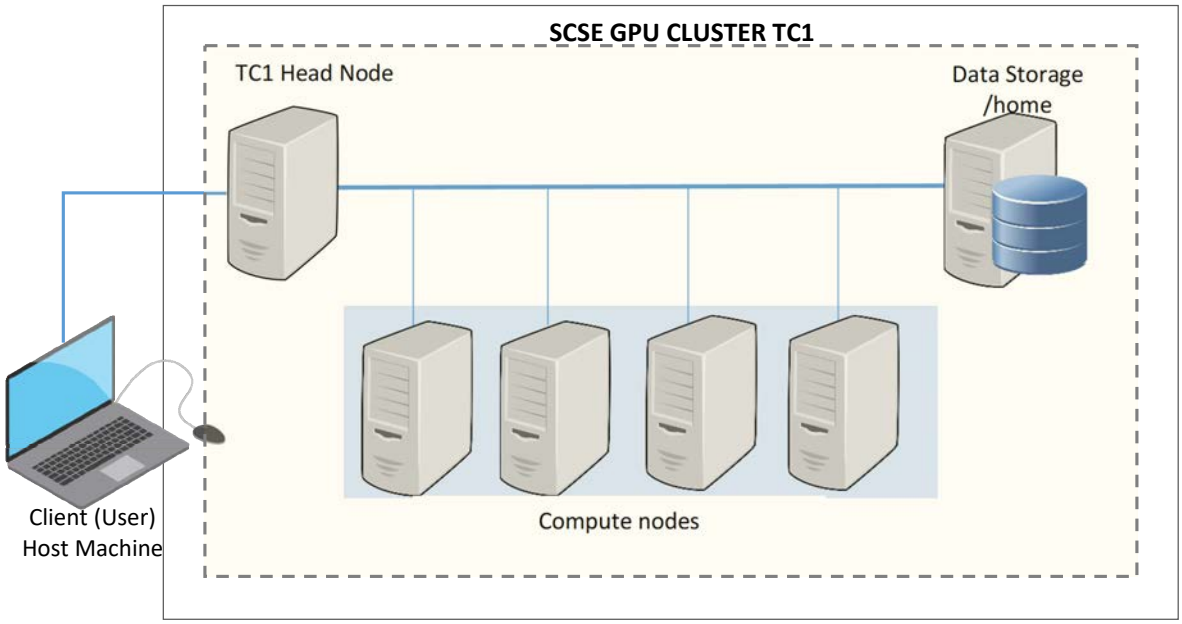
SCSE GPU CLUSTER TC1 USER GUIDE

Introduction



In SCSE GPU cluster TC1, there are two diverse types of nodes for diverse types of tasks.
The initial node you log in to is called the **Head Node**, serving as the main access point for the cluster.

The GPU cards for computation are in those Compute Nodes, NOT in the Head Node.
The users are **not allowed** to compile and run code on the Head Node. The users must create the job script to submit the computation request (knowns as **non-interactive job**) to SLURM (**Simple Linux Utility for Resource Management**) job scheduler, for the system to process in the allocated GPU Compute Nodes.

The available resources for the user to utilise for GPU computation is limited by the assigned **QoS** (SLURM-Quality of Service).

Client Tools for accessing the SCSE GPU Cluster Head Node

Name of Program	Description & Purpose	Where to download
PuTTY	**[SSH Client for Windows]** **PuTTY** is an open-source software as SSH and Telnet Client for *Windows Platform*. **SSH** (Secure Socket Shell) is a network protocol that gives users, particularly system administrators, a secure way to access a computer over an unsecured network.	The installer is available online and can be easily located using any search engine. **Tip:** Download the portable edition "putty.exe" – no need to install, ready for use *http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html* **Other Online References:** *https://www.ssh.com/academy/ssh/putty*
	[How to SSH on MAC] **Online References:** *https://winonmacs.com/* *https://www.servermania.com/kb/articles/ssh-mac/*	

WinSCP	**[SFTP Client for Windows]** **WinSCP** (Windows Secure Copy) is a free and open-source SFTP, FTP, WebDAV and SCP client for Microsoft Windows **SFTP** (SSH File Transfer Protocol) is an encrypted or secure file transfer protocol.	The official download site for WinSCP: *https://winscp.net/eng/index.php* **Installation tip:** Select "*Explorer*" for user interface style **Other Online References:** *https://winscp.net/eng/docs/guide_install* *https://www.puttygen.com/winscp*
FileZilla	**[SFTP Client for MAC]**	Download Site for FileZilla: *https://filezilla-project.org/index.php* **Other online References of using SFTP on MAC:** *https://lemp.io/how-to-connect-to-sftp-mac-os/* *https://beebom.com/how-to-use-mac-terminal-ftp-sftp-client/* *https://tipsmake.com/use-terminal-on-mac-as-ftp-or-sftp-client*

Workflow for the user:

1. Establish your first access via SSH. Your home directory will only be created after your first login

2. Uploading all the necessary job script, coding, dataset from your host machine to the cluster's storage (your home directory) via SFTP client

3. Setup and load your Conda environment(s) with required applications for computation

4. Submitting the non-interactive job using SLURM scheduler

5. Copy the job output back to your host machine, via SFTP client

This guide assumes that you have basic knowledge of accessing the Linux system via command-line interface:

- Connecting to a Linux server via SSH
- Moving files to a Linux server from your local computer, and vice versa
- The Linux basic commands such as: ls, cd, pwd, cp, mv, rm, chmod etc.

References for Basic Linux Commands:
https://serverdale.com/en/linux-commands
https://centoshelp.org/resources/commands/linux-system-commands/
https://www.guru99.com/must-know-linux-commands.html

- The GPU cards for your computation are NOT in the Head Node. So, do not attempt to execute command to acquire the GPU information in the Head Node e.g. "*nvidia-smi*", "*lspci | grep -E 'VGA|3D'*" and etc.

Logging into the cluster

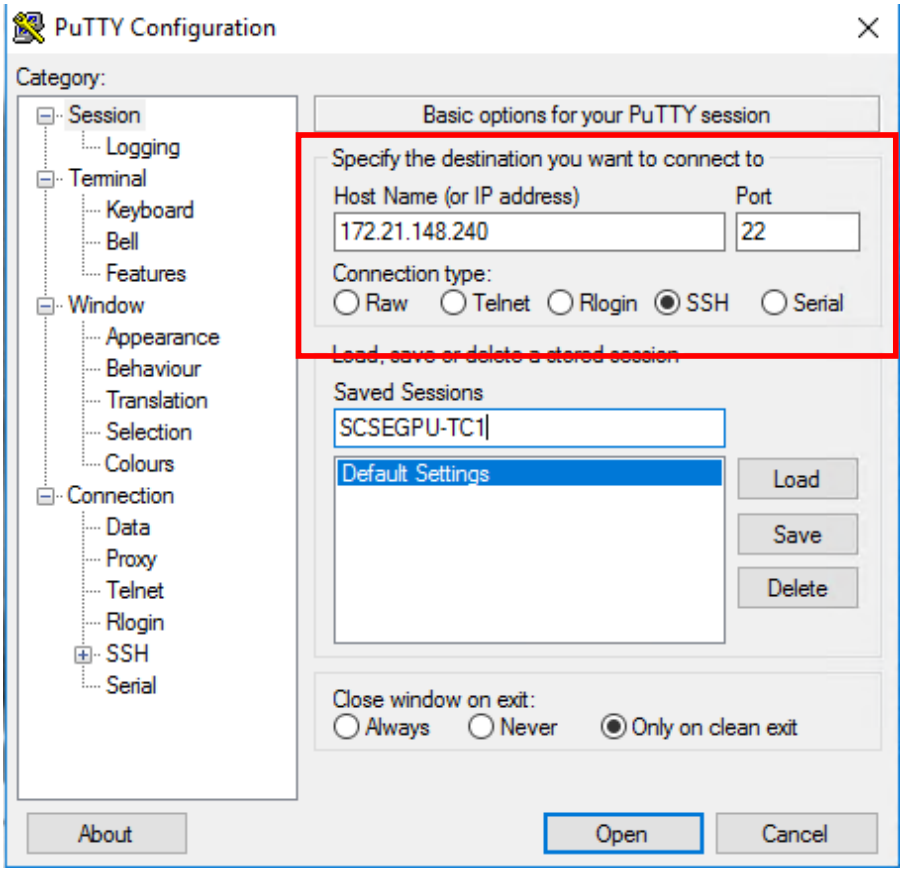
The SCSE GPU cluster is only accessible **within NTU network**
For off-campus access, the user must login to NTU Virtual Private Network (VPN) *[https://ntuvpn.ntu.edu.sg]*

Hostname of SCSE GPU Cluster Head Node	**scsegpu-tc1**
IP Address	**172.21.148.240**
Login Credential	Your NTU Network Account ID ***(in Lower Case)*** & Password

SSH via PuTTY

1. Launch PuTTY.

   Under the Host Name, enter the IP Address [172.21.148.240]
   Ensure that Port is 22.

