[https://wiki.gacrc.uga.edu/wiki/Getting\\_Help](https://wiki.gacrc.uga.edu/wiki/Getting_Help)

---

## ➤ Job Troubleshooting:

Please tell us details of your question or problem, including but not limited to:

- ✓ Your user name
- ✓ Your job ID
- ✓ Your working directory
- ✓ The queue name and command you used to submit the job

## ➤ Software Installation:

- ✓ Specific name and version of the software
- ✓ Download website
- ✓ Supporting package information if have

When you ask GACRC to test or troubleshoot your jobs, Please make sure of the correctness of your datasets being used!

# GACRC Service Catalog

Georgia Advanced Computing Resource Center (GACRC) service catalog.

If you would like to reach out to GACRC and do not have a UGA MyID, please send an email to [gacrc-help@uga.edu](mailto:gacrc-help@uga.edu), and we will respond promptly.

## Categories (3)

### Services For Users

General user support, request software installation or update, request training.

### Services for PIs

For PIs only: Lab registration, user account creation/modification, class account requests, storage quota modifications.

### For GACRC Staff

For GACRC's internal use only.

# Services For Users

General user support, request software installation or update, request training.

## Services (3)

---

### General Support

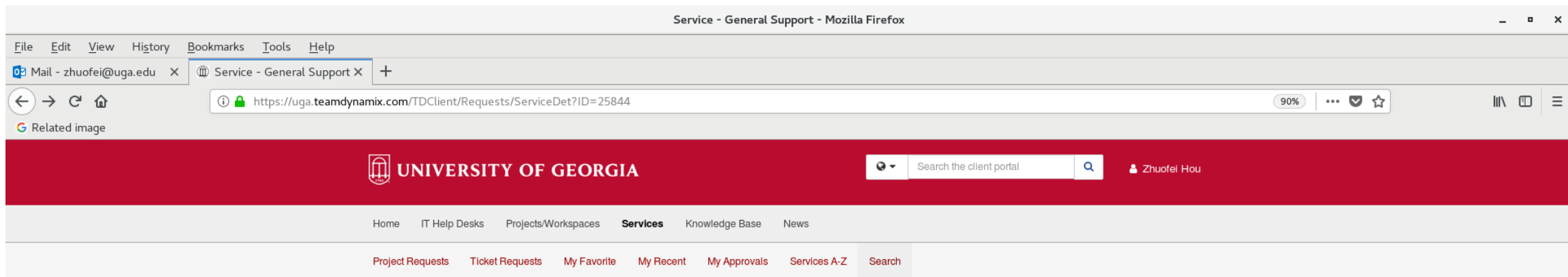
Report issues and request help with GACRC systems, except for software installation requests and account/lab creation requests.

### Software Installation/Update

Request software and common application database (e.g. NCBI blast databases) installation and upgrade.

### Training

Request support related to training provided by the GACRC.



[Service Catalog](#) / [Academics, Learning & Research](#) / [GACRC Service Catalog](#) / General Support

## General Support

If you do not have a myid, please mail [gacrc-help@uga.edu](mailto:gacrc-help@uga.edu), and we will respond promptly.

The purpose of this form is to provide a method to report issues and to request help with GACRC systems.

Please use this form for all questions and support needs (e.g. to report issues, to troubleshoot jobs, to request resources or grant writing help, etc). Please do not use this form for software installation requests or lab/user account management, which all have separate forms.

Please refer to the GACRC documentation for information on GACRC resources, how to connect and transfer files, how to run jobs, installed software list, training schedule, and a FAQ.

The link to this documentation is <https://wiki.gacrc.uga.edu>

 Request Service

 Share

 Add to Favorites

 Click to request

This site is operated by [Enterprise Information Technology Services \(EITS\)](#) at the University of Georgia.

[Privacy](#) | [Accessibility](#) | [Website Feedback](#)

<https://uga.teamdynamix.com/TDClient/Requests/ServiceCatalogSearch>


General Support - Mozilla Firefox



FileEditViewHistoryBookmarksToolsHelp


Mail - zhuofei@uga.eduGeneral Support

https://uga.teamdynamix.com/TDClient/Requests/TicketRequests/NewForm?ID=MNeY5EiDcMY\_90%

Related image

UNIVERSITY OF GEORGIA


Search the client portal


Zhuofei Hou

HomeIT Help DesksProjects/WorkspacesServicesKnowledge BaseNews


[Service Catalog](#) / [Academics, Learning & Research](#) / [GACRC Service Catalog](#) / [General Support](#)

General Support

 Show Help

 Hide Help

Report issues and request help with GACRC systems, except for software installation requests and account/lab creation requests.

