Vanda System

Onboarding Session

NUS IT Research Computing

Notes

Vanda is hosted in STDCT, an experimental data center testbed. The research team may conduct experiments and briefly impact operational continuity.

Please report any issues through ntouch ticket, we will address it promptly

User Code of Conduct

By using the NUS IT HPC System, users acknowledge and agree to:

- Comply with all applicable regulations and policies, including the NUS Acceptable Use Policy, NUS Code of Conduct, and Singapore PDPA.
- Only upload and process non-confidential and non-sensitive data for NUS-related research.
- Refrain from any illegal, harmful, offensive, or unethical use of the system.

  Example: crypto-mining, scanning, reverse tunnelling, generating offensive materials, etc.

User Access

Prerequisites

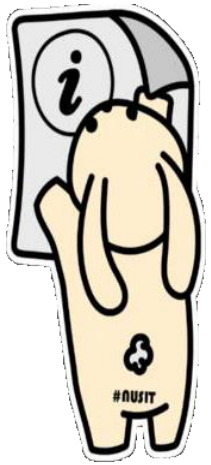
The prerequisites to access the system are the following:

- The project has been approved, and resources have been allocated, either in full or partially.
- PI must provide the NUS IDs of all project members who will access the system.
  - NUS-ID is the original NUS account username (nusstf\xxx, nusstu\xxx, nusext\xxx)
  - It is **NOT** the friendly email username, **NOT** the staff number, Neither the student Matri-card number
- Users are required to enable their HPC access via:

  https://nusit.nus.edu.sg/hpc/get-a-hpc-account/

How to Access

- Ensure you are on NUS Campus network or NUS VPN
- **Option 1**: Terminal Interface or SSH Client tool (Putty, MobaXterm)
  - ssh **nus_id@vanda.nus.edu.sg**
  - "nus_id" is the original NUS email username
  - Login password is the NUS email account password

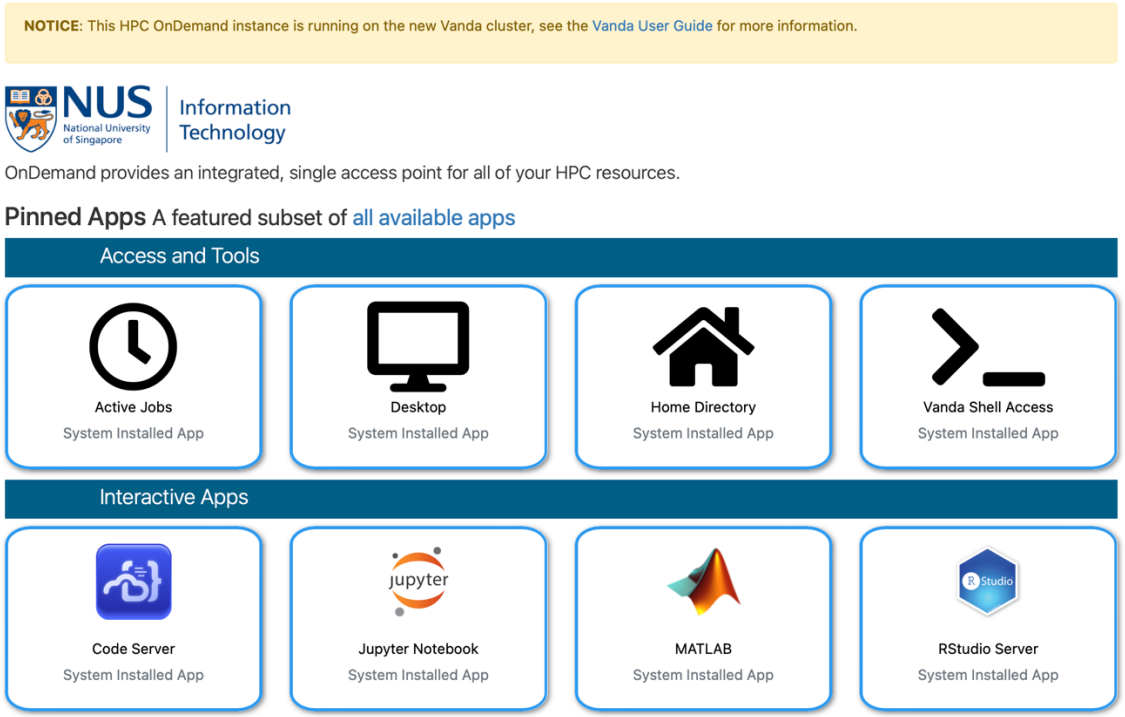


If you are unable to login on campus network, please connect to NUS VPN and try again.

Note: SoC VPN is not supported!

How to Access (Experimental)

- **Option 2**: Use HPC-on-Demand to access the resources (*intranet*)
  - Open **https://vanda.nus.edu.sg** in your browser
  - Login using your NUS ID account and password (MFA may be required)
  - You can use virtual terminal or Jupyter notebook apps



Caveat: current implementation is still in experimental stage.

How to Transfer File

- Mobaxterm built-in sftp client
- Filezilla client
- Linux/Mac OS/Windows Terminal Tools
  - scp
  - rsync
  - sftp

Computing Resources

Resources: Hardware

The Vanda system is comprised of the following resources:

- 1 dedicated login node [vanda.nus.edu.sg] (Internet Access Available)

CPU Partition

168x CPU nodes, each with:

- 2x 36 cores Intel 8452Y CPUs
- 512GB RAM
- 100GbE network connectivity

GPU Partition

102x GPU nodes, each with:

- 2x NVIDIA A40 – 48GB
- 2x 36 cores Intel 8452Y CPUs
- 512GB RAM
- 100GbE network connectivity

This system is a capacity cluster which has the following characteristics:

- *Designed for high-volume independent workloads, e.g. embarrassingly parallel jobs*
- *No non-blocking or low-latency network between nodes. Large MPI jobs performance may be limited.*
- *A40 GPUs has good performance for AI and single-precision simulation (e.g. molecular dynamics) but lower on double-precision FP64 workloads.*

Resources: Data Storage

Directories	Feature	Who can use it?	Disk Quota	Description
/home/svu/$USERID	Available across Vanda Cluster (not the same as atlas or hopper)	Personal	40 GB	Home Directory.
/scratch/$USERID	Available across Vanda Cluster	Personal	500 GB	For quick read/write access to datasets. Files older than 60 days are purged automatically
/scratch/project/CFP-01/$PROJECTID	Available across Vanda Cluster	Project team member	Based on allocation	For quick read/write access to datasets based on project request.
/Project_Storage/CFP-01/$PROJECTID	Available on Login node Only	Project team member	Based on allocation	For storing archive or result datasets. NOT accessible via batch job.
/atlas/home/$USERID /atlas/hpctmp/$USERID	Storage Volume from Atlas/Volta cluster. Available on Login node Only	Personal	Based on allocation	Legacy storage from Atlas/Volta cluster. Use it only to transfer data between system. NOT accessible via batch job.