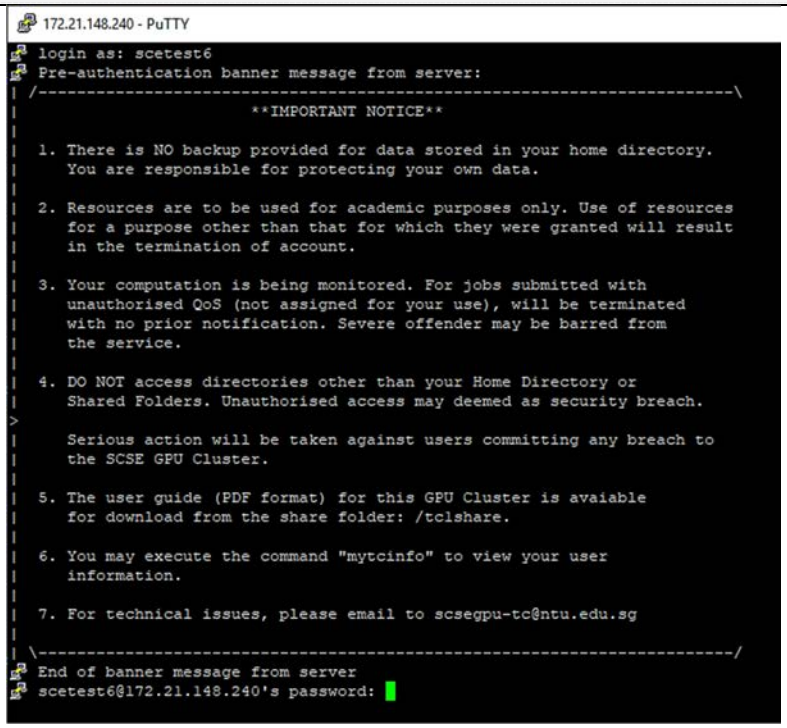


2. Examples of command-line access via your LINUX or MAC Terminal:

   >> *ssh -l <Your Username in Lower Case> 172.21.148.240*

   >> *ssh -p 22 <Your Username in Lower Case>@172.21.148.240*

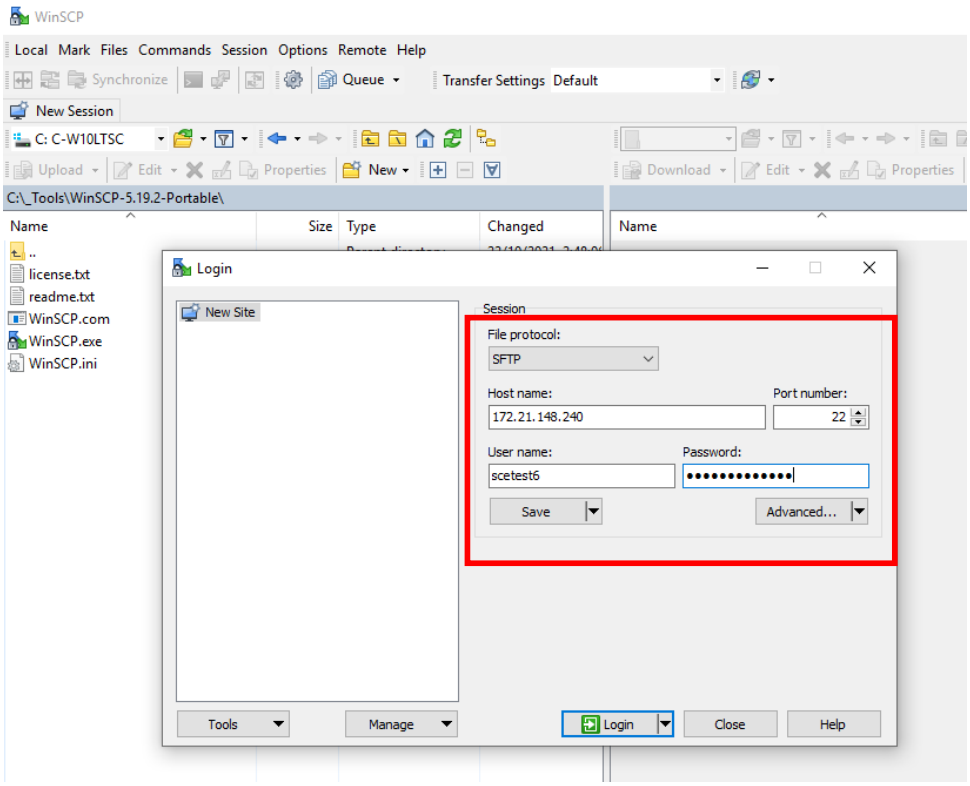
3. On your first login, you will be asked to accept the host key.
   Do click "**yes**" to continue, and you should get a terminal window, where you will be prompted for your credentials. Log in with your NTU Network Account ID (*in Lower Case*) and password.



Best practice to exist the SSH session: type "**exit**" or press the keys **Ctrl + D**

SFTP via WinSCP/FileZilla – For file transfer between your local host machine and GPU cluster

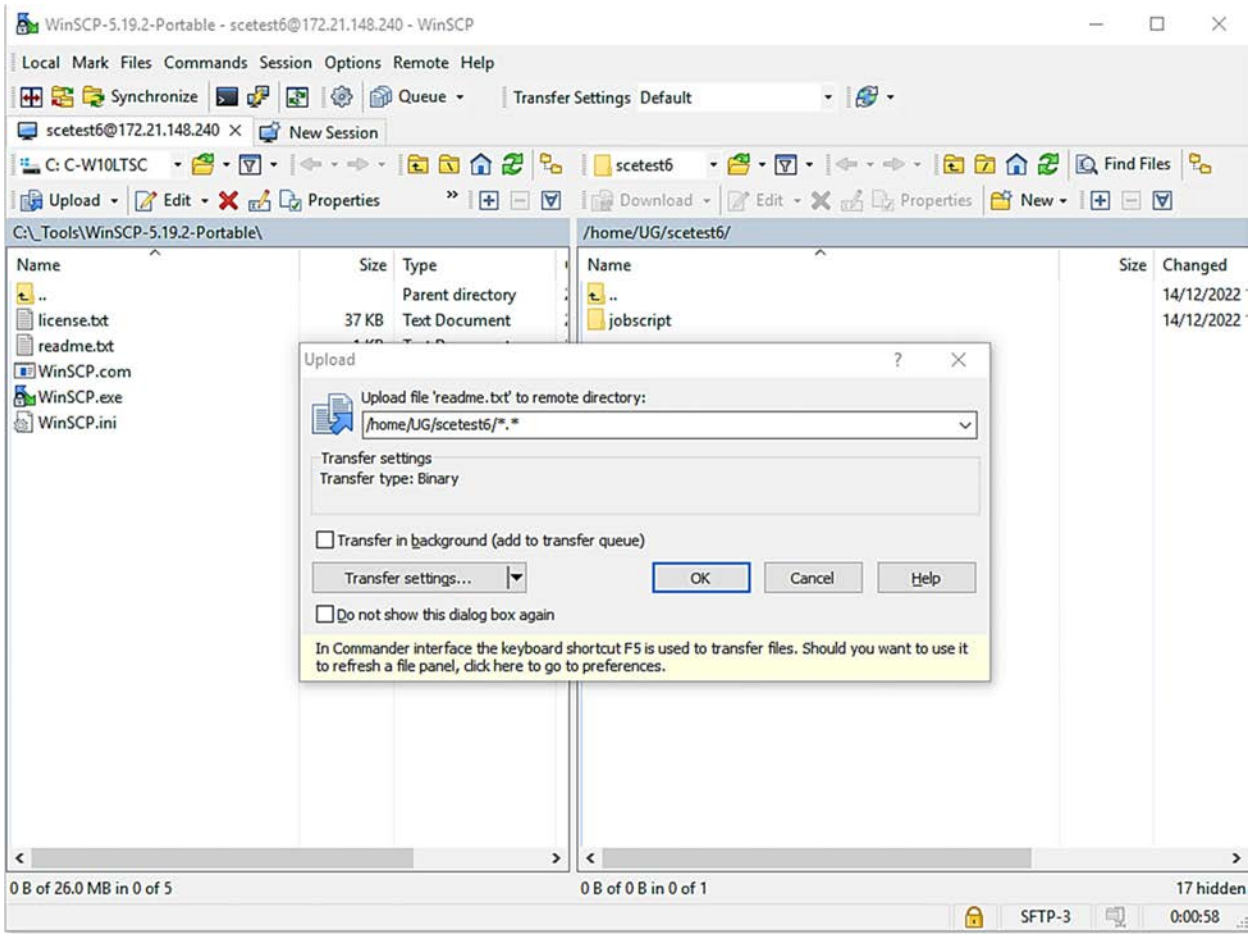
1. In this example, the client tool using for file transfer is WinSCP.
2. Launch WinSCP. Check that the File Protocol is set to SFTP.
3. Under the Host Name, key in the IP Address [172.21.148.240]. Ensure that Port is 22. Key in your NTU Network Account ID (in lower case) and password for the User name and Password field.