Short Description \*

Email \*

MyID \*

Phone Number \*

Support Needed For

☐ Galaxy

☐ Sapelo2

☐ Teaching Cluster

☐ Work Filesystem

☐ Home Filesystem

☐ Scratch Filesystem

☐ Project Filesystem

☐ Xfer Nodes

☐ Other

Lab \*

# Slurm job states

Code	State	Meaning
R	Running	Job is running on compute node(s)
PD	Pending	Job is waiting for compute node(s)
CD	Completed	Job completed
CG	Completing	Job is completing
CA	Canceled	Job was canceled
F	Failed	Job terminated with non-zero exit code
NF	Node Fail	Job terminated due to failure of node(s)



# Commands for submitting and canceling jobs

Description	Slurm Command
Submit a batch job to queue	<code>sbatch sub.sh</code>
Delete a job from queue	<code>scancel &lt;jobID&gt;</code>
Cancel all your job(s)	<code>scancel -u &lt;username&gt;</code>
Cancel all your pending job(s)	<code>scancel -t PENDING -u &lt;username&gt;</code>
Cancel your job(s) by job name	<code>scancel --name &lt;jobname&gt;</code>
Cancel an element (index) of an array job (jobID)	<code>scancel &lt;jobID&gt;_&lt;index&gt;</code>

# Commands for monitoring jobs

Description	Command
Information about currently running jobs	<code>squeue</code> or <code>sq</code>
Information about <b>your</b> currently running jobs	<code>squeue --me</code> or <code>sq --me</code>
Information about a user's currently running jobs	<code>squeue -u &lt;username&gt;</code>
Information about your running or finished jobs	<code>sacct</code> or <code>sacct-gacrc</code>
Job's resource usage	<code>seff &lt;job ID&gt;</code>

# Slurm headers for running a **Serial (single-core)** job

[https://wiki.gacrc.uga.edu/wiki/Sample\\_batch\\_job\\_submission\\_scripts\\_on\\_Sapelo2](https://wiki.gacrc.uga.edu/wiki/Sample_batch_job_submission_scripts_on_Sapelo2)

```
#!/bin/bash
#SBATCH --job-name=testBowtie2          # Job name (testBowtie2)
#SBATCH --partition=batch               # Queue name (batch)
#SBATCH --ntasks=1                     # Run in a single task using one CPU core on a single node
#SBATCH --mem=4G                        # Job memory limit (4 GB)
#SBATCH --time=1:00:00                  # Time limit hrs:min:sec or days-hours:minutes:seconds
#SBATCH --export=NONE                   # Do not load any users' explicit environment variables
#SBATCH --output=%x_%j.out              # Standard output log, e.g., testBowtie2_1234.out
#SBATCH --error=%x_%j.err               # Standard error log, e.g., testBowtie2_1234.err
#SBATCH --mail-type=END,FAIL            # Mail events (BEGIN, END, FAIL, ALL)
#SBATCH --mail-user=username@uga.edu    # Where to send mail

cd $SLURM_SUBMIT_DIR                    # Change directory to job submission directory
ml Bowtie2/2.4.1-GCC-8.3.0              # Load software module and run bowtie2 below
bowtie2 -x ./index/lambda_virus -U ./myreads.fq -S output.sam
```

# Slurm headers for running a Threaded job

```
#!/bin/bash
#SBATCH --job-name=testBowtie2          # Job name (testBowtie2)
#SBATCH --partition=batch                # Queue name (batch)
#SBATCH --nodes=1                       # Run all processes on a single node
#SBATCH --ntasks=1                      # Run in a single task on a single node
#SBATCH --cpus-per-task=8                # Number of CPU cores per task (8)
#SBATCH --mem=10G                       # Job memory limit (10 GB)
#SBATCH --time=1:00:00                  # Time limit hrs:min:sec or days-hours:minutes:seconds
#SBATCH --export=NONE                   # Do not load any users' explicit environment variables
#SBATCH --output=%x_%j.out              # Standard output log, e.g., testBowtie2_1234.out
#SBATCH --error=%x_%j.err               # Standard error log, e.g., testBowtie2_1234.err
#SBATCH --mail-type=END,FAIL             # Mail events (BEGIN, END, FAIL, ALL)
#SBATCH --mail-user=username@uga.edu    # Where to send mail

cd $SLURM_SUBMIT_DIR
ml Bowtie2/2.4.1-GCC-8.3.0
bowtie2 -p 8 -x ./index/lambda_virus -U ./myreads.fq -S output.sam
```