Resources: Applications

- Scientific applications are managed via environment module and EasyBuild framework.
  - see https://bobcat.nus.edu.sg/hpc/support/ebenv/ for reference
- Users can run the following commands to enable the module and list all available software:

```
module avail
```

- Users can load the module with the following command:

```
module load "module-name"
```

Resources: Applications

Useful Module commands

module list	List currently loaded modules.
module avail	List available packages.
module spider	List available packages in a different format.
module help [*modulefile*]	Description of specified module.
module show [*modulefile*]	Displays information about specified module, including environment changes, dependencies, software version and path.
module load [*modulefile*]	Loads module or specifies which dependencies have not been loaded.
module unload [*modulefile*]	Unloads specified module from environment.
module purge	Unloads all loaded modules

Resources: Applications

- If you would like to build your own software, do note of the following:
  - See https://bobcat.nus.edu.sg/hpc/support/ebenv/building/ for build options and toolchain dependencies
  - On Vanda, the Intel toolchain is recommended
  - Since the architecture of the login node is different, any compilation should be done in a compute node, by submitting an interactive job
- Or you can raise a ticket on nTouch to install any open-source software

Resources: Containers

- The Singularity/Apptainer Containers available for Hopper are also available on Vanda
- If the containers we provide do not suit your needs, you can build your own Singularity/Apptainer containers or put in a request to get us to build it for you
- When building containers yourself, do ensure that **CUDA libraries** are installed and that the Deep Learning framework you install is the **GPU version**
- **Pre-built containers available at**
  - **/app1/common/singularity-img/hopper/**
- See Hopper guide for more information

Compute Budget

Type of Budgets

- There are two types of compute budget in Vanda
  - Project budget
  - Personal budget

Project Budget

- Project Budgets have been allocated and enabled according to the call-for-projects allocation.
- Check your allocated project ID/name and consumption using
  - **'hpc project'**
- If you have just login, you will be prompted to run '**amgr login**' and input your password one more time.

```
[rikky@vanda ~]$ hpc proj

================================================================================
     Vanda CPU and GPU-Hours Credits for:  rikky
================================================================================
  Project-Name  | Total CPU Hours | CPU Net Balance | Total GPU Hours | GPU Net Balance
--------------------------------------------------------------------------------
     NUSIT-HPC  |      3000000.0  |      2997552.0  |          100000 |          99997
         Users  : mcosta,sauyong,cceyeoeh,rikky,ccewjh,ccev3422,ccekwk
--------------------------------------------------------------------------------
personal-rikky  |      8000000.0  |      8000000.0  |         6666.67 |         6666.67
         Users  :                          rikky
--------------------------------------------------------------------------------
```

Project Budget

- The command will show projects that you are in, list of members, and the remaining compute budget

```
[rikky@vanda ~]$ hpc proj

================================================================================
     Vanda CPU and GPU-Hours Credits for:  rikky
================================================================================
  Project-Name  | Total CPU Hours | CPU Net Balance | Total GPU Hours | GPU Net Balance
--------------------------------------------------------------------------------
     NUSIT-HPC  |       3000000.0 |       2997552.0 |          100000 |           99997
         Users  : mcosta,sauyong,ceeyeoeh,rikky,ccewjh,ccev3422,ccekwk
--------------------------------------------------------------------------------
personal-rikky  |       8000000.0 |       8000000.0 |         6666.67 |         6666.67
         Users  :                          rikky
--------------------------------------------------------------------------------
```

Total allocated CPU hours

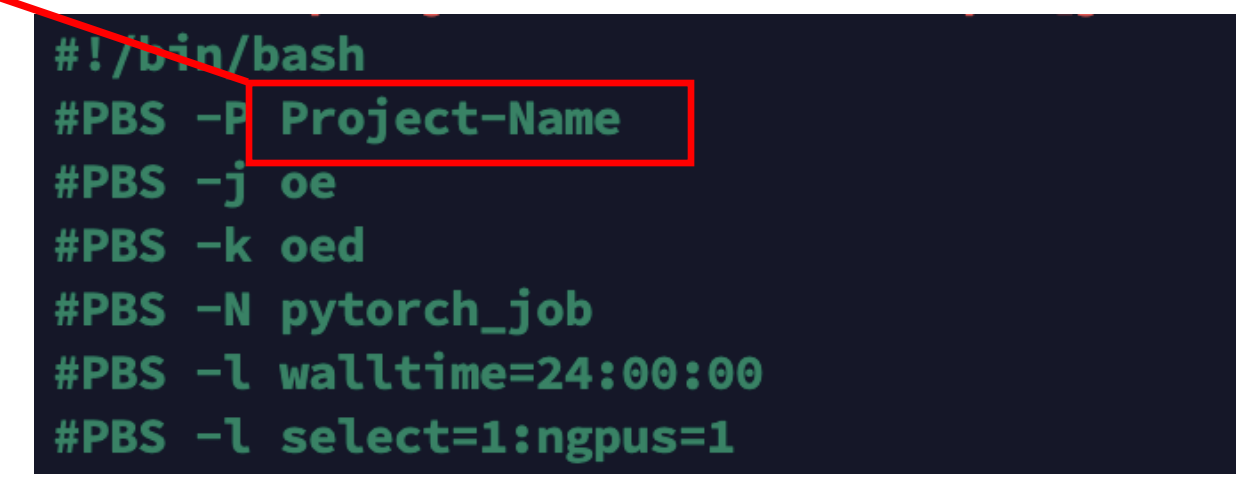
Balance of CPU hours allocation

Total allocated GPU hours

Balance of GPU hours allocation

Project Budget

- In your job submission script, set the Project name in the #PBS -P Project-Name field in your job script
  - **#PBS -P CFP01-CF-XXX**
  - **#PBS -P CFP01-SF-XXX**



