Team Alpha

Compute Canada Tutorial

Accounts *(Please use your assigned accounts)*:

kevuoft	**Whatthefuck.12**
jayluoft	**Projectalpha8**
leouoft	**Leoleouoft@**
Pololi	**Pololi_5566**

The following is not for the first-time setup. All the setups are finished and ready for use.

The following examples are for CPU Runtimes. However, our project requires scaled GPU runtimes (running neural nets parallelly across multiple GPUS). For GPU runtimes, pre-process the data along with EDA on jupyter notebook on CPU runtimes. Once the data and neural nets architecture is ready to run, we will take care of the job submission since it's quite complex. Don't worry about the checkpoints and pickle. We will take care of the training.

Windows:

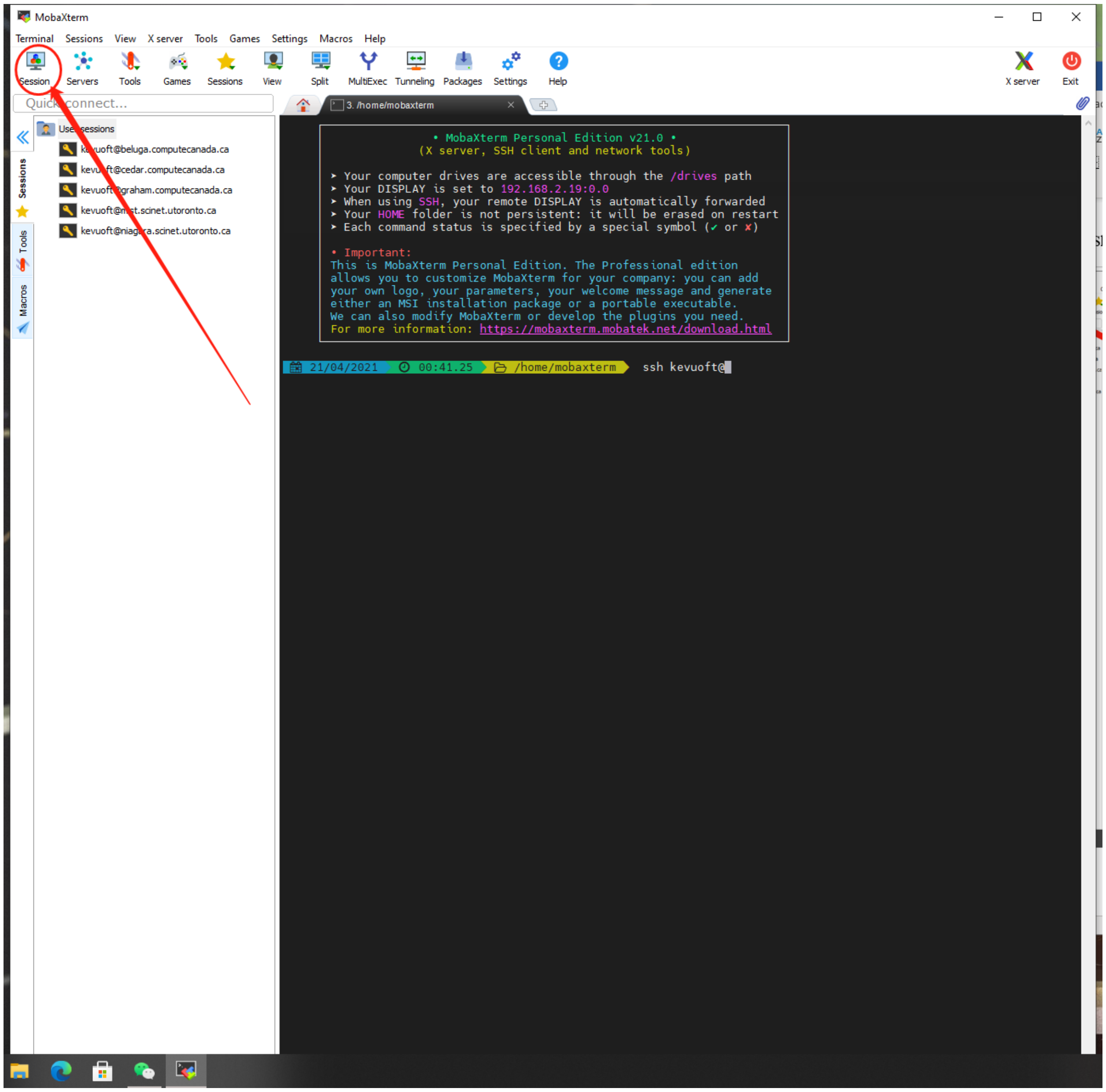
Step 1

Download The MobaXterm Portable Edition

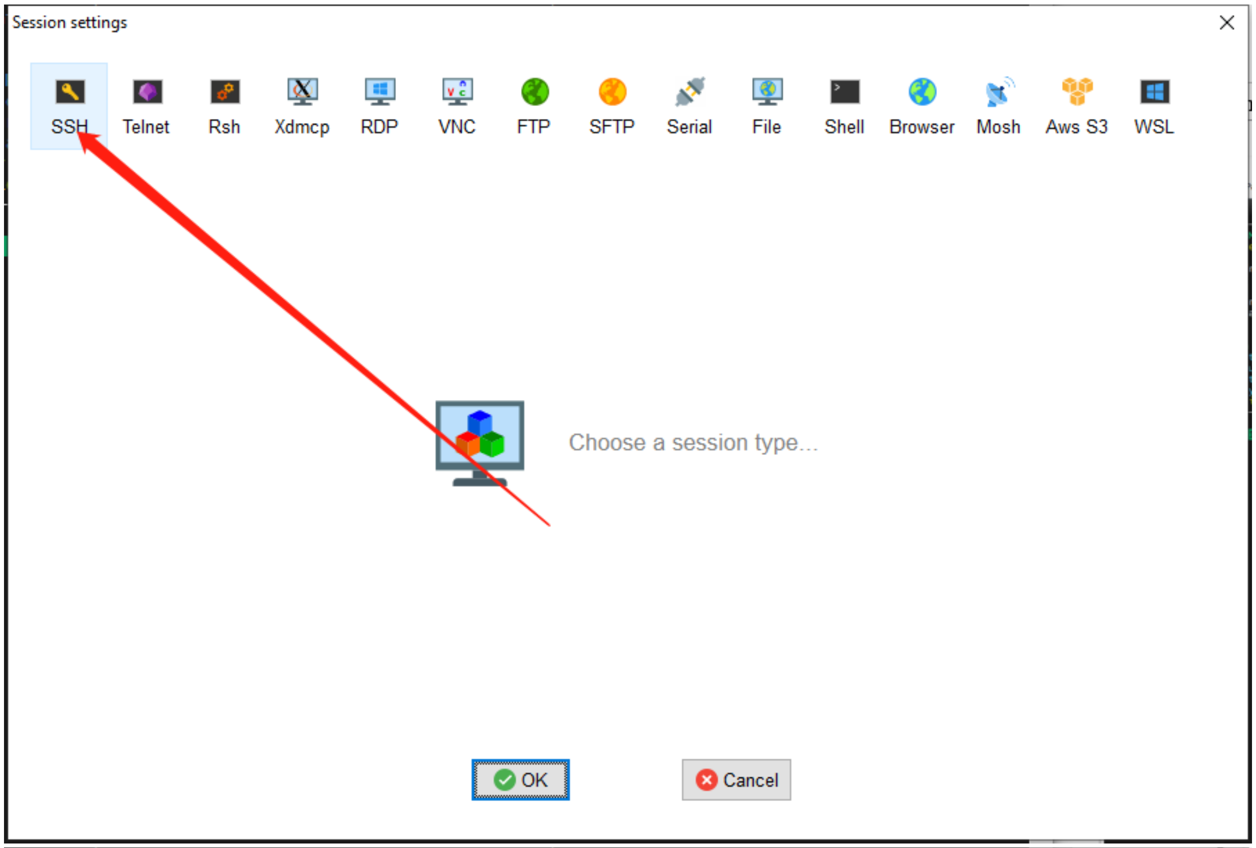
https://mobaxterm.mobatek.net/download-home-edition.html