4. Ensure that you are in the correct directory before transferring your files over.
   You can verify the pathname of your home directory with following command in SSH session:
   >> *pwd*

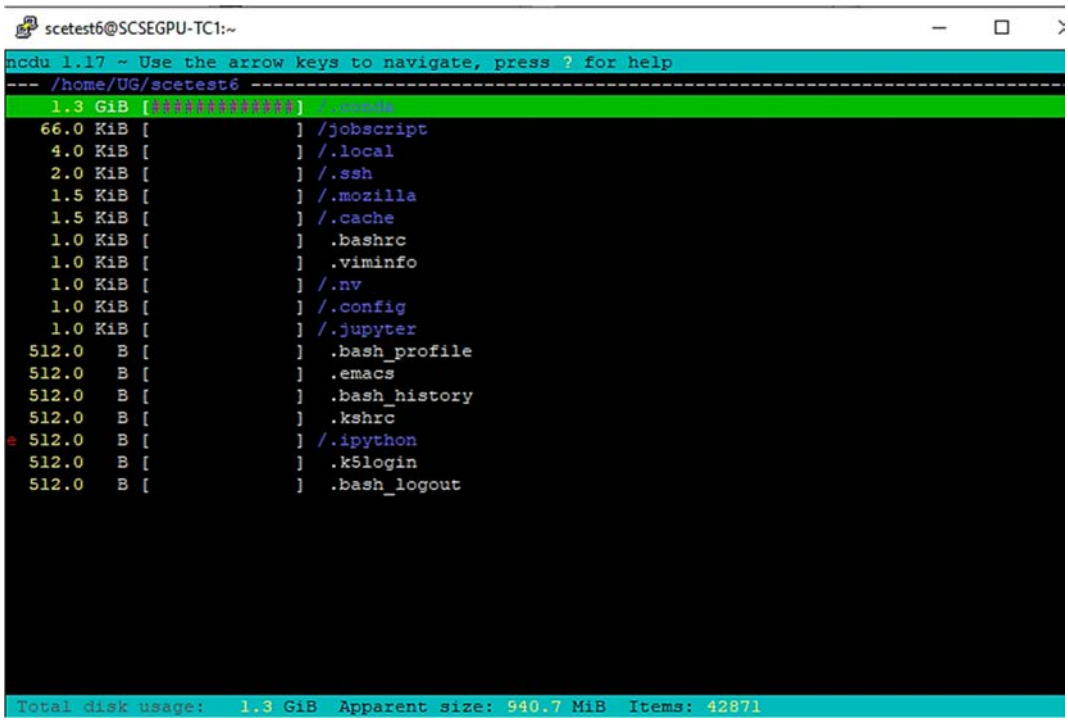
```
[scetest6@SCSEGPU-TC1:~]# pwd
/home/UG/scetest6
[scetest6@SCSEGPU-TC1:~]#
```

Select the file that you want transfer and press the key "***F5***"
The *Upload-Window* will pop out to transfer the file from your host machine to your home directory.
The *Download-Window* will pop out to transfer file from your home directory to selected local disk in your host machine.



Maintain Data in Your Directory

1. Your data and computation are bounded within your assigned home directory and share folder (if any).
2. All home directories are set with disk quota limitation. The student users may have disk quota of 100GB, depending on their usage assignment.
3. The user may encounter following issues when the quota usage reached more than 98%:
   - Unable to login
   - Unable to compute data
   - Unable to save in new data, or having data lost
4. You should do regular housekeeping, ensure there is sufficient free space for your computation.
5. Command to verify the disk usage at command line: ***ncdu***



Press the key "***q***" to abort disk scanning or exit.

Reference on ncdu: *https://ostechnix.com/check-disk-space-usage-linux-using-ncdu/*

6. Best practices to ensure enough disk space for your computation:
   - Regularly check on the disk usage for your home directory
   - Transfer and backup your data to your personal device
   - Remove unwanted data in your home directory
   - *Do not remove those system directories and files, which naming with a "**.**" in front, such as ".bash_profile", ".bashrc", ".config" etc*
   - The purpose of the system folder "*.conda*" is to store the packages and environments that you have setup and installed for your computation. *Remove those unwanted environments to free the disk space.*

Conda Package and Environment

The module package tool is available on the GPU Cluster, allowing users to easily configure their environment based on the application needed. As this cluster is shared among students of different courses in SCSE, there may be some applications that are not relevant to you.

To view the available share applications, apply the modules under ***/cm/shared/modulefiles***

>> module avail

```
[scetest6@SCSEGPU-TC1:~]# module avail
------------------------------------------------- /cm/local/modulefiles --------------------------------------------------
boost/1.71.0        cmd     cmsh            gcc/8.4.0  luajit       openldap  python37
cluster-tools/9.0   cmjob   freeipmi/1.6.4  gcc/9.2.0  module-info  python3   shared

------------------------------------------------- /cm/shared/modulefiles -------------------------------------------------
anaconda                           fftw3/openmpi/gcc/64/3.3.8  intel-tbb-oss/ia32/2020.3     python/3.7.12
blacs/openmpi/gcc/64/1.1patch03    gcc/8.5.0                   intel-tbb-oss/intel64/2020.3  python/3.9.13
blas/gcc/64/3.8.0                  gcc/10.4.0                  iozone/3_487                  python/3.10.5
cuda/10.2                          gdb/8.3.1                   lapack/gcc/64/3.8.0           R
cuda/11.7                          go                          matlab/R2020a                 scalapack/openmpi/gcc/2.1.0
default-environment                hdf5_18/1.8.21              miniconda-py37                singularity
fftw2/openmpi/gcc/64/double/2.1.5  hpl/2.3                     miniconda-py39
fftw2/openmpi/gcc/64/float/2.1.5   hwloc/1.11.11               openmpi/gcc/64/1.10.7
[scetest6@SCSEGPU-TC1:~]#
```

To view the description of a module:

>> module show <module_name>

To load the selected module for operation:

>> module load <module_name>

To list loaded module:

>> module list

To unload the selected module:

>> module unload <module_name>

To unload all loaded modules:

>> module purge

To know more about the command "module":

>> module --help

Setup Your own Conda Environment

In this GPU Cluster, the user has no right to execute Sudo and install application to the system folders.

You may download and install the Anaconda or Miniconda in your home directory for operation.

Reference links:
https://www.anaconda.com/products/distribution
https://docs.conda.io/en/latest/miniconda.html

OR use the available Anaconda in this GPU Cluster:

To load the module of **anaconda**:

>> module load anaconda

To view the information of the loaded **anaconda**:

>> conda info

The Conda commands can only run after the module is loaded. By default, the read-only base environment is activated. There are ready packages in the centralised Anaconda (or the base environment), you may issue the following command to view the list:

>> conda list

You are **not allowed** to install any packages in the base environment. If you find those packages in the base environment do not meet your requirement, you may setup own conda environment in the home directory, and install the require packages for your operation.

For first-time running Conda, the system may prompt you to execute the command "***conda init <shell>***".