Personal Budget

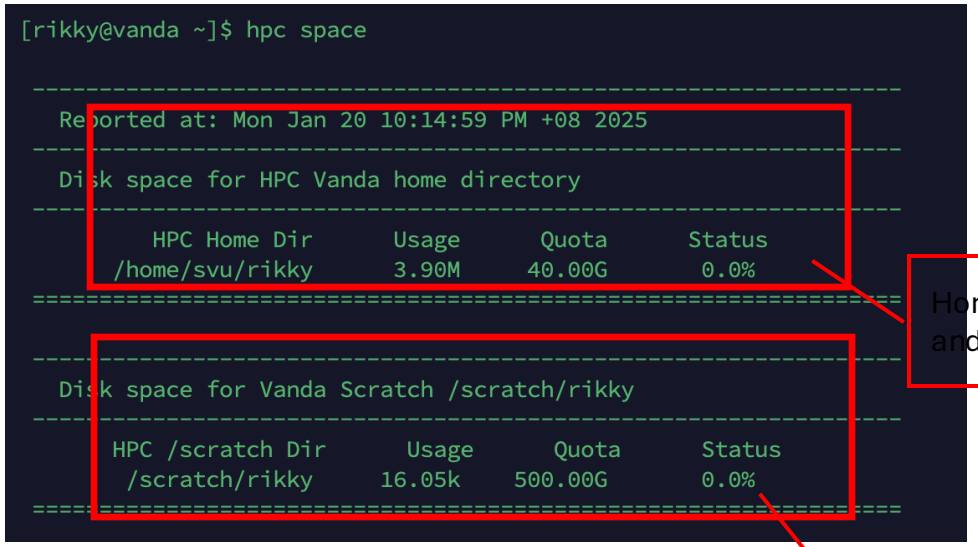
- Personal budget is assigned to every users automatically with the following limit:
  - 10,000 CPUhr and 1,000 GPUhr
  - Quota will be reset every year and cannot be added
- You do not need to specify anything under “#PBS -P ” in the script to use your personal budget.

```
#!/bin/bash
#PBS -j oe
#PBS -k oed
#PBS -N Job_Name
#PBS -l walltime=1:00:00
#PBS -l select=1:ncpus=36:mpiprocs=1:ompthreads=36:mem=250gb

cd $PBS_O_WORKDIR;
```

Storage Quota (/home and /scratch)

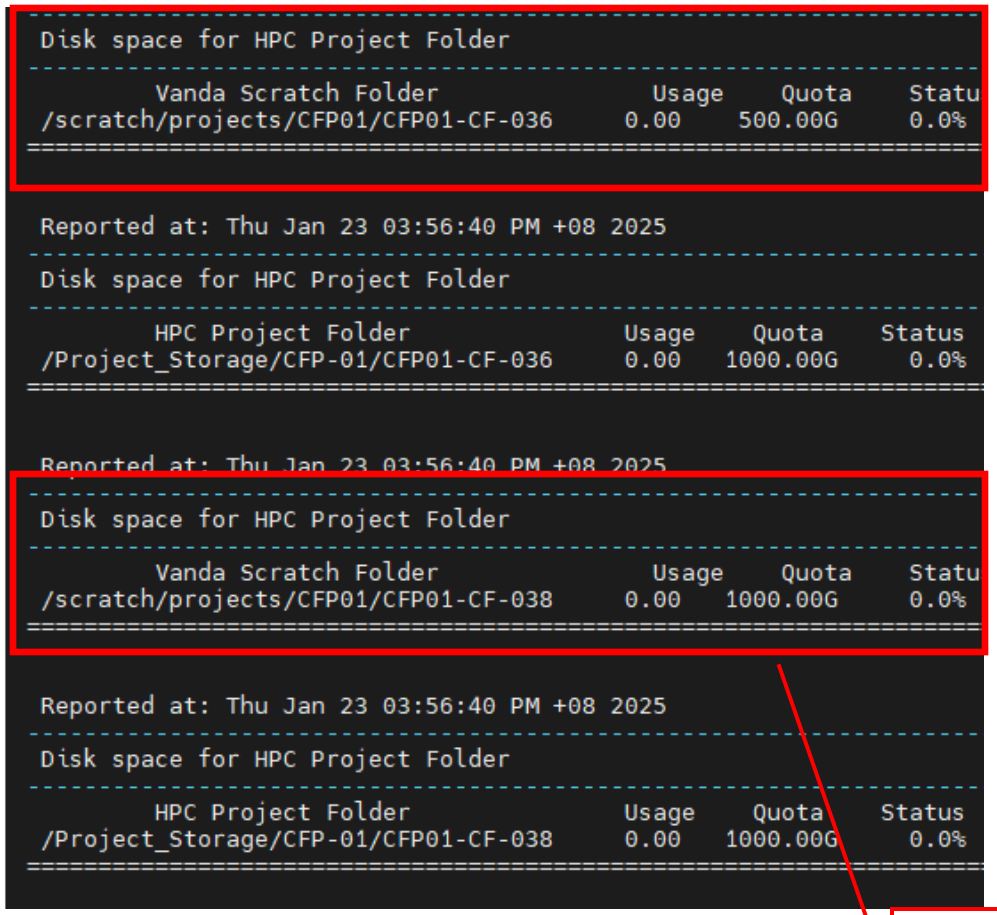
- The workspace storage quota has been allocated and enabled according to your request.
- Check your allocated project ID/name and consumption using
  - **'hpc space'**
- The output will show storage usage for home, scratch, and project



Storage Quota (/scratch/project/)

- Based on project request, we provide a communal scratch space that can be used by the project's team members.
  - Mount point "*/scratch/project/CFP0X/<projectID>*"
- Team members should be able to read/write the project folder
- Check your allocated project ID/name and consumption using

  'hpc space --project'



Project scratch usage and quota

Storage Quota (/Project_Storage/)

- The Project storage is intended for storing result or static data of your workloads
  - The volume is only accessible from login node
- The location can be found in this mount point
  - *"/Project_Storage/CFP0X/<projectID>"*
- Team members should be able to read/write the project folder
- Check your allocated project ID/name and consumption using

  'hpc space --project'

```
Disk space for HPC Project Folder
        Vanda Scratch Folder                Usage    Quota    Statu
/scratch/projects/CFP01/CFP01-CF-036      0.00    500.00G     0.0%
================================================================

Reported at: Thu Jan 23 03:56:40 PM +08 2025
Disk space for HPC Project Folder
        HPC Project Folder                  Usage    Quota    Status
/Project_Storage/CFP-01/CFP01-CF-036      0.00   1000.00G     0.0%
================================================================

Reported at: Thu Jan 23 03:56:40 PM +08 2025
Disk space for HPC Project Folder
        Vanda Scratch Folder                Usage    Quota    Statu
/scratch/projects/CFP01/CFP01-CF-038      0.00   1000.00G     0.0%
================================================================

Reported at: Thu Jan 23 03:56:40 PM +08 2025
Disk space for HPC Project Folder
        HPC Project Folder                  Usage    Quota    Status
/Project_Storage/CFP-01/CFP01-CF-038      0.00   1000.00G     0.0%
================================================================
```