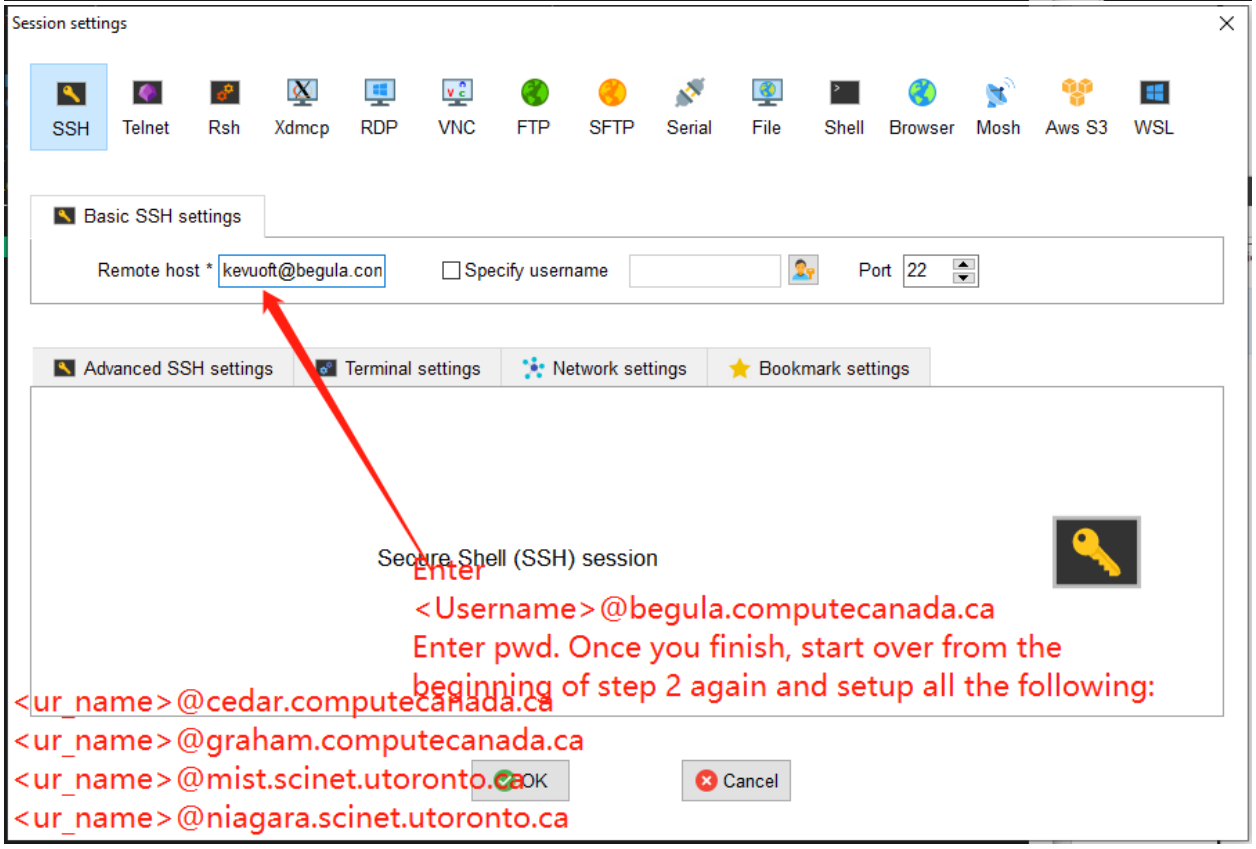
Step 2

Use the Securities Shell to connect with the clusters.


