

HELIOS

# About

- HELIOS is a computer cluster
  - computers are aggregated together to create a more powerful machine.
  - User can access a specific machine (node) inside the cluster
- or
- use multiple nodes at the same time for task that are computation heavy



# Connect to Helios

- Should have received your logging info

# Connect with linux

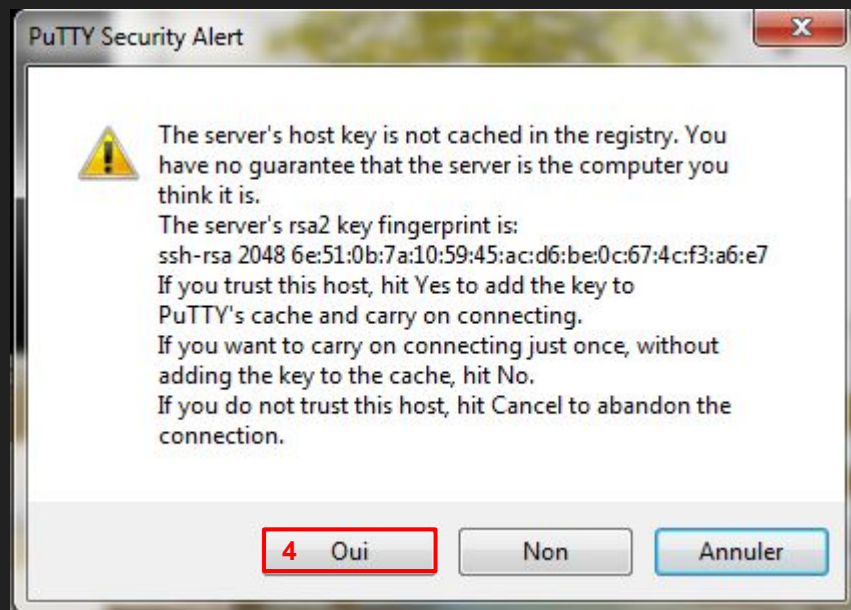
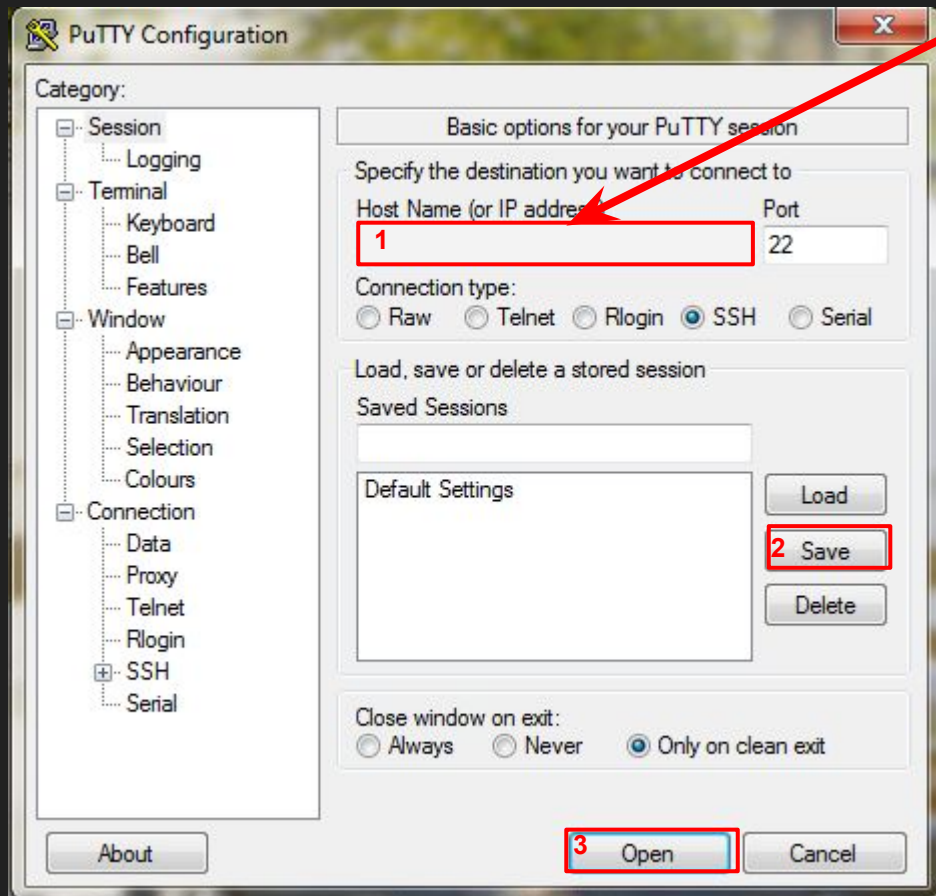
- `ssh <account-name>@helios.calculquebec.ca`