Project scratch usage and quota

Job Submission

Submission Steps

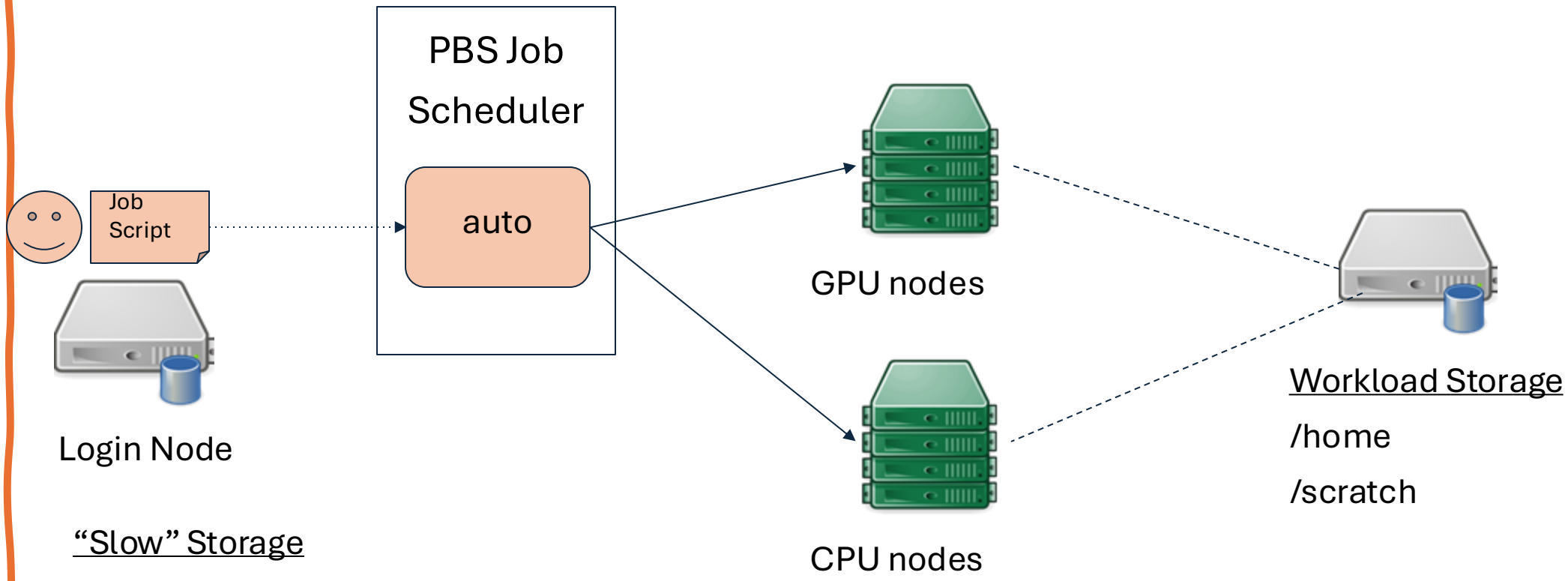
To submit jobs, users typically have to run these steps:

1. Prepare your scripts/apps/data in your working directory
2. Create a PBS job script and save it in your working directory
3. Submit PBS job script to PBS Job Scheduler

Once job is submitted

1. Job will enter PBS Job Scheduler queue
2. Job Scheduler waits for server resources to be available
3. If available, Job Scheduler executes your script on compute nodes

Conceptual View of Job Submission



There is no need to define queue. It will be decided automatically based on resource requirement.

Jobs

- All workloads must be submitted as jobs to the job scheduler
- Use ‘hpc cpu’ or ‘hpc gpu’ command to list out the sample job submission script and instructions
- Jobs can be submitted as:
  - Interactive job
  - Long Batch job

Short Interactive jobs

Job profile

- Direct access to the CPU and GPU to do quick run or testing before running long batch jobs

Queue policy:

- Max wall time 12 hours
- Default 1x CPU (36c) or 1x A40 GPU
- Max 2x CPUs (72c) or 2x A40 GPU
- Max Concurrent Jobs - 4

Interactive: Command Line Interface

CPU

- ***qsub -I -l select=1 -l walltime=02:00:00 -P <PROJECTID>***
  1. "***select=1***" will allocate 1 socket, 36 cores, 250gb of memory
  2. To use a full node (2 CPU sockets), use "***-l select=2***" with "***-l place=pack***"
- You should see your prompt change to nusnet_id@cn-XXX

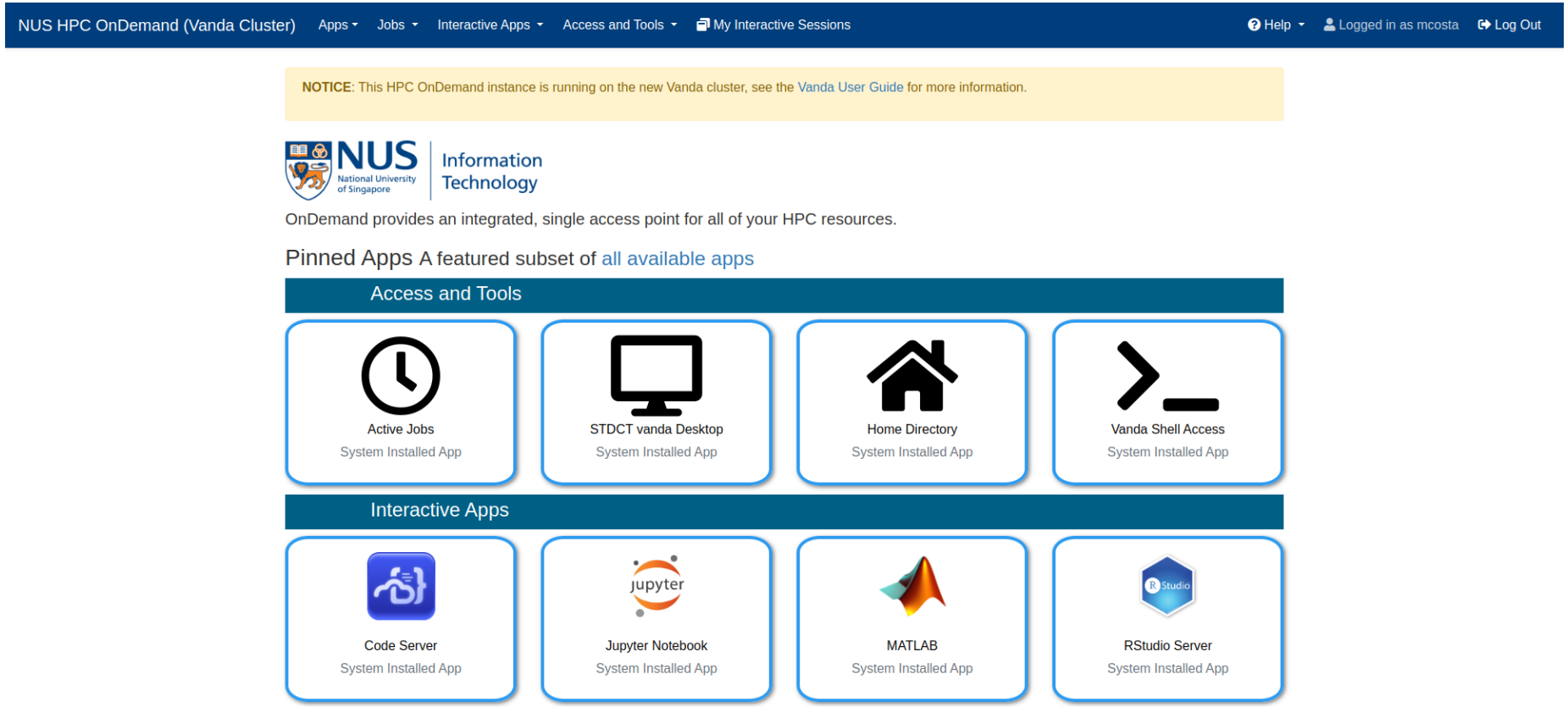
GPU

- ***qsub -I -l select=1:ngpus=1 -l walltime=02:00:00 –P <PROJECTID>***
  1. "***select=1:ngpus=1***" will allocate 1 socket, 36 cores, 250gb of memory, 1 GPU
  2. To use a full node (2 CPU sockets, two GPUs), use "***select=2:ngpus=1***" with "***-l place=pack***"
- You should see your prompt change to nusnet_id@gn-a40-XXX