MobaXterm
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
X server Exit
Quick connect...
3. /home/mobaxterm
User sessions
kevuoft@beluga.computecanada.ca
kevuoft@cedar.computecanada.ca
kevuoft@graham.computecanada.ca
kevuoft@mist.scinet.utoronto.ca
kevuoft@niagara.scinet.utoronto.ca
Sessions
Tools
Macros

```
              • MobaXterm Personal Edition v21.0 •
            (X server, SSH client and network tools)

 ➤ Your computer drives are accessible through the /drives path
 ➤ Your DISPLAY is set to 192.168.2.19:0.0
 ➤ When using SSH, your remote DISPLAY is automatically forwarded
 ➤ Your HOME folder is not persistent: it will be erased on restart
 ➤ Each command status is specified by a special symbol (✓ or ✗)

 • Important:
 This is MobaXterm Personal Edition. The Professional edition
 allows you to customize MobaXterm for your company: you can add
 your own logo, your parameters, your welcome message and generate
 either an MSI installation package or a portable executable.
 We can also modify MobaXterm or develop the plugins you need.
 For more information: https://mobaxterm.mobatek.net/download.html

 21/04/2021   00:41.25   /home/mobaxterm   ssh kevuoft@
```





Remote host * kevuoft@begula.con

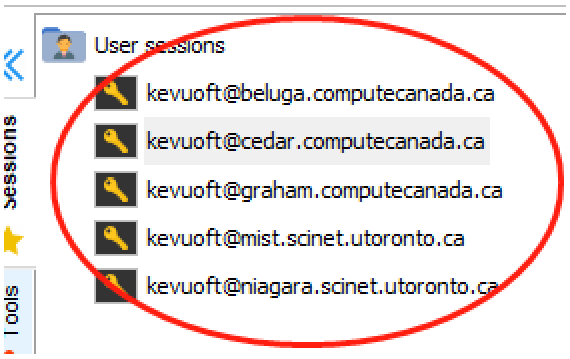
Specify username

Port 22

Enter <Username>@begula.computecanada.ca
Enter pwd. Once you finish, start over from the beginning of step 2 again and setup all the following:

<ur_name>@cedar.computecanada.ca
<ur_name>@graham.computecanada.ca
<ur_name>@mist.scinet.utoronto.ca
<ur_name>@niagara.scinet.utoronto.ca

Once you have finished this step, you won't have to do it again. Instead, you would just click on the existing sessions to connect after the sessions have been set up.



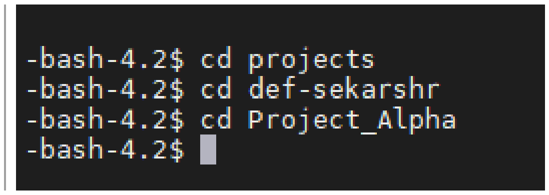
User sessions
kevuoft@beluga.computecanada.ca
kevuoft@cedar.computecanada.ca
kevuoft@graham.computecanada.ca
kevuoft@mist.scinet.utoronto.ca
kevuoft@niagara.scinet.utoronto.ca

The following is the connection to the compute node on the clusters. Note that the steps are different for some of the clusters.

Beluga & Cedar & Graham:

Step 1

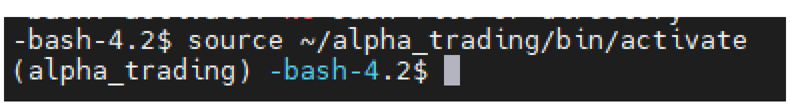
Get in the directory where the data is stored.



```
-bash-4.2$ cd projects
-bash-4.2$ cd def-sekarshr
-bash-4.2$ cd Project_Alpha
-bash-4.2$ 
```

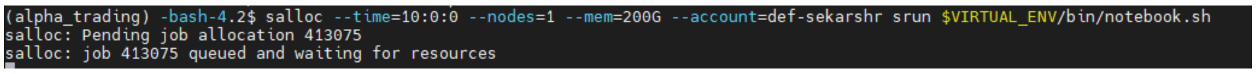
Step 2

Activate the virtual environment.



```
-bash-4.2$ source ~/alpha_trading/bin/activate
(alpha_trading) -bash-4.2$ 
```

Step 3 (Interactive)