# Slurm headers for running an **Array** job

```
#!/bin/bash
#SBATCH --job-name=testBowtie2Array      # Job name (testBowtie2Array)
#SBATCH --partition=batch                # Queue name (batch)
#SBATCH --ntasks=1                      # Run in a single task using one CPU core on a single node
#SBATCH --mem=4G                        # Job memory limit (4 GB)
#SBATCH --time=1:00:00                  # Time limit hrs:min:sec or days-hours:minutes:seconds
#SBATCH --export=NONE                   # Do not load any users' explicit environment variables
#SBATCH --output=%x_%j.out               # Standard output log, e.g., testBowtie2Array_1234.out
#SBATCH --error=%x_%j.err                # Standard error log, e.g., testBowtie2Array_1234.err
#SBATCH --array=0-9                     # Array element range from 0 to 9, i.e. 10 element jobs

cd $SLURM_SUBMIT_DIR
ml Bowtie2/2.4.1-GCC-8.3.0               # Original data is split into 10 pieces and run in each element job
bowtie2 -x ./index/lambda_virus -U ./myreads_$(SLURM_ARRAY_TASK_ID).fq \
-S output_$(SLURM_ARRAY_TASK_ID).sam
```

# Slurm headers for running a Threaded (OpenMP) job

```
#!/bin/bash
#SBATCH --job-name=testOpenMP           # Job name (testOpenMP)
#SBATCH --partition=batch                # Queue name (batch)
#SBATCH --nodes=1                       # Run all processes on a single node
#SBATCH --ntasks=1                      # Run in a single task on a single node
#SBATCH --cpus-per-task=12              # Number of CPU cores per task (12)
#SBATCH --mem=10G                       # Job memory limit (10 GB)
#SBATCH --export=NONE                   # Do not load any users' explicit environment variables
#SBATCH --time=24:00:00                 # Time limit hrs:min:sec or days-hours:minutes:seconds
#SBATCH --output=%x_%j.log              # Standard output and error log, e.g., testOpenMP_1234.log
#SBATCH --mail-type=END,FAIL            # Mail events (BEGIN, END, FAIL, ALL)
#SBATCH --mail-user=username@uga.edu    # Where to send mail

cd $SLURM_SUBMIT_DIR
export OMP_NUM_THREADS=$SLURM_CPUS_PER_TASK # Sets the number of threads to use for OpenMP parallel regions
ml foss/2019b                               # Load toolchain module
./myProgram.x                              # Run your program binary compiled with OpenMP
```

# Slurm headers for running an MPI job

```
#!/bin/bash
#SBATCH --job-name=testMPI           # Job name (tesMPI)
#SBATCH --partition=batch           # Queue name (batch)
#SBATCH --nodes=2                   # Run on two nodes
#SBATCH --ntasks-per-node=16        # How many tasks on each node; Number of tasks=32=MPI ranks
#SBATCH --cpus-per-task=1           # Number of CPU cores per task; 16 CPU cores per node
#SBATCH --mem-per-cpu=500M          # Memory per allocated CPU; 8GB (500MB*16) memory per node
#SBATCH --time=24:00:00             # Time limit hrs:min:sec or days-hours:minutes:seconds
#SBATCH --export=NONE               # Do not load any users' explicit environment variables
#SBATCH --output=%x_%j.log          # Standard output and error log, e.g., testMPI_1234.log
#SBATCH --mail-type=END,FAIL        # Mail events (BEGIN, END, FAIL, ALL)
#SBATCH --mail-user=username@uga.edu # Where to send mail

cd $SLURM_SUBMIT_DIR
ml foss/2019b                       # Load toolchain module
mpirun -n 32 ./myProgram.x          # Run your program binary compiled with OpenMPI with 32 ranks
```

# Slurm headers for running a Hybrid MPI/OpenMP job

```
#!/bin/bash
#SBATCH --job-name=testHybrid      # Job name (testHybrid)
#SBATCH --partition=batch          # Queue name (batch)
#SBATCH --nodes=2                  # Run on two nodes
#SBATCH --ntasks-per-node=8        # How many tasks on each node; Number of tasks=16=MPI ranks
#SBATCH --cpus-per-task=4          # Number of CPU cores per task; 32 CPU cores per node
#SBATCH --mem-per-cpu=500M         # Memory per allocated CPU; 16GB (500MB*32) memory per node
#SBATCH --time=24:00:00            # Time limit hrs:min:sec or days-hours:minutes:seconds
#SBATCH --export=NONE              # Do not load any users' explicit environment variables
#SBATCH --output=%x_%j.log         # Standard output and error log
#SBATCH --mail-type=END,FAIL       # Mail events (BEGIN, END, FAIL, ALL)
#SBATCH --mail-user=username@uga.edu # Where to send mail

cd $SLURM_SUBMIT_DIR               # Change directory to job submission directory
export OMP_NUM_THREADS=$SLURM_CPUS_PER_TASK # Sets the number of threads to use for OpenMP parallel regions
ml foss/2019b                      # Load toolchain module
mpirun -n 16 ./myProgram.x         # Run your program binary compiled with OpenMPI with 16 ranks
```