If no –P <projected> provided, will default to personal credits

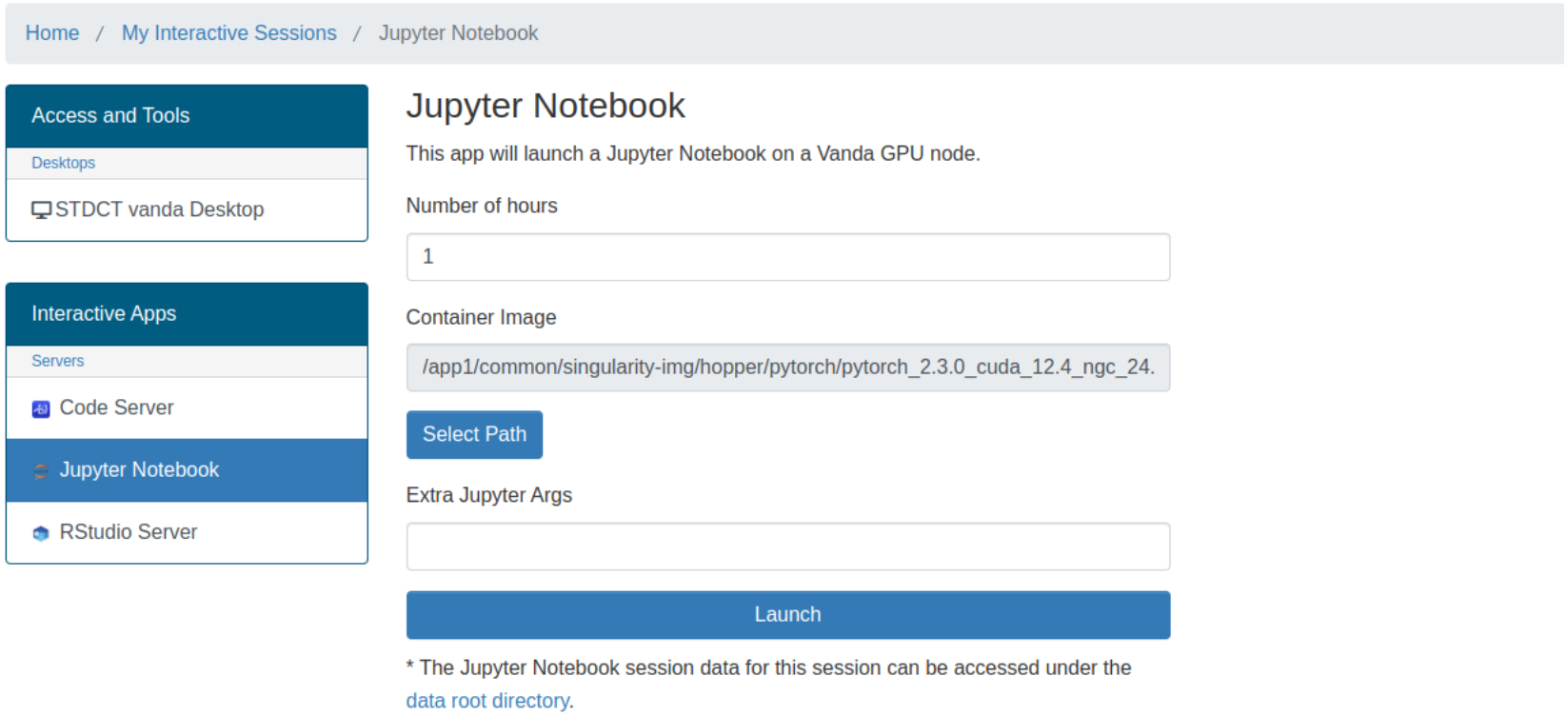
Interactive: Jupyter, RStudio, Matlab, etc