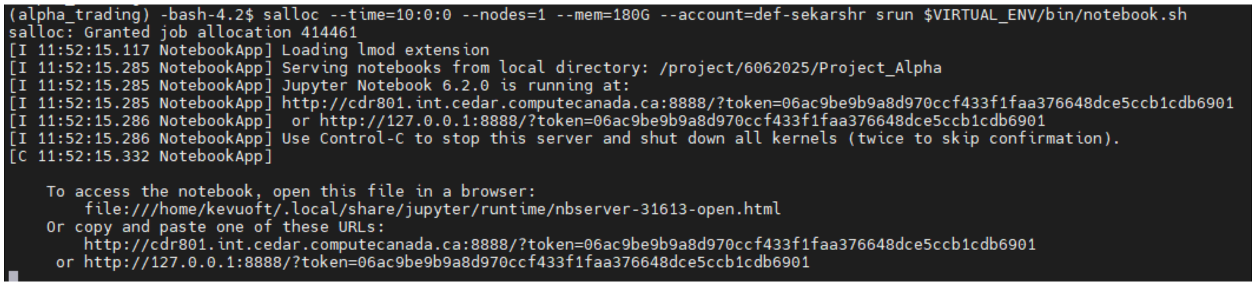
```
(alpha_trading) -bash-4.2$ salloc --time=10:0:0 --nodes=1 --mem=200G --account=def-sekarshr srun $VIRTUAL_ENV/bin/notebook.sh
salloc: Pending job allocation 413075
salloc: job 413075 queued and waiting for resources
```

Type the command above you will get a CPU node with 200GB of memory and 10 hrs of runtime. Your job will be queued and you have to wait until a node becomes available for your job (This will take up to half an hour for CPU node and couple of hours for GPU nodes).

Please see the **appendix** for more commands and cluster information.

Once your job has been granted, a node will be ready for job. For interactive jobs, you will be given a token and a hostname for you to connect to your local runtime.

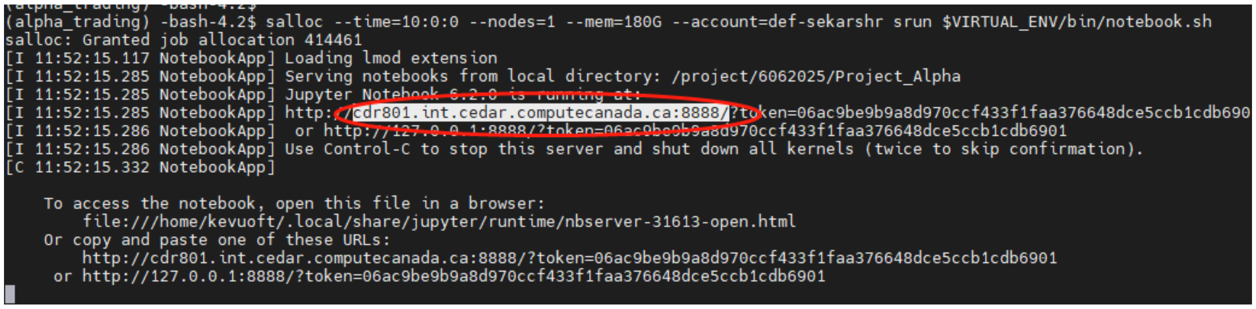
If you get a port error that means your localhost 8888 is probably in use. To solve this problem, run the command “jupyter notebook stop 8888” and try to request the job again.



```
(alpha_trading) -bash-4.2$ salloc --time=10:0:0 --nodes=1 --mem=180G --account=def-sekarshr srun $VIRTUAL_ENV/bin/notebook.sh
salloc: Granted job allocation 414461
[I 11:52:15.117 NotebookApp] Loading lmod extension
[I 11:52:15.285 NotebookApp] Serving notebooks from local directory: /project/6062025/Project_Alpha
[I 11:52:15.285 NotebookApp] Jupyter Notebook 6.2.0 is running at:
[I 11:52:15.285 NotebookApp] http://cdr801.int.cedar.computecanada.ca:8888/?token=06ac9be9b9a8d970ccf433f1faa376648dce5ccb1cdb6901
[I 11:52:15.286 NotebookApp]  or http://127.0.0.1:8888/?token=06ac9be9b9a8d970ccf433f1faa376648dce5ccb1cdb6901
[I 11:52:15.286 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 11:52:15.332 NotebookApp]

    To access the notebook, open this file in a browser:
        file:///home/kevuoft/.local/share/jupyter/runtime/nbserver-31613-open.html
    Or copy and paste one of these URLs:
        http://cdr801.int.cedar.computecanada.ca:8888/?token=06ac9be9b9a8d970ccf433f1faa376648dce5ccb1cdb6901
     or http://127.0.0.1:8888/?token=06ac9be9b9a8d970ccf433f1faa376648dce5ccb1cdb6901
```