# Slurm headers for running a GPU job

```
#!/bin/bash
#SBATCH --job-name=amber                # Job name
#SBATCH --partition=gpu_p               # Partition (queue) name, i.e., gpu_p
#SBATCH --gres=gpu:1                   # Requests one GPU device; --gres=gpu:P100:1, --gres=gpu:K40:1
#SBATCH --ntasks=1                     # Run a single task
#SBATCH --cpus-per-task=2               # Number of CPU cores per task
#SBATCH --mem=40gb                      # Job memory request
#SBATCH --time=10:00:00                 # Time limit hrs:min:sec
#SBATCH --export=NONE                   # Do not load any users' explicit environment variables
#SBATCH --output=%x_%j.out              # Standard output and error log, e.g., amber_1234.out
#SBATCH --mail-type=END,FAIL            # Mail events (BEGIN, END, FAIL, ALL)
#SBATCH --mail-user=username@uga.edu    # Where to send mail

cd $SLURM_SUBMIT_DIR
ml Amber/18-fosscuda-2018b-AmberTools-18-patchlevel-10-8
mpirun $AMBERHOME/bin/pmemd.cuda -O -i ./prod.in -o prod_c4-23.out -p ./dimerFBP_GOL.prmtop -c ./restart.rst \
-r prod.rst -x prod.mdcrd
```

# Slurm headers for running a Singularity container

```
#!/bin/bash
#SBATCH --job-name=test_sortmerna           # Job name
#SBATCH --partition=batch                   # Partition (queue) name
#SBATCH --ntasks=1                         # Run on a single CPU
#SBATCH --mem=8gb                           # Job memory request
#SBATCH --time=02:00:00                    # Time limit hrs:min:sec
#SBATCH --export=NONE                       # Do not load any users' explicit environment variables
#SBATCH --output=sortmerna.%j.out           # Standard output log, e.g., sortmerna.1234.out
#SBATCH --error=sortmerna.%j.err            # Standard error log, e.g., sortmerna.1234.err
#SBATCH --cpus-per-task=4                   # Number of CPU cores per task
#SBATCH --mail-type=END,FAIL                # Mail events (NONE, BEGIN, END, FAIL, ALL)
#SBATCH --mail-user=username@uga.edu        # Where to send mail

cd $SLURM_SUBMIT_DIR
singularity exec /apps/singularity-images/sortmerna-3.0.3.simg sortmerna --threads 4 --ref db.fasta,db.idx --reads file.fa \
--aligned base_name_output
```

# General guidelines

---

- Do NOT use Login nodes to run CPU/memory intensive tasks directly → submit jobs to Compute nodes!
- Do NOT use Login nodes to transfer data between your local computer and cluster → use Transfer nodes!
- Do NOT use Home for storing job data → use /scratch/MyID
- Do NOT park data in Scratch or Local Scratch → clean up when job finishes or exits from node
- Do NOT park data permanently in Project → download data to your local drive
- NO large memory job running on batch partition → use highmem\_p
- NO small memory job running on highmem\_p partition → use batch
- In general, number of threads you want to run with a parallel job = number of cores requested
- When you archive data using **tar** on /scratch, please **do not use the z option** (compression option). After you archived data with tar, you can use gzip to compress it.

# General guidelines

- **No directory should not have too many files inside!** A rule of thumb would be to try to keep no more than a few tens of thousands of files (<10000 would be even better) in any single directory which is accessed frequently



All files are in ONE single dir!



Files are organized in subdirs!



---

# Thank You!

## **Telephone Support**

EITS Help Desk: 706-542-3106

Monday – Thursday: 7:30 a.m. – 7:30 p.m.

Friday: 7:30 a.m. – 6 p.m.

Saturday – Sunday: 1 p.m. – 7 p.m.

## ***Georgia Advanced Computing Resource Center***

*101-108 Computing Services building*

*University of Georgia*

*Athens, GA 30602*

<https://gacrc.uga.edu/>