# Connect with windows

- Multiple solutions
  - a. Windows subsystem for linux
    - Makes ubuntu command line available on your windows machine
    - Can install all ubuntu packages
  - b. Install putty
    - This is a ssh client for windows
  - c. Use git bash
    - Git comes with ssh

# Putty

<account-name>@helios.calculquebec.ca



# HELIOS

- Once connected you should see something like this →

```
user1@mila-145D ➤ ssh delaunay@helios.calculquebec.ca
delaunay@helios.calculquebec.ca's password:
Last login: Tue Dec 11 13:54:22 2018 from 206.167.243.8
=====
Vous êtes sur un noeud de login de Helios (Calcul Québec).
- Nous n'effectuons pas de sauvegarde de vos fichiers.
- N'utilisez pas le noeud de login pour executer votre code.

this is a Calcul Québec login node for Helios.
- There is no backup of users files.
- Do not use this node to run code.

Rapportez tout problème à / Report any problems to: helios@calculquebec.ca
Documentation: https://wiki.calculquebec.ca/
Suivre sur Twitter/Follow on Twitter: https://twitter.com/CQ_Helios
État des serveurs: http://serveurscq.computecanada.ca
=====
Vous pouvez maintenant utiliser l'environnement logiciel de Calcul Canada
en utilisant la commande

source /admin/bin/enable_cc_cvmfs

ou

touch $HOME/.helios_ccstack

(cette dernière sera persistante)
=====
You can now use the software environment of Compute Canada by running
the command

source /admin/bin/enable_cc_cvmfs

or

touch $HOME/.helios_ccstack

(this last one will be persistent)
=====
Les répertoires $SCRATCH et $RAP ont été fixés pour le projet jvb-000-aa. Si vous voulez changer cette op
on, exécutez la commande
The directories $SCRATCH and $RAP have been set for the project jvb-000-aa. If you want to change this op
on, execute the command
source /etc/profile.d/1-env_directories.sh <project_id>.
=====
Description /home+/rap SCRATCH
=====
Nb fichiers/Num files user: delaunay 16 (0%) 0 (0%)
Espace groupe/Space group: jvb-000-01 14 TB (142%) 17 TB (355%)
=====
ATTENTION, vous ou votre groupe dépassez les quotas sur nos systèmes de fichiers.
Veuillez effacer ou archiver les fichiers inutilisés.
WARNING, you or your group is over quota on our filesystems.
Please erase or archive unused files.
=====
Dernière mise à jour / Last updated : 2018-12-12 08:24

Due to MODULEPATH changes, the following have been reloaded:
1) openmpi/2.1.1

[delaunay@helios1 ~]$
```

# Bash - A few useful functions

- **mkdir:** make directory

```
[delaunay@helios1 ~]$ mkdir my_project
```

- **ls:** list directories

```
[delaunay@helios1 ~]$ ls  
my_project  script_moab
```

- **cd:** change directory

```
[delaunay@helios1 ~]$ cd my_project/  
[delaunay@helios1 my_project]$
```

- **pwd:** print working directory

```
[delaunay@helios1 my_project]$ pwd  
/home/delaunay/my_project
```

- **cat:** show file content

```
[delaunay@helios1 ~]$ cat .bashrc
```

- **vi:** open a file with vi
  - to edit file press i
  - to quit press ESC then enter :q

```
[delaunay@helios1 ~]$ vi .bashrc
```



# How to run AI Stuff

- Compute Resource are shared among a lot of people
- You need to demand access to those resources
- The resources will be allocated to you by the ...
  - resource manager (torque) / job scheduler: moab
  - <https://wiki.calculquebec.ca/w/Moab/en>

# How to run AI Stuff

```
> msub -N skynet_1 -A jvb-000-ag -l nodes=1:gpus=1,walltime=00:01:00 srun_1.pbs
```

- -N: job name
  - -A: Account name (never changes)
  - -l: Requested Resources
    - nodes=1                   => 1 node
    - gpus=1                    => 1 GPU
    - walltime=06:00:00       => Run for 6 hours
  - -M: Email Notification
  - -m: Event that triggers an email notification
    - -m bea (b: begin, e: ends, a: killed)
- 
- All the output files will be placed inside the directory where the job was created
  - Jobs can only run for **12 hours max!**
  - <http://docs.adaptivecomputing.com/maui/commands/msub.php>

# How to run AI Stuff

- Singularity
  - Sandboxed Execution
  - You have control over the container (image)
  - Pre configured container to simplify your life

```
singularity exec --nv --bind source:dest container script.sh
```