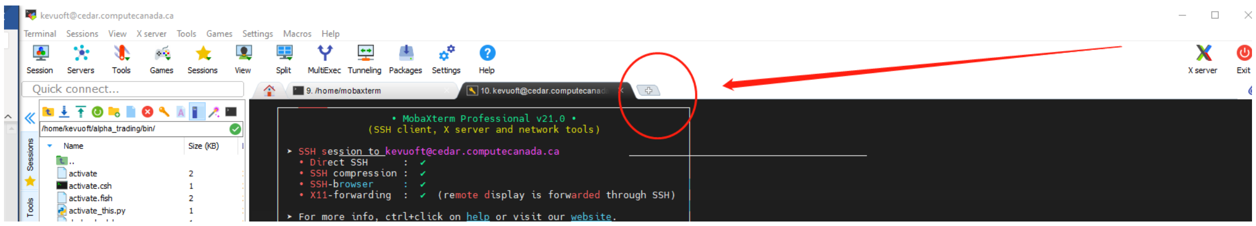
Step 4 (Interactive)

Copy the hostname and port (Do not use control + C, instead, select and drag on the text and right click. You will use the copy option)



```
(alpha_trading) -bash-4.2$
(alpha_trading) -bash-4.2$ salloc --time=10:0:0 --nodes=1 --mem=180G --account=def-sekarshr srun $VIRTUAL_ENV/bin/notebook.sh
salloc: Granted job allocation 414461
[I 11:52:15.117 NotebookApp] Loading lmod extension
[I 11:52:15.285 NotebookApp] Serving notebooks from local directory: /project/6062025/Project_Alpha
[I 11:52:15.285 NotebookApp] Jupyter Notebook 6.2.0 is running at:
[I 11:52:15.285 NotebookApp] http://cdr801.int.cedar.computecanada.ca:8888/?token=06ac9be9b9a8d970ccf433f1faa376648dce5ccb1cdb690
[I 11:52:15.286 NotebookApp]  or http://127.0.0.1:8888/?token=06ac9be9b9a8d970ccf433f1faa376648dce5ccb1cdb6901
[I 11:52:15.286 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 11:52:15.332 NotebookApp]

    To access the notebook, open this file in a browser:
        file:///home/kevuoft/.local/share/jupyter/runtime/nbserver-31613-open.html
    Or copy and paste one of these URLs:
        http://cdr801.int.cedar.computecanada.ca:8888/?token=06ac9be9b9a8d970ccf433f1faa376648dce5ccb1cdb6901
     or http://127.0.0.1:8888/?token=06ac9be9b9a8d970ccf433f1faa376648dce5ccb1cdb6901
```

Start a new session in MobaXterm by pressing on the plus button on top of the bash window.

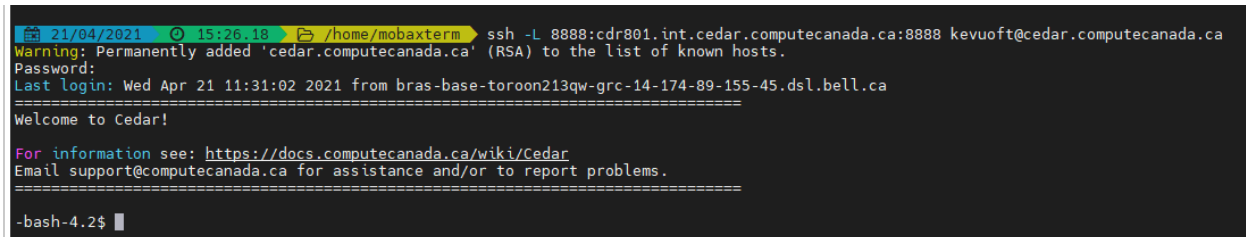

kevuoft@cedar.computecanada.ca
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
X server Exit
Quick connect...
9. /home/mobaxterm
10. kevuoft@cedar.computecanada
/home/kevuoft/alpha_trading/bin/
Name Size (KB)
Sessions
Tools