Visit https://vanda.nus.edu.sg and log in

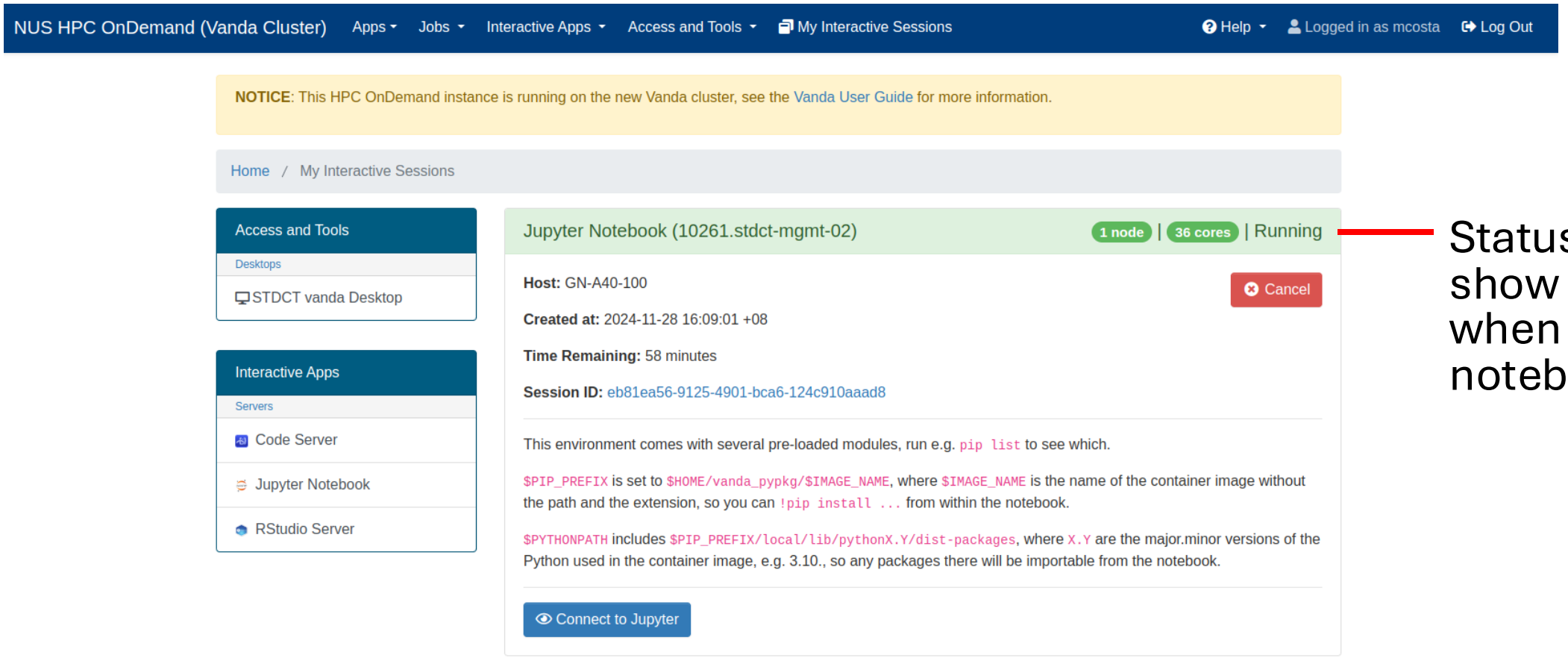


Interactive: Jupyter Notebooks

Example: submitting interactive Jupyter Notebooks apps



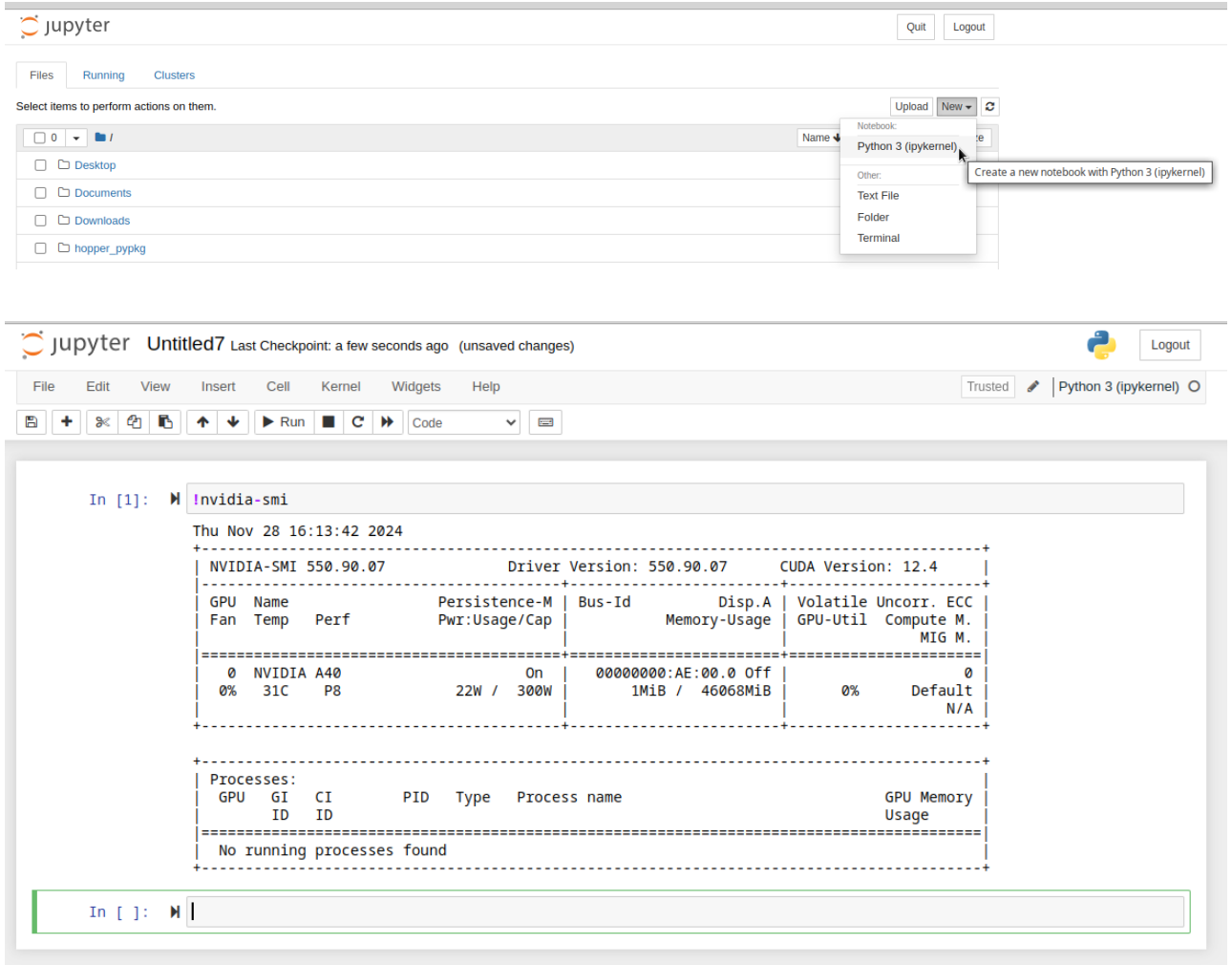
Interactive: Jupyter Notebooks



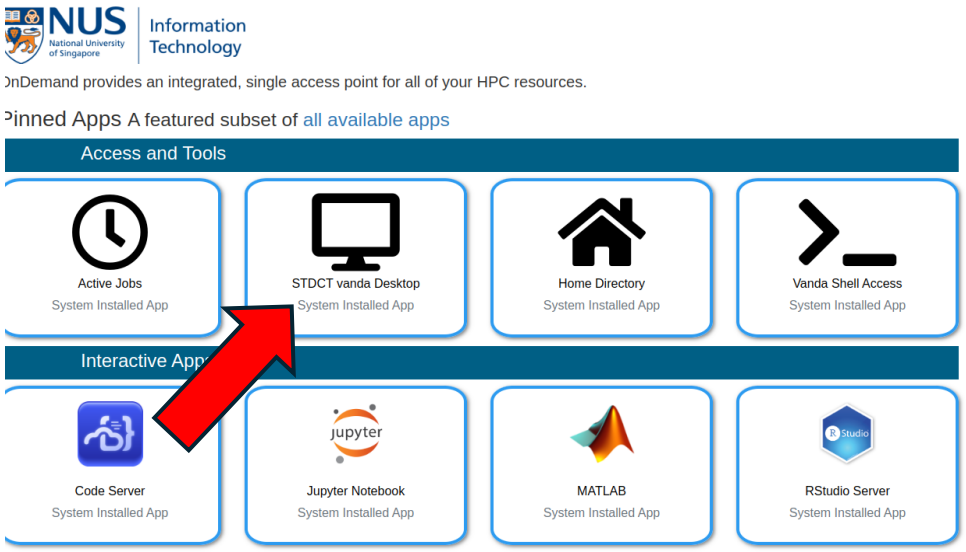
Status page will show "**Running**" when your notebook is ready

Interactive: Jupyter Notebooks

Once the job is running, you can create a new notebook.