- `exec` : execute a script
- `--nv` : mount NVIDIA GPUs
- `--bind` : make host folder (source) available inside the container (dest)
- `container:` image of the container you want to use
- `script.sh:` the script you want to run

# How to run AI Stuff

- Create a batch wrapper

```
[delaunay@helios1 ~]$ cat srun_1.pbs
#!/bin/bash

module --force purge
PATH=$PATH:/opt/software/singularity-3.0/bin/

# set the working directory to where the job is launched
cd "${PBS_O_WORKDIR}"

# Singularity options
IMAGE=/rap/jvb-000-aa/singularityimages/pytorch.simg

FOLDERS=$RAP,$HOME,$SCRATCH

SINGULARITY_EXEC="singularity exec --nv --bind $FOLDERS $IMAGE"

# start your python script
$SINGULARITY_EXEC python mnist.py
```

# How to run AI Stuff

```
[delaunay@helios1 MixedPrecisionTutorial]$ msub -N skynet_1 -A jvb-000-ag -l nodes=1:gpus=1,walltime=00:01:00 run.pbs
```


371993

Job ID



```
[delaunay@helios1 MixedPrecisionTutorial]$ ls -all | grep 371993
-rw----- 1 delaunay jvb-000-01 1406 Dec 13 14:19 371993.err
-rw----- 1 delaunay jvb-000-01   64 Dec 13 14:19 371993.out
```

Job Output



# Demo

```
# copy the example locally
git clone https://github.com/Delaunay/helios

# enter the example
cd helios

# Schedule the example to be run
msub -N skynet_1 -A jvb-000-ag -l nodes=1:gpus=1,walltime=00:01:00 srun_1.pbs

# Show <job_id>.out
watch tail -n 20 $(ls -rt | grep .out | tail -n 1)
```

# How to run AI Stuff

- Specifying all the arguments can be tedious
- You can specify the arguments that do not change in your script itself!

```
#!/bin/bash
#PBS -N skynet_1
#PBS -A jvb-000-ag
#PBS -l nodes=1:gpus=1,walltime=00:01:00
```

# How to run AI Stuff

```
# Schedule the example to be run
msub run_2.pbs

# Show <job_id>.out
watch tail -n 20 $(ls -rt | grep .out | tail -n 1)
```



# How to run AI stuff interactively

```
> msub -N skynet_1 -A jvb-000-ag -l nodes=1:gpus=1,walltime=15:00 -I -qtest
```

```
> singularity shell --nv --bind $RAP,$HOME,$SCRATCH /rap/jvb-000-aa/singularityimages/pytorch.simg
```

- Useful for testing & debugging
- The walltime/user time allocated to them is small

# Monitor Jobs

- Email notification from moab
- `showq -w user=<username>`
  - Show the current jobs running for you
- `checkjob <jobid>`
  - Show details on a particular job (resource usage, status)
- `mjobctl -c <jobid>`
  - Cancel job