Name	Size (KB)
..	
activate	2
activate.csh	1
activate.fish	2
activate_this.py	1

```
                    • MobaXterm Professional v21.0 •
                  (SSH client, X server and network tools)

 ➤ SSH session to kevuoft@cedar.computecanada.ca
   • Direct SSH      :  ✓
   • SSH compression :  ✓
   • SSH-browser     :  ✓
   • X11-forwarding  :  ✓  (remote display is forwarded through SSH)

 ➤ For more info, ctrl+click on help or visit our website.
```

Once you are in the new shell, run the command

ssh -L 8888:<hostname:port> <username>@<cluster>.computecanada.ca

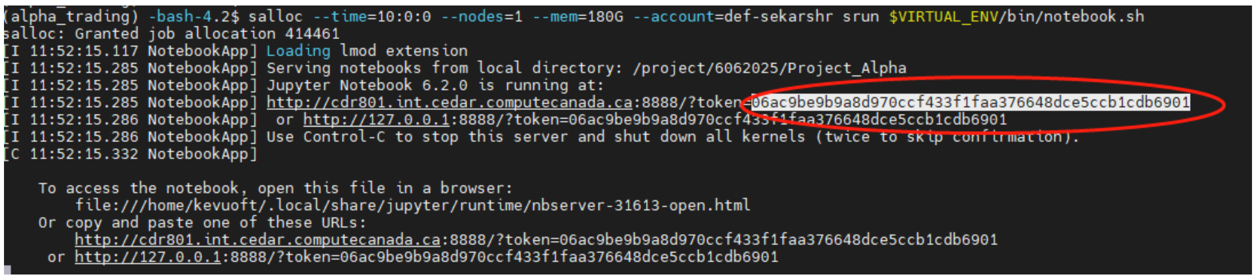


```
 21/04/2021   15:26.18   /home/mobaxterm  ssh -L 8888:cdr801.int.cedar.computecanada.ca:8888 kevuoft@cedar.computecanada.ca
Warning: Permanently added 'cedar.computecanada.ca' (RSA) to the list of known hosts.
Password:
Last login: Wed Apr 21 11:31:02 2021 from bras-base-toroon213qw-grc-14-174-89-155-45.dsl.bell.ca
================================================================================
Welcome to Cedar!

For information see: https://docs.computecanada.ca/wiki/Cedar
Email support@computecanada.ca for assistance and/or to report problems.
================================================================================

-bash-4.2$ 
```

Once you see the cluster's welcome message, you have successfully connected to the cluster!

Step 5 (Interactive)

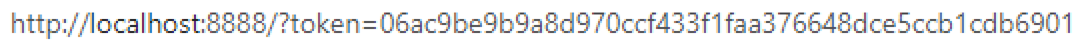
Go back to the previous bash window and copy the token. (Again, do not use control + C!)



```
(alpha_trading) -bash-4.2$ salloc --time=10:0:0 --nodes=1 --mem=180G --account=def-sekarshr srun $VIRTUAL_ENV/bin/notebook.sh
salloc: Granted job allocation 414461
[I 11:52:15.117 NotebookApp] Loading lmod extension
[I 11:52:15.285 NotebookApp] Serving notebooks from local directory: /project/6062025/Project_Alpha
[I 11:52:15.285 NotebookApp] Jupyter Notebook 6.2.0 is running at:
[I 11:52:15.285 NotebookApp] http://cdr801.int.cedar.computecanada.ca:8888/?token=06ac9be9b9a8d970ccf433f1faa376648dce5ccb1cdb6901
[I 11:52:15.286 NotebookApp]  or http://127.0.0.1:8888/?token=06ac9be9b9a8d970ccf433f1faa376648dce5ccb1cdb6901
[I 11:52:15.286 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 11:52:15.332 NotebookApp]

    To access the notebook, open this file in a browser:
        file:///home/kevuoft/.local/share/jupyter/runtime/nbserver-31613-open.html
    Or copy and paste one of these URLs:
        http://cdr801.int.cedar.computecanada.ca:8888/?token=06ac9be9b9a8d970ccf433f1faa376648dce5ccb1cdb6901
     or http://127.0.0.1:8888/?token=06ac9be9b9a8d970ccf433f1faa376648dce5ccb1cdb6901
```