The purpose of the command is to write some shell code into the system file (*e.g. ~/.bashrc*) as startup script for conda. For default shell using bash, execute the following

>> conda init bash

For the change to take effect, execute

>> source .bashrc

Commands to manage the Conda environment:

Description	Examples of Command Execution
To create a new Conda environment naming as "*TestEnv*"	***>> conda create -n TestEnv***
Activate the environment "*TestEnv*"	***>> conda activate TestEnv***
To exit the environment "*TestEnv*"	***>> conda deactivate***
List the environment created in your home directory	***>> conda env list***
To export the configuration of an environment into a *.yml* file.	***>> conda env export > TestEnv.yml***
To create the environment from the existing *.yml* file	***>> conda env create -f TestEnv.yml***
To update the content of an existing *.yml* file, with option "*prune*" to remove the outdated configuration and dependencies	***>> conda env update --prefix ./TestEnv --file TestEnv.yml --prune***
To remove the environment "*TestEnv*"	***>> conda env remove -n TestEnv***

Reference on managing the environment:
https://docs.conda.io/projects/conda/en/latest/user-guide/tasks/manage-environments.html#creating-an-environment-with-commands

Commands to manage the variables in the environment:

Description	Examples of Command Execution
List the variables in the environment	***>> conda env config vars list***
To set a new variable in the environment "*my_var1*"	***>> conda env config vars set my_var1=value***
To unset (remove) the variable "*my_var1*"	***>> conda env config vars unset my_var1***
Always re-activate the environment after adding a new variable and removing a variable	***>> conda activate TestEnv***

```
[scetest6@SCSEGPU-TC1]# conda activate TestEnv
(TestEnv) [scetest6@SCSEGPU-TC1]# conda env config vars list
(TestEnv) [scetest6@SCSEGPU-TC1]# conda env config vars set my_var1=10
To make your changes take effect please reactivate your environment
(TestEnv) [scetest6@SCSEGPU-TC1]# conda activate TestEnv
(TestEnv) [scetest6@SCSEGPU-TC1]# conda env config vars list
my_var1 = 10
(TestEnv) [scetest6@SCSEGPU-TC1]# conda env config vars unset my_var1
To make your changes take effect please reactivate your environment
(TestEnv) [scetest6@SCSEGPU-TC1]# conda activate TestEnv
(TestEnv) [scetest6@SCSEGPU-TC1]# conda env config vars list
(TestEnv) [scetest6@SCSEGPU-TC1]#
```

Commands to manage the packages in the environment

The user must always create a new environment to install the package for own use.

Description	Examples of Command Execution
Search for available packages. For this example, search for available python to install	***>> conda search python*** ***>> conda search python=3.9***
To search and install available package from third-party channel The conda package manager usually installs the package from the official default channels. You must specify the channel in the command if to search and install package from third-party channel ***conda-forge*** is one of the third-party channels, providing latest conda packages. **Reference links:** *https://docs.conda.io/projects/conda/en/latest/user-guide/concepts/channels.html* *https://ostechnix.com/enable-conda-forge-channel-for-conda-package-manager/*	# To search for a package in 3rd party channel "conda-forge" ***>> conda search -c conda-forge <name of package>*** # To install the package from the selected third-party channel ***>> conda install -c conda-forge <name of package>***
List the installed packages	***>> conda list***
Install selected version of the package	***>> conda install python=3.9.7***

The GPU Compute Nodes are ready with their own version of Cuda by NVIDIA, which may not be up-to-date or applicable for user's operation. The user may setup own Conda Environment, and install the desired version of Cuda Toolkit for operation.
 	
Reference links:
https://docs.nvidia.com/deploy/cuda-compatibility/index.html
https://anaconda.org/nvidia/cudatoolkit
	`[scetest6@SCSEGPU-TC1]# module avail | grep cuda`
	`anaconda cuda-11.7`
	`cuda-10.2 fftw3/openmpi/gcc/64/3.3.8`
	# To apply Cuda available in the Nodes, add the module loading command in the job script
	>> module load cuda-11.7
	
	# Search for available version of Cuda Toolkit
	>> conda search cudatoolkit
	>> conda search -c nvidia cudatoolkit
	
	# Then install the select version for operation
	>> conda install cudatoolkit=<version>
	>> conda install -c nvidia cuda-toolkit=<version>

Remove a package	***>> conda uninstall python=3.9***

Reference links on package installation in conda:
https://docs.anaconda.com/anaconda/user-guide/tasks/install-packages/
https://docs.conda.io/projects/conda/en/latest/user-guide/tasks/manage-pkgs.html#installing-packages

IMPORTANT NOTE:
Advise to install the packages using ***conda install***. Only use ***pip install*** if the required package is not available in conda.
This is to prevent package incompatibility.

If you want your conda environment to be loaded each time you start your SSH session, you may append "module load anaconda" and "conda activate yourenvname" at the end of the system file "***.bashrc***", located at your home directory.

```
# .bashrc

# Source global definitions
if [ -f /etc/bashrc ]; then
        . /etc/bashrc
fi

# Uncommon the following line if you don't like systemctl's auto-paging feature
:
# export SYSTEMD_PAGER=

# User specific aliases and functions
module load anaconda
conda activate TestEnv
```

Remove unwanted Conda environment, to free disk space

1. Exit from the conda environment >> *conda deactivate*
2. List the conda environment in your home directory >> *conda env list*
3. Only remove unwanted environment in your home directory>> *conda env remove –name <Name of ENV to remove>*

```
[scetest6@SCSEGPU-TC1]# conda env list
# conda environments:
#
base                  *  /apps/anaconda3
TC                       /home/MSAI/scetest6/.conda/envs/TC
TestEnv                  /home/MSAI/scetest6/.conda/envs/TestEnv
envtest                  /home/MSAI/scetest6/.conda/envs/envtest
mm2lab                   /home/MSAI/scetest6/.conda/envs/mm2lab
mmlab                    /home/MSAI/scetest6/.conda/envs/mmlab
py38-tfgpu               /home/MSAI/scetest6/.conda/envs/py38-tfgpu

[scetest6@SCSEGPU-TC1]# conda env remove --name envtest

Remove all packages in environment /home/MSAI/scetest6/.conda/envs/envtest:
```

SLURM User Guide

SLURM (*Simple Linux Utility for Resource Management*) is a software package for submitting, scheduling, and monitoring jobs on large computer clusters.

In this GPU cluster, you are **not allowed** to compile and run your code on the head node. You must create a job script to submit your computation request (**non-interactive jobs**) to SLURM Scheduler, for the system to process in allocated GPU Compute Nodes.

The resources for GPU computation is limited by the **QoS** (SLURM-Quality of Service) assigned to you.

Command to view the QoS assigned to the user
>> sacctmgr show user <username> withassoc format=user,qos

 # You may copy and paste the commands to the command prompt in your SSH session window for execution
 # Replace the text "<username>" with your own username

```
[scetest6@SCSEGPU-TC1]# sacctmgr show user scetest6 withassoc format=user,qos
      User                  QOS
---------- --------------------
  scetest6               normal
[scetest6@SCSEGPU-TC1]#
```

 # The above example shown the user "scetest6" can use the assigned QoS "normal" for GPU computation

---

Command to show the resources configured for the QoS:
>> sacctmgr -P show qos <Name of QoS> withassoc format=name,MaxTRESPU,MaxJobsPU,MaxWall
>> sacctmgr show qos <Name of QoS> withassoc format=name%+15,MaxTRESPU%+40,MaxJobsPU%+10,MaxWall%+10

 # %+<value> is to add value to expand the viewing field

```
[scetest6@SCSEGPU-TC1:~]# sacctmgr -P show qos normal withassoc format=name,MaxTRESPU,MaxJobsPU,MaxWall
Name|MaxTRESPU|MaxJobsPU|MaxWall
normal|cpu=20,gres/gpu=1,mem=64G|2|06:00:00
[scetest6@SCSEGPU-TC1:~]# sacctmgr show qos normal withassoc format=name%+15,MaxTRESPU%+40,MaxJobsPU%+10,MaxWall%+10
           Name                                  MaxTRESPU  MaxJobsPU    MaxWall
--------------- ---------------------------------------- ---------- -----------
         normal                cpu=20,gres/gpu=1,mem=64G          2    06:00:00
[scetest6@SCSEGPU-TC1:~]#
```

Example showing the Resources assigned to QoS name "**normal**":