Interactive: GUI Applications



- You can launch a GUI session in a GPU node with the Desktop App
- From there, you can open a terminal, load modules and launch applications
- For hardware 3D acceleration, you may need to unload Mesa and run the application with vglrun
- Examples:
  - *module load VMD && module unload --force Mesa && vglrun -d egl vmd*
  - *module load ParaView && module unload --force Mesa && vglrun -d egl paraview*
  - *module load ANSYS && vglrun -d egl fluent*

Long Batch Jobs

Job profile

- Long running jobs with more CPUs or GPUs

Queue policy:

- CPU
  - Default 1x CPU (36c), max 10x CPUs (360c)
  - Max Concurrent Jobs – 4
- GPU
  - Default 1x GPU, max 2x GPUs
  - Max Concurrent Jobs – 4
- Default wall time 24 hours, max wall time 168 hours

Sample MPI Job Script (CPU)

```
#!/bin/bash
#PBS -N Job_Name
#PBS –P <project ID>
#PBS -l select=1:ncpus=36:mpiprocs=36:ompthreads=1:mem=250gb

cd $PBS_O_WORKDIR

# for example
module load HPL/2.3-intel-2023b

# run command
mpirun  xhpl
```

Sample MPI Job Script (CPU)

NOTES on MPI Jobs

- Using a full node
  - The default "chunk" is one CPU socket, and each node has two sockets
  - You request a full node with either
    - #PBS -l select=2:ncpus=36:mpiprocs=36:ompthreads=1:mem=250gb –l place=pack **,or**
    - #PBS -l select=1:ncpus=72:mpiprocs=72:ompthreads=1:mem=500gb

- For MPI applications, you can use more than one node, but note that the inter-node fabric is 100G Ethernet, not Infiniband
  - You can request e.g. 2 full nodes with either
    - #PBS -l select=4:ncpus=36:mpiprocs=36:ompthreads=1:mem=250gb –l place=pack **,or**
    - #PBS -l select=2:ncpus=72:mpiprocs=72:ompthreads=1:mem=500gb
  - But do measure the scaling between using one node or two

Sample non-MPI Job Script (CPU)

```
#!/bin/bash
#PBS -N Job_Name
#PBS –P <project ID>
#PBS -l select=1:ncpus=36:mpiprocs=1:ompthreads=36:mem=250gb
cd $PBS_O_WORKDIR

# using R as an example non-MPI application
source /app1/ebapps/arches/flat-avx2/load_R-4.2.1_modules.sh

# run command
Rscript script.R
```

Sample non-MPI Job Script (GPU)

```
#!/bin/bash
#PBS -N Job_Name
#PBS –P <project ID>
#PBS -l select=1:ncpus=36:mpiprocs=1:ompthreads=36:mem=250gb:ngpus=1
cd $PBS_O_WORKDIR

image=/app1/common/singularity-img/hopper/pytorch/pytorch_2.4.0a0-cuda_12.5.0_ngc_24.06.sif

singularity exec $image bash << EOF > stdout.$PBS_JOBID 2> stderr.$PBS_JOBID

python -c "import torch; print(torch.cuda.get_device_name())"
# or python script.py, etc.

EOF
```

Useful PBS Job Commands

Action	**Command**
Job submission	`qsub my_job_script.txt`
Job deletion	`qdel my_job_id`
Job listing (Simple)	`qstat`
Job listing (Detailed)	`qstat -ans1`
Queue listing	`qstat -q`
Completed Job listing	`qstat -H`
Completed and Current Job listing	`qstat -x`
Full info of a job	`qstat -f job_id`

Tips: Array Jobs

- One often has many similar jobs to run. An array job is simply a job duplicated many times. It is organised as an array, with each job having in the array its own array index, which is also stored in an environment variable PBS_ARRAY_INDEX
- Within the job script, the PBS_ARRAY_INDEX variable allows you to control what each job in the array will do differently, by e.g.
  - passing a different parameter,
  - reading a different input file or
  - moving into a different directory
- See e.g. https://sydney-informatics-hub.github.io/training.artemis.interhpc/02-array-jobs/index.html for some examples

Tips: Use of resources

- Note that resources are allocated in units of *vnodes** , each with:
  - 36 CPU cores,
  - 256GB of memory
  - and, in the case of GPU nodes, one GPU with 48GB of memory

- There are multiple options to fully utilize these resources, e.g.:
  - multiple MPI processes
  - multiple OpenMP threads
  - multiple concurrent processes
  - or a combination of the above

* each physical node has 2 *vnodes*

Tips: Frequently Asked Questions

- Something works on the compute nodes but not on the login node, why?
  - The login node is a VM, so an emulated CPU micro-architecture. For anything that depends on the specific CPU micro-architecture, use an interactive session in a compute node.
- Why is e.g. VASP so slow on Vanda's GPUs?
  - NVIDIA's A40 GPUs are not meant for double precision simulation, their FP64 performance is much lower (1:64) than their FP32 performance
  - For double precision workloads, you can apply for time in the Hopper cluster

Tips: Frequently Asked Questions

- Received error “Required Budget is not available”
  - Do Check your project budgets. Your budget should be sufficient to cover the total CPUhr or GPUhr requested in your job script.

```
[sauyong@vanda ~]$ qsub -I -l select=1:ngpus=1 -l walltime=02:00:00
qsub: Budgets-am_hook: Required budget is not available for this transaction.
[sauyong@vanda ~]$ qsub -I -l select=1:ngpus=1 -l walltime=02:00:00 -P "CPD"
qsub: Budgets-am_hook: project with CPD is not found
[sauyong@vanda ~]$ qsub -I -l select=1:ngpus=1 -l walltime=02:00:00 -P "CFP01-SF-013"
qsub: Budgets-am_hook: User sauyong is not authorized for CFP01-SF-013 project
```