# Monitor resource usage

- RAM and CPU usage monitoring

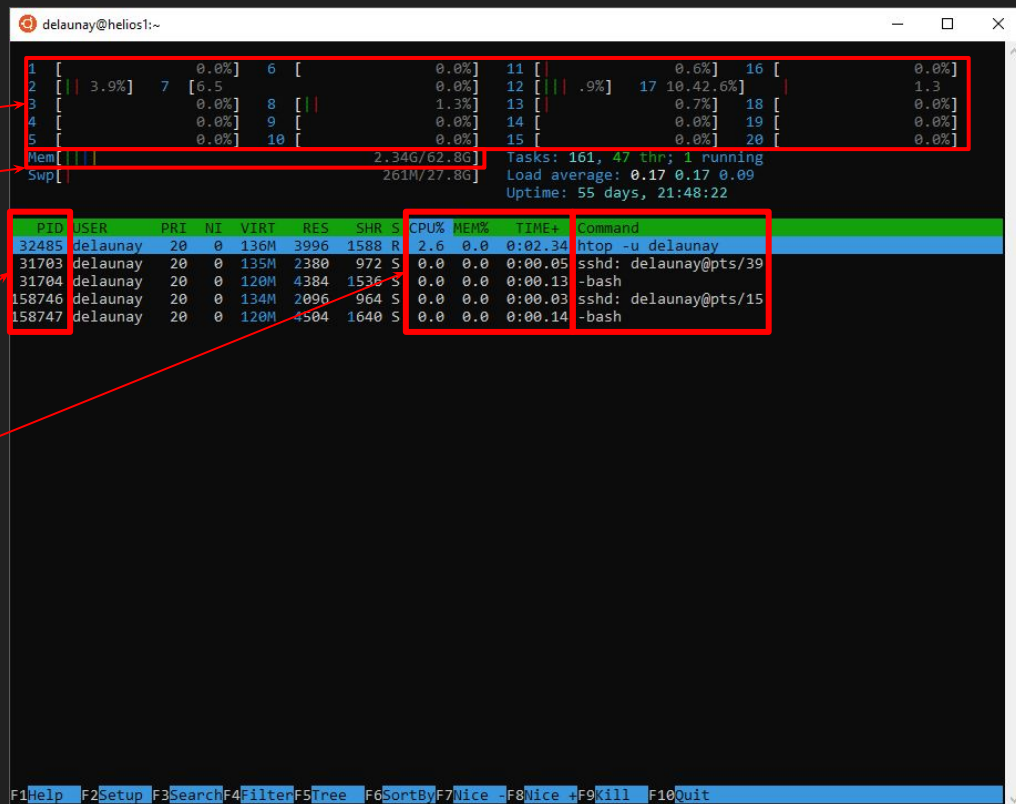
- `htop -u <username>`

- % usage per cores
    - Memory usage

- PID: Program Identifier
    - Resource usage per program

- To kill a running program:

- `kill -9 <PID>`



# Monitor resource usage

- GPU usage monitoring
  - nvidia-smi

```
[delaunay@gpu-k20-03 ~]$ nvidia-smi
Thu Dec 13 12:35:39 2018
```

NVIDIA-SMI 410.73		Driver Version: 410.73		CUDA Version: 10.0	
GPU	Name	Persistence-M	Bus-Id	Disp.A	Volatile Uncorr. ECC
Fan	Temp	Perf	Pwr:Usage/Cap	Memory-Usage	GPU-Util Compute M.
=====					
0	Tesla K20m	On	00000000:05:00.0	Off	0
N/A	23C	P8	16W / 225W	0MiB / 4743MiB	E. Process
-----					
Processes:					GPU Memory
GPU	PID	Type	Process name		Usage
=====					
No running processes found					

Software versions

GPU Compute Usage

GPU Memory Usage

Program using GPU

# Monitor resource usage

- `nvidia-smi --loop=1 --query-gpu=utilization.gpu,utilization.memory,memory.used,memory.total --id=0 --format=csv`

```
[delaunay@gpu-k20-03 ~]$ nvidia-smi --loop=1 --query-gpu=utilization.gpu,utilization.memory,memory.used,memory.total --id=0 --format=csv
utilization.gpu [%], utilization.memory [%], memory.used [MiB], memory.total [MiB]
0 %, 0 %, 0 MiB, 4743 MiB
0 %, 0 %, 0 MiB, 4743 MiB
0 %, 0 %, 0 MiB, 4743 MiB
0 %, 0 %, 0 MiB, 4743 MiB
0 %, 0 %, 0 MiB, 4743 MiB
0 %, 0 %, 0 MiB, 4743 MiB
0 %, 0 %, 0 MiB, 4743 MiB
0 %, 0 %, 0 MiB, 4743 MiB
```

- `--loop=1` run nvidia-smi every second
- `--query-gpu` specify which statistic to print
- `--id=0` show statistic only for the first gpu
- `--format=csv` print each iteration as a new CSV line

# Quality of life

- Modify your `~/ .bashrc` to pre configure your environment

```
module --force purge
```

```
# make singularity 3 available
```

```
PATH=$PATH:/opt/software/singularity-3.0/bin/
```

```
# make a shortcut to create an interactive session
```

```
alias mdebug="msub -N skynet_1 -A jvb-000-ag -l nodes=1:gpus=1,walltime=01:00:00 -I -qtest"
```

```
export SINGULARITY_ARGS="--nv --bind $RAP,$HOME,$SCRATCH /rap/jvb-000-aa/singularityimages/pytorch.simg"
```

```
alias s_shell="singularity shell $SINGULARITY_ARGS"
```

```
alias s_exec="singularity exec $SINGULARITY_ARGS"
```

```
alias show_err="watch tail -n 20 $(ls -rt | grep .err | tail -n 1)"
```

```
alias show_out="watch tail -n 20 $(ls -rt | grep .out | tail -n 1)"
```

```
alias rm_logs="rm *.out *.err"
```

- reload the configuration

```
source ~/.bashrc
```

```
mdebug
```

# More documentation

- How to use Compute Canada Clusters
  - <https://github.com/SMART-Lab/smartdispatch/wiki/How-To-Use-Compute-Canada-Clusters>
  - <https://docs.computecanada.ca/wiki/Python>
- nvidia-smi
  - <http://developer.download.nvidia.com/compute/DCGM/docs/nvidia-smi-367.38.pdf>
- Singularity
  - <https://www.sylabs.io/guides/3.0/user-guide/index.html>