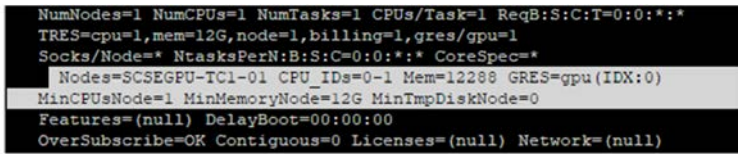
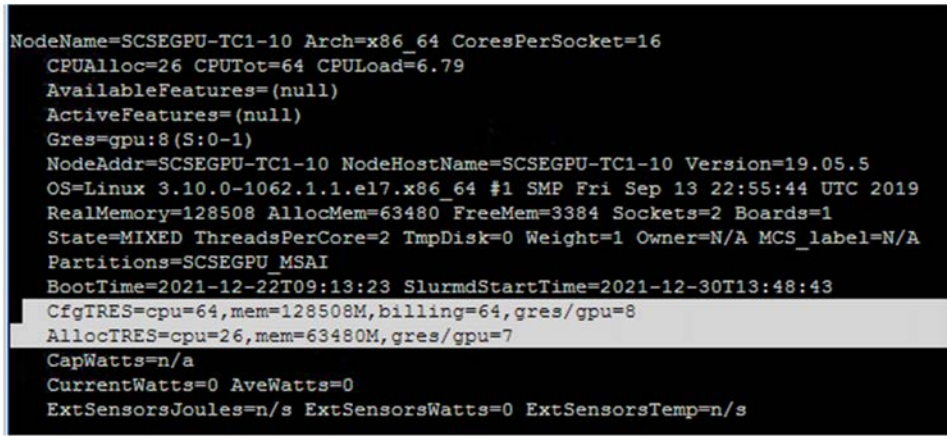
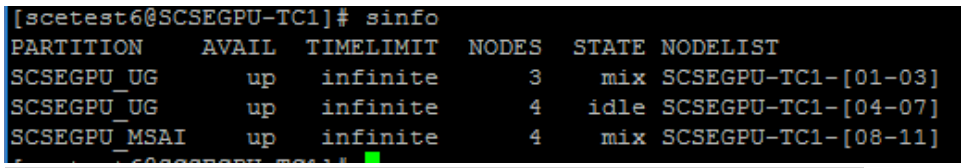
1. MaxTRESPU (*Maximum Trackable RESources Per User*):
   cpu (*Maximum number of CPUs/CORES for the user to deploy*) = 20
   gres/gpu (*Maximum number of GPU for the user to deploy*) = 1
   mem (*Maximum size of Memory for the user to deploy*) = 64G
2. MaxJobsPU (*Maximum number of Jobs Per User, can run at a given time*) = 2
3. MaxWall (*Maximum Wall clock time per user, to run the job, DD-HH:MM:SS*) = 6-hours

---

The SLURM job script is required, to specify the necessary resources, application, and path to execute your code. Then, submit the job script to SLURM at command line.

References links on creating the SLURM job script:
https://slurm.schedmd.com/quickstart.html
https://www.carc.usc.edu/user-information/user-guides/hpc-basics/slurm-templates
https://svante.mit.edu/use_slurm.html

Common SLURM commands for you to manage the job submission :

Command	Definition (#) & Use Examples (>>)
scontrol	*# View the details of a running job, based on its jobid and showing the id of GPU card in the node running the job, under the line "Nodes=SCSEGPU…"* **>> scontrol show -d jobid <jobid>**  *# View the status of the nodes in the Partition "SCSEGPU_UG"* **>> scontrol show node SCSEGPU-TC1-[01-07]**  *You can reference the resource status of the node, based on the information at the lines "CfgTRES" and "AllocTRES". The values shown under "AllocTRES" are the resources being allocated to the running jobs in the node.*
sinfo	*# Display state of the compute nodes in respective partition* **>> sinfo**  *For STATE not showing "idle" or "mix", may indicate the node under NODELIST has been down*
sacct	*# To view your own job history* **>> sacct --format=jobid,jobname%+15,qos,alloctres%+50,nodelist,start,elapsed,state,reason%+20** *# To view all job history for the day, option "-a"* **>> sacct -a --format=user,jobid,jobname%+15,qos,alloctres%+50,nodelist,start,elapsed,state,reason%+20** *# Job viewing, excluding those lines with "batch..."* **>> sacct -a --format=user,jobid,jobname%+15,qos,alloctres%+50,nodelist,start,elapsed,state,reason%+20 | grep -wv batch** *# List the jobs executed by specific user <username>* **>> sacct -u <username>**
sbatch	*# Submit a job script to the job queue* **>> sbatch job.sh**
scancel	*# The jobid can be obtained by the command "squeue"* *# Cancel a job base on its jobid* **>> scancel <jobid>**

squeue	*# Display the jobs in the job queue* **>> squeue** *# List more available information with option 'la'* **>> squeue -la** *# List the running jobs submitted by the user <username>* **>> squeue -u <username>**
seff	*# Slurm Job Efficiency Report* *# Display the statistics of resource being utilized by the completed job, allow to review on the resource assignment for the job* **>> seff <jobid of completed job>**

Guideline for SLURM job submission

1. **Only apply the QoS assigned to you**

   The job submitted with unauthorized QoS (QoS not assigned for your use) will be terminated by the system with no prior notification.

2. **Apply the QoS resources within the limit in your job script**

   You may use customized script/command to view your account and QoS Info

   >> mytcinfo

```
[scetest6@SCSEGPU-TC1:~]# mytcinfo
-------User Info of scetest6 in TC Cluster----------------------
uid=30902(scetest6) gid=30902(scetest6) groups=30902(scetest6),17184921(tclusers)
Home Directory = /home/UG/scetest6
========User Info of scetest6 in SLURM DB=======================================
          User              Cluster                              QOS
-------------- ------------------- ------------------------------
       scetest6    scse-gpu-cluster1                           normal

===QOS Info=====================================================================
          Name                        MaxTRESPU       MaxJobsPU         MaxWall
-------------- ------------------------------ --------------- ---------------
        normal      cpu=20,gres/gpu=1,mem=64G               2        06:00:00
     Partition
--------------
 SCSEGPU_UG

------End-------------------------------------------------------
```

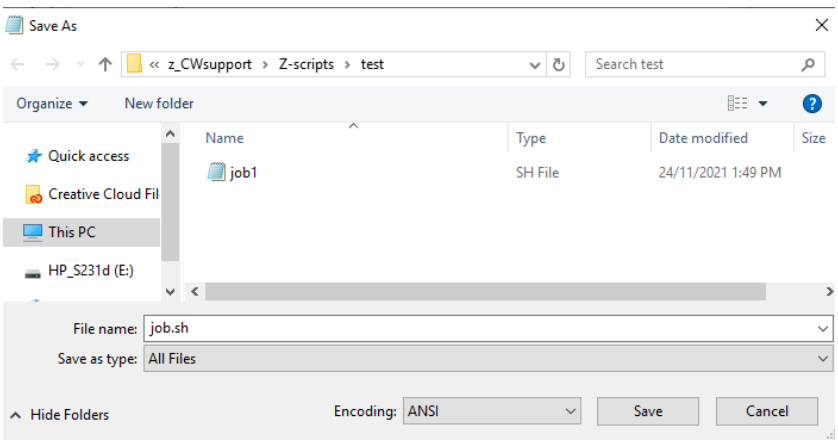
   In your job script, ensure to apply the job flag with stated value below or equal to (**<=**) the amount in the assigned QoS.

MaxJobsPU	The maximum of number of jobs allow to run at a time **<= 2** *You may submit two jobs to run at a time*
MaxWall	Total computation of your running jobs must not exceed the maximum wall time **<= 6-hr**
MaxTRESPU	Total amount of resources allocated in your two job scripts must not exceed the maximum values. *For example: No of CPU/cores <=20, Memory <=64GB and No of GPU Card=1, as stated in QoS assigned.*

   # Other customized scripts/commands are available under /tc1share/tc-scripts #

3. **Create/Edit your SLURM job script**

   You may use any text editor available at your host machine, to create the job script, ensuring the job script is saved with file extension "***.sh***". Then, transfer to your home directory in the GPU Cluster to execute.