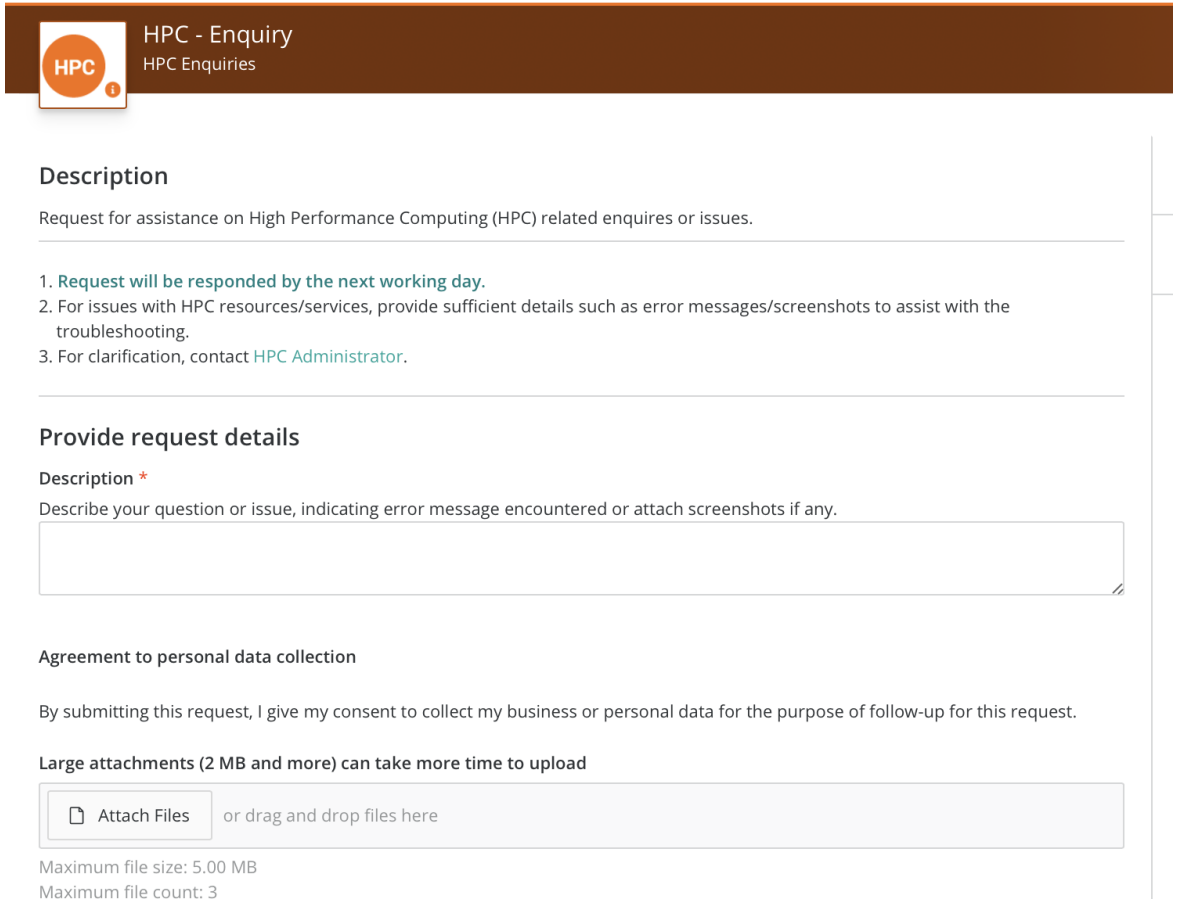
Other Matters

Optimizations

- Ensure that your jobs can fully utilize the allocated resources.
- A PBS script request only reserves the CPUs/GPUs; users must configure the application/code to utilize all resources.
- CPU and GPU usage is expensive. Underutilization will lead to waste and loss of opportunity.

Support Matters

- If you encounter any issue, please raise a ticket through ntouch
- https://ntouch.nus.edu.sg > “HPC Enquiries”

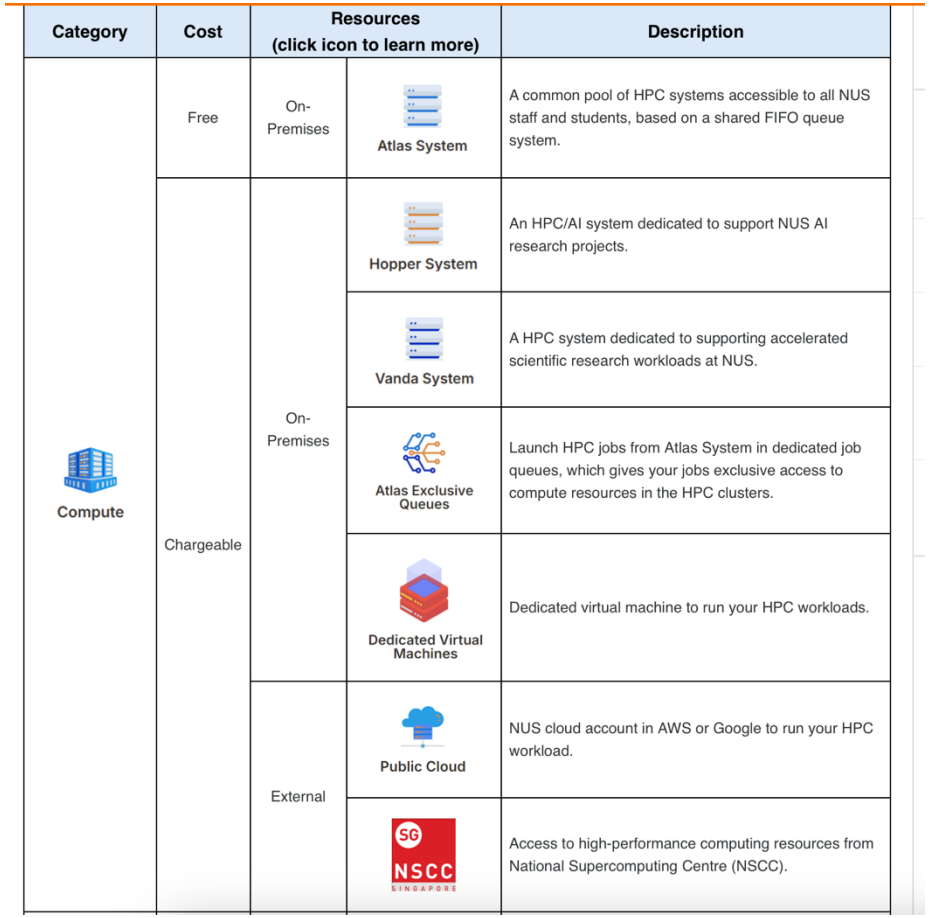


Other HPC Service

- For other HPC services, please go to ntouch
- Select the **guide for HPC catalog**

Resource:

IT quick start guide for high-performance computing resources



Category	Cost	Resources (click icon to learn more)		Description
Compute	Free	On-Premises	Atlas System	A common pool of HPC systems accessible to all NUS staff and students, based on a shared FIFO queue system.
	Chargeable	On-Premises	Hopper System	An HPC/AI system dedicated to support NUS AI research projects.
			Vanda System	A HPC system dedicated to supporting accelerated scientific research workloads at NUS.
			Atlas Exclusive Queues	Launch HPC jobs from Atlas System in dedicated job queues, which gives your jobs exclusive access to compute resources in the HPC clusters.
			Dedicated Virtual Machines	Dedicated virtual machine to run your HPC workloads.
		External	Public Cloud	NUS cloud account in AWS or Google to run your HPC workload.
			NSCC SINGAPORE	Access to high-performance computing resources from National Supercomputing Centre (NSCC).

Acknowledgements

Our mission is to provide top-tier HPC resources to support your research. To help improve our services and track our impact, please kindly cite the REC team in your publications.

Below is an example of a citation that may work for you:

“We would like to acknowledge that computational work involved in this research work is partially/fully supported by NUS IT’s Research Computing group under grant number NUSREC-HPC-00001”

Maintenance News

Vanda will be running on reduced capacity on between **8th and 14th February** for data center retrofitting and experiments.