   In the Head Node, you may use those linux command line editors to edit your job script: ***vi***, ***nano*** or ***ne***

   Some references links on Linux Command-line Editors:

   https://www.guru99.com/the-vi-editor.html

   https://www.geeksforgeeks.org/nano-text-editor-in-linux/

The following are the basic job flags to apply in the job script:

Resource	Flag Syntax	Description
Partition*	--partition=SCSEGPU_UG	# Specify the node group for your job execution, defines as *Partition* # Command to list the available Partitions and allowed QoS in the Cluster ***>> scontrol show partition***
QoS*	--qos=normal	# Specify the QoS to apply for the job # The QoS must be one of them allow to execute in the Partition
nodes*	*--nodes=1	# Number of compute nodes for the job # Max node allows per job: 1
	--nodelist=<hostname>	# ***Optional***: you may specify the node (hostname) to run the job # Command to list the idle nodes and their hostname in the Partition ***>> sinfo -N <Name of Partition> | grep idle*** ***>> sinfo -N SCSEGPU_UG | grep idle***
GPU*	--gres=gpu:1	# Specify the use of GPUs on compute nodes. # The number of GPU card to use is corresponding to the assigned QoS # This option must be present in the job script, for the system to deploy GPU resource
Memory*	--mem=8000M Or --mem=8G	# Specify on the memory to apply for the job. # The value is corresponding to the assigned QOS, must not exceed the assigned maximum memory size. # This option must be present in the job script, for the system to deploy the memory for the computation.
CPUs/cores	--ntasks-per-node=2	# ***Optional***: only add this flag if want to apply more than one core for the computation # Specify the number of "tasks" (cores) per node, for use with distributed parallelism. *Default value=1*
	--cpus-per-tasks=2	# ***Optional***: only add this flag if want to apply more than one core for the computation # Specify the number of CPU-cores to allocate to per task, for use with shared memory parallelism. *Default value=1*
Job name*	--job-name=MyJob	# Name of job
Output file*	--output=output_%j.out Or --output=output_%x_%j.out	# State the name of the file for standard output ***Filename patterns:*** ***%x: job name*** ***%j: job id, generated by SLRUM***
Error file*	--error=error_%j.err Or --error=error_%x_%j.err	# State the name of the file for error log – if any
time	--time=01:00:00	# Specify a time limit for the job – hh:mm:ss # The max timing is corresponded to *MaxWall* in assigned QOS # Optional, only input if requires limiting the job timing

*The flags with * must specify in the job script*

The following is an example of a standard job script: job.sh

```
#!/bin/bash
#SBATCH --partition=SCSEGPU_UG
#SBATCH --qos=normal
#SBATCH --nodes=1
#SBATCH --gres=gpu:1
#SBATCH --mem=8G
#SBATCH --job-name=MyJob
#SBATCH --output=output_%x_%j.out
#SBATCH --error=error_%x_%j.err


module load anaconda
source activate TestEnv
python test.py
```

***Remarks**:*
You may add necessary flags from the table above.
Do note that if you exceed the limit for your QOS, the job will not run.
From this job script, 2 files will be created:
output_%x_%j.out
- *The standard output from running the code will be saved here*

error_%x_%j.err
- *Error log from job if any*

Load necessary modules needed to run the code
Activate your conda environment (if any)
Run your code

4. Submit your job script

Action to submit a job	Command
To submit your job script, where job.sh is the name of your job script	**sbatch job.sh**

The system should respond with a job ID:

```
[scetest6@SCSEGPU-TC1 sample-1]$ sbatch job.sh
Submitted batch job 34483
```

5. Avoid using the command "*srun*" to submit job

The command "***srun***" is to submit job at the command line for ***real time execution***. You have to maintain your SSH session until the whole process completed. The disconnection of your SSH session may kill the process and causing you to lose the control over the execution. Thus, for the jobs requiring more than an hour to compute, are advised to submit using the command "***sbatch***".

The command "***sbatch***" is to submit job for later execution, handling by SLURM in the background.
Once submitted the job using "***sbatch***", you may exist from your SSH session, with no consequence.

To avoid high volume of SSH connections to the Cluster Head Node, all users are advised to use the command "***sbatch***" for job submission.

6. Verify on the job queue and node operational status

Command to verify on the job queue:
>> squeue

Command to verify on the resources being in used by the running jobs
For nodes under partition "SCSEGPU_UG" ***>> scontrol show node SCSEGPU-TC1-[01-07]***

 # Look at the overall resources being allocated for node operation and running jobs

```
NodeName=SCSEGPU-TC1-11 Arch=x86_64 CoresPerSocket=16
   CPUAlloc=22 CPUTot=64 CPULoad=34.48
   AvailableFeatures=(null)
   ActiveFeatures=(null)
   Gres=gpu:8(S:0-1)
   NodeAddr=SCSEGPU-TC1-11 NodeHostName=SCSEGPU-TC1-11 Version=19.05.5
   OS=Linux 3.10.0-1062.1.1.el7.x86_64 #1 SMP Fri Sep 13 22:55:44 UTC 2019
   RealMemory=128508 AllocMem=103488 FreeMem=75357 Sockets=2 Boards=1
   State=MIXED ThreadsPerCore=2 TmpDisk=0 Weight=1 Owner=N/A MCS_label=N/A
   Partitions=SCSEGPU_MSAI
   BootTime=2021-12-22T09:14:22 SlurmdStartTime=2021-12-30T13:48:43
   CfgTRES=cpu=64,mem=128508M,billing=64,gres/gpu=8
   AllocTRES=cpu=22,mem=103488M,gres/gpu=8
   CapWatts=n/a
   CurrentWatts=0 AveWatts=0
   ExtSensorsJoules=n/s ExtSensorsWatts=0 ExtSensorsTemp=n/s
```

- The node selected to run the job is based on the available resources (CPU, Memory, GPU and Compute nodes) at the time of submission
- For the jobs failed the assignment will go into PENDING queue
- The running jobs terminated due to node reset or failure, will also be appearing in the PENDING queue

 # How to spot failing node #

- *For node with state showing "DRAIN", indicating it has failed task. This node is alive but may not able to accept new job and the performance of existing running jobs may also be affected.*
- *Requiring the administrator to reset the affected node for operation. The impact of the reset is the existing running jobs in the affected node will be terminated. The affected jobs will have to join the PENDING queue for re-assignment.*

```
NodeName=SCSEGPU-TC1-11 Arch=x86_64 CoresPerSocket=16
   CPUAlloc=18 CPUTot=64 CPULoad=20.63
   AvailableFeatures=(null)
   ActiveFeatures=(null)
   Gres=gpu:8(S:0-1)
   NodeAddr=SCSEGPU-TC1-11 NodeHostName=SCSEGPU-TC1-11 Version=19.05.5
   OS=Linux 3.10.0-1062.1.1.el7.x86_64 #1 SMP Fri Sep 13 22:55:44 UTC 2019
   RealMemory=128508 AllocMem=89344 FreeMem=84255 Sockets=2 Boards=1
   State=MIXED+DRAIN ThreadsPerCore=2 TmpDisk=0 Weight=1 Owner=N/A MCS_label=N/A
   Partitions=SCSEGPU_MSAI
   BootTime=2021-12-22T09:14:22 SlurmdStartTime=2021-12-30T13:48:43
   CfgTRES=cpu=64,mem=128508M,billing=64,gres/gpu=8
   AllocTRES=cpu=18,mem=89344M,gres/gpu=6
   CapWatts=n/a
   CurrentWatts=0 AveWatts=0
   ExtSensorsJoules=n/s ExtSensorsWatts=0 ExtSensorsTemp=n/s
   Reason=Kill task failed [root@2022-04-19T14:17:45]
```

- Common reasons showing for such affected jobs in the PENDING queue: *Priority*, *Resources*
- The job will be executed when there is ready node and resource to assign within Time Limit (referenced to the maximum wall time *(MaxWall)* stated in the QoS)
- The pending job will be removed by the system when exceeded the Time Limit

Advices:

- If there are many jobs in the PENDING queue with reasons of "*Priority" or "Resources", please d*efer your job submission until those jobs are cleared in the queue

7. Process your submitted SLURM job

Actions to check on job status	Command
Display the jobs in the scheduling queue	**squeue -la**
Display job history for a user <username>	**sacct -u <username>**
Show the detail for a running/pending job <jobid>	**scontrol show jobid <jobid>**

If the Status indicates PD (PENDING) and NODELIST (REASON) indicates a reason in bracket, do note that the job has failed to run. Please cancel the job if this happens.

```
[scetest6@SCSEGPU-TC1 ~]$ squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
               263   SCSEGPU  TestJob scetest6 PD       0:00      1 (QOSMaxJobsPerUserLimit)
```

For submitted job failed the QoS limit check (*exceeded any of the values stated for the QoS*), will go into the PENDING queue

Common reasons for job associates to QoS failure:

- *QOSMaxJobsPerUserLimit*
  - Verify on your total number of jobs in Running queue, must not exceed the value stated in "**MaxJobsPU**"
  - Extra job will go into the PENDING queue
- *QOSMaxMemoryPerUser, QOSMaxCpuPerUserLimit, QOSMaxGRESPerUser*
  - The total resource count [**#SBATCH --mem=x**, **#SBATCH --ntasks-per-node=x**] stated in all your submitted job scripts using the same QoS, must not exceed the max value stated in "**MaxTRESPU**"
- *QOSMaxGRESPerUser*
  - This error may appear if unable to assign your jobs using the same QoS, to the allocated (1) GPU Card
  - For this case, restrict to submit only one (1) job to run at a time

The job will only be executed when passed the checks for QoS limit and node assignment within the Time Limit (referenced to **MaxWall** stated in the QoS)

The pending job will be removed by the system when reached the Time Limit.

Advice for such encounter:

- Cancel your pending jobs failed the QoS limit Check

- Review and modify the resource values in your job script
- Restrict your number of job submission

Action to cancel the job	Command
To cancel a job, where 263 is your job id	**scancel 263**

Once the job has finished running, it will be removed from the queue. You should be able to see the output file. If you did not specify a specific path for the output file in your SLURM script, it will be created in the directory where you submitted the SLURM job.

```
[scetest6@SCSEGPU-TC1 sample-1]$ ls
Error_MyJob_34483.err  job.sh  Output_MyJob_34483.out  test.py
[scetest6@SCSEGPU-TC1 sample-1]$
```

8. To know more about your completed jobs

You may use the customised script to view your job history for the day, *with details on applied QoS, allocated resources (cpu, gpu, memory assigned for the job), node assigned, job-submit time, start time, elapsed time, state and reason*

>> myjobhistory

```
[scetest6@SCSEGPU-TC1:~/jobscript]# myjobhistory
--List Running Jobs---------------------------------------------------
| User | jobid | jobname | qos | allocResources | start | elapsed | nodelist | state | reason
   scetest6      65897        runl     normal         billing=1,cpu=1,gres/gpu=1,mem=10G,node=1      2022-12-14T13:10:57   00:25:18  SCSEGPU-TC1-01  COMPLETED
   scetest6      65898        runl     normal         billing=1,cpu=1,gres/gpu=1,mem=10G,node=1      2022-12-14T13:42:48   00:21:12  SCSEGPU-TC1-01 CANCELLED+
```

Command to display the resource utilization statistics for your completed job, let you review and deploy the right number of resources (CPU/cores and Memory) for your computation

>> seff <job id>

```
[scetest6@SCSEGPU-TC1:~/jobscript]# seff 65897
Job ID: 65897
Cluster: scse-gpu-cluster1
User/Group: scetest6/scetest6
State: COMPLETED (exit code 0)
Cores: 1
CPU Utilized: 00:00:06
CPU Efficiency: 0.40% of 00:25:18 core-walltime
Job Wall-clock time: 00:25:18
Memory Utilized: 74.91 MB
Memory Efficiency: 0.73% of 10.00 GB
```

 # The above result shows applying x1 CPU is sufficient; can reduce the memory down to 1GB for computation #

Notice for Specific Coursework

For CE4042/CZ4042, there are ready shared conda environments created for your assignment:

/apps/conda_env/CZ4042_v2	/apps/conda_env/CZ4042_v3	/apps/conda_env/CZ4042_v4
Apps installed: Python 3.9 Tensorflow 2.6 Cudatoolkit 11.3	**Apps installed:** Python 3.8 Tensorflow 2.6 Cudatoolkit 11.3	**Apps installed:** Python 3.10 Tensorflow 2.9 Cudatoolkit 11.7
		Environment Variable *(For Tensorflow 2.9)* TF_ENABLE_ONEDNN_OPTS=0

To load Anaconda and activate your shared environment, simply type the following:

- module load anaconda
- conda activate /apps/conda_env/CZ4042_v4

```
[scetest6@SCSEGPU-TC1:~]# module load anaconda
[scetest6@SCSEGPU-TC1:~]# module list
Currently Loaded Modulefiles:
 1) anaconda
[scetest6@SCSEGPU-TC1:~]# conda activate /apps/conda_env/CZ4042_v4
(/apps/conda_env/CZ4042_v4) [scetest6@SCSEGPU-TC1:~]# python -V
Python 3.10.4
(/apps/conda_env/CZ4042_v4) [scetest6@SCSEGPU-TC1:~]# conda list | grep cudatoolkit
cudatoolkit               11.7.0              hd8887f6_10    nvidia
(/apps/conda_env/CZ4042_v4) [scetest6@SCSEGPU-TC1:~]# conda env config vars list
TF_ENABLE_ONEDNN_OPTS = 0
(/apps/conda_env/CZ4042_v4) [scetest6@SCSEGPU-TC1:~]# python -c 'import tensorflow as tf; print(tf.__version__)'
2.9.1
(/apps/conda_env/CZ4042_v4) [scetest6@SCSEGPU-TC1:~]#
```

Miscellaneous

SSH Tunneling

The GPU Cluster is only accessible via SSH connection. For the user requiring running *Jupyter Notebooks*, must learn to setup the SSH tunnel for computation.

SSH tunneling (also known as SSH port forwarding) is a method of creating an encrypted SSH connection between a client and a server through which services port can be relayed.

The following is an exemplary guide:
Running Jupyter Notebook on a computer node that needs to be contacted by the web browser on your local computer

1. **SSH to GPU Head Node and setup Jupyter Notebook in your home directory**

Description	Execution
Load the anaconda module	*# Execute Command* **>> module load anaconda**
Create and activate a conda environment	*# Execute Command* **>> conda create -n RunJupyter** **>> conda activate RunJupyter**
Install the required package for Jupyter Notebook from third-party channel "*conda-forge*"	*# Execute Command* **>> conda install -c conda-forge notebook** **>> conda install -c conda-forge nb_conda_kernels** **>> conda install -c conda-forge jupyter_contrib_nbextensions**
Deactivate the conda environment	*# Execute Command* **>> conda deactivate**

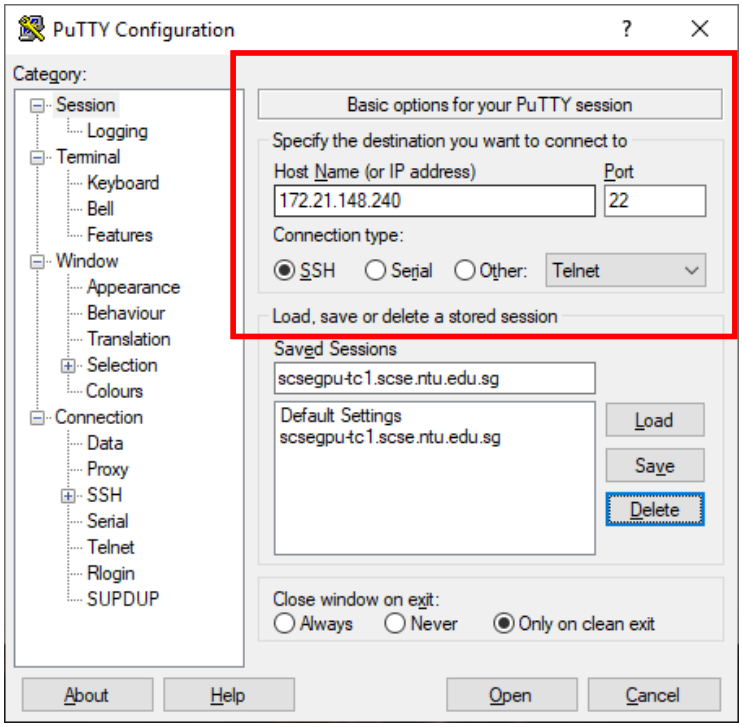
2. **Create the job script**

Description	Execution
Create the job script to run Jupyter Notebook. For this example, the job script is named as "*run1.sh*"	*# Sample of the job script* *#!/bin/sh* *#SBATCH --partition=SCSEGPU_UG* *#SBATCH --qos=normal* *#SBATCH --gres=gpu:1* *#SBATCH --nodes=1* *#SBATCH --mem=10G* *#SBATCH --job-name=run1* *#SBATCH --output=output_%x_%j.out* *#SBATCH --error=error_%x_%j.err* *module load anaconda* *source activate RunJupyter* *jupyter-notebook --ip=$(hostname -i) --port=8887*
Submit the job script	*# Execute Command* **>> sbatch run1.sh**

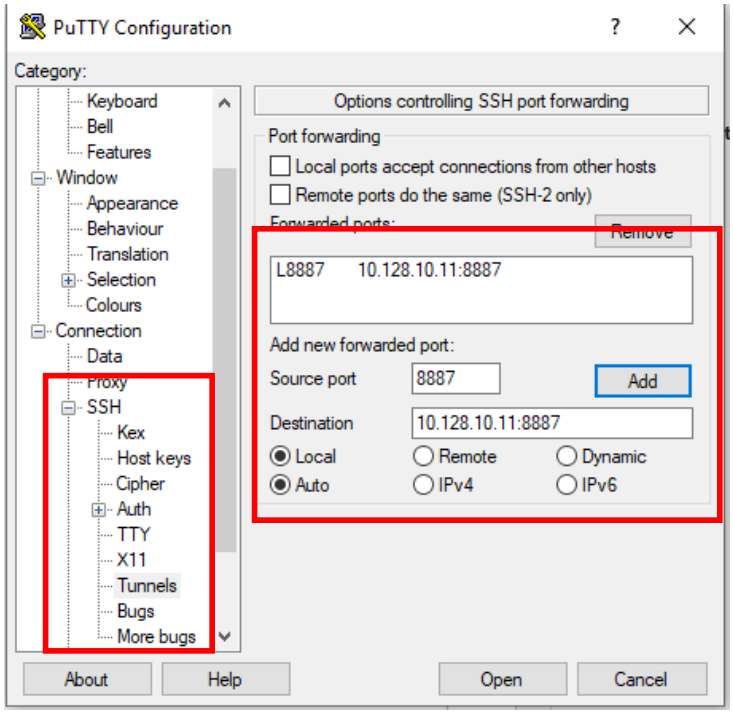
3. **Verify the access information in the error log.**
   Take note of the ***IP address***, ***port number*** and the ***URL*** for web access

```
[scetest6@SCSEGPU-TC1:~/jobscript]# ll
total 33
-rw-rw-r-- 1 scetest6 scetest6 1191 Dec 14 13:11 error_run1_65897.err
-rw-rw-r-- 1 scetest6 scetest6  277 Dec 14 13:11 output_run1_65897.out
-rw-rw-r-- 1 scetest6 scetest6  308 Dec 14 13:10 runj.sh
[scetest6@SCSEGPU-TC1:~/jobscript]# more output_run1_65897.out

[scetest6@SCSEGPU-TC1:~/jobscript]# more error_run1_65897.err
[I 13:11:04.187 NotebookApp] [nb_conda_kernels] enabled, 2 kernels found
[I 13:11:04.391 NotebookApp] Writing notebook server cookie secret to /home/UG/scetest6/.l
[I 13:11:05.506 NotebookApp] [jupyter_nbextensions_configurator] enabled 0.6.1
[I 13:11:05.509 NotebookApp] Serving notebooks from local directory: /home/UG/scetest6/job
[I 13:11:05.509 NotebookApp] Jupyter Notebook 6.5.2 is running at:
[I 13:11:05.509 NotebookApp] http://10.128.10.11:8887/?token=af4d500318edf22b5e314dfc066f7
[I 13:11:05.509 NotebookApp]  or http://127.0.0.1:8887/?token=af4d500318edf22b5e314dfc066f
[I 13:11:05.509 NotebookApp] Use Control-C to stop this server and shut down all kernels (
[C 13:11:05.530 NotebookApp]

    To access the notebook, open this file in a browser:
        file:///home/UG/scetest6/.local/share/jupyter/runtime/nbserver-45528-open.html
    Or copy and paste one of these URLs:
        http://10.128.10.11:8887/?token=af4d500318edf22b5e314dfc066f7fflcf880e474fbb1222
     or http://127.0.0.1:8887/?token=af4d500318edf22b5e314dfc066f7fflcf880e474fbb1222
```

4. **In your local machine, start another SSH session to SCSEGPU-TC1 Cluster:**



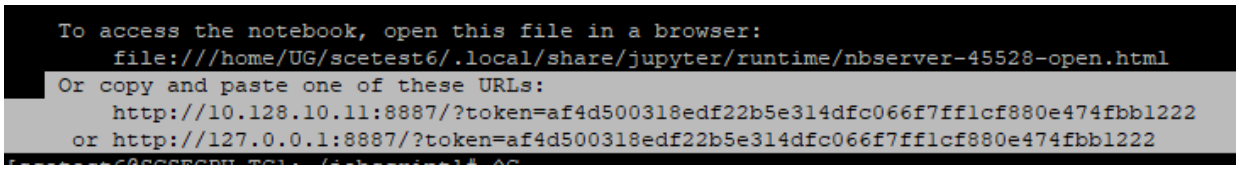
5. **Expand the category of "*SSH*" and select "*Tunnels*"**

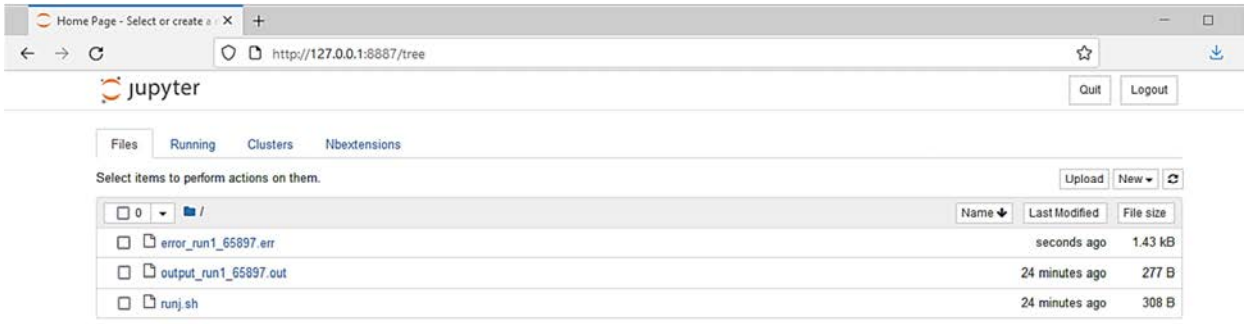


Input the IP address and port number obtained in the Error Log as show above.
*Click on the button "**Add**" to add into the Forwarded ports*

6. **Click on the button "*Open*" to start the SSH session. Login using your NTU username and password to establish the connection.**

7. Finally, you **may access the URL in your local machine** (Copy and paste one of the URLs to your browser for access)





8. The operation of Jupyter will terminate when you click on the button "***Quit***" in the web page. Then, you may logout from the SSH session set with the forwarded port.

Other Resources for application

QoS

The user may apply additional QoS for…

- longer computation hour, selecting range: 8-hr, 12-hr, 24-hr, till max 48-hr
- computation with more resource allocation: CPUs, Memory and GPU Card

Usage period: 1-month

Method to apply:

- Must provide the reason for the usage and verification (coding and computation result executed in the cluster) on the assigned QoS unable to support the user's operation.
- Send the request e-mail to SCSEGPU-TC-Support [*scsegpu-tc@ntu.edu.sg*]
- The administrator will evaluate, verify on the real requirement, and then assign the QoS accordingly
- The requestor will be notified, upon successful application

Additional Storage Space

The user may apply additional storage space for computation data more than 100GB
A share folder will be created in another storage to assign to the user.

Usage period: 4-months
Maximum usage quota: depending on the available free space and approval

Method to apply:

- Send the request e-mail to SCSEGPU-TC-Support [*scsegpu-tc@ntu.edu.sg*]
- Must provide the reason and verification for the usage request
- The user will be receiving the PFD form, via e-mail, to complete for the application
- The user will be notified of the resource assignment, upon successful application

Terms and Conditions

- The application is subjecting to resource availability, cluster operational workload and administration approval
- The usage of the assigned resource will be monitored
- The administrator claims the right to revoke the assigned resource if the usage has been verified underuse or impacting on the operation of the GPU Cluster
- The assigned resource will be removed upon usage expiry, with no prior notification

Important Notice

1. There is NO backup provided for data stored in your home directory. You are responsible for protecting and maintaining your own data.

2. Resources are to be used for academic purposes only. Use of resources for a purpose other than that for which they were granted will result in the termination of account.

3. Your computation is being monitored. For jobs submitted with Unauthorised QoS (NOT assigned for your use) and operation, will be terminated with no prior notification. Severe offender may be barred from the service.

4. DO NOT access directories other than your Home Directory or Shared folder. Unauthorised access to other directories or files, even for the purpose of "browsing", shall be deemed as a security breach. Serious action will be taken against users committing any breach to the SCSE GPU Cluster.

5. For issue regarding your coursework and project, please consult your Supervisor or TA (Teaching Assistant)

6. For technical issue, drop an email to SCSEGPU-TC-Support [*scsegpu-tc@ntu.edu.sg*]

~ End ~