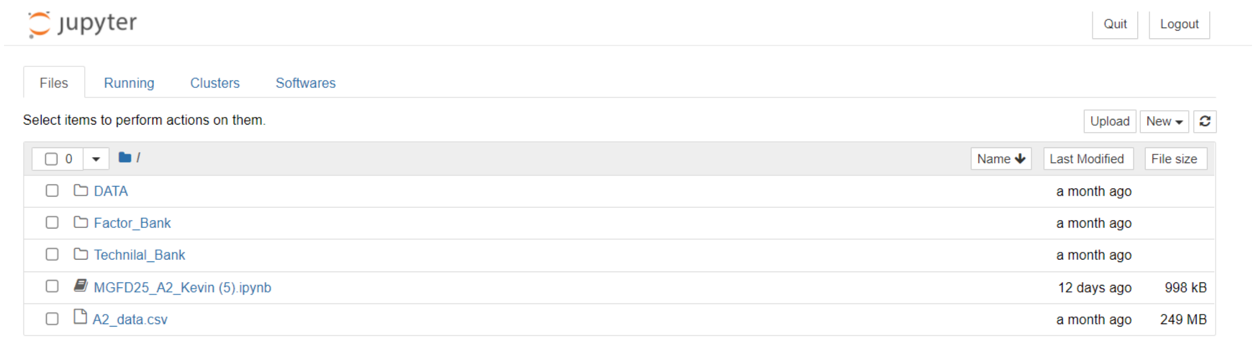
Open your browser and open your localhost port 8888

http://localhost:8888/?token=<token>



http://localhost:8888/?token=06ac9be9b9a8d970ccf433f1faa376648dce5ccb1cdb6901

Congratulations! You have connected the jupyter notebook on your localhost.



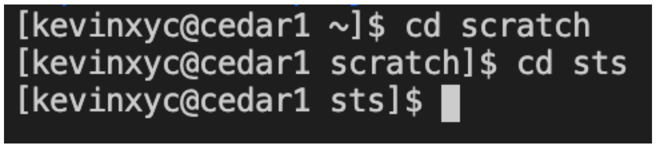
Select items to perform actions on them.

Name	Last Modified	File size
DATA	a month ago	
Factor_Bank	a month ago	
Technilal_Bank	a month ago	
MGFD25_A2_Kevin (5).ipynb	12 days ago	998 kB
A2_data.csv	a month ago	249 MB

Non-Interactive (Run <file>.py directly without interaction)

Step 1

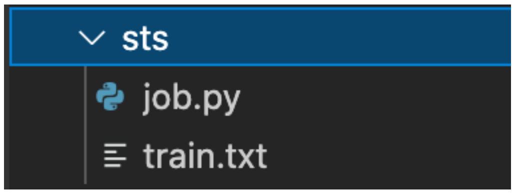
Get in the directory where the data is stored.



```
[kevinxyc@cedar1 ~]$ cd scratch
[kevinxyc@cedar1 scratch]$ cd sts
[kevinxyc@cedar1 sts]$
```

This is the directory as an example in the tutorial:

The python script for executing training is “job.py” and our data is train.txt.



```
sts
  job.py
  train.txt
```

Step 2

Create a bash script (.sh) file in the same directory, here is a template above called myjob.sh:

```
#!/bin/bash
#SBATCH --mem=32G
#SBATCH --cpus-per-task=2
#SBATCH --time=2:0:0
#SBATCH --account=def-sekarshr
#SBATCH --gres=gpu:1
source ~/venv/bin/activate
python job.py --data train.txt
```

Line 1 stays the same.

Line 2-6 is SBATCH configuration which can be modified to your own needs (mem is memory required, cpus-per-task is number of cpus used, time is the total time estimated, account stays the same, gres is number of gpus used).

Line 7 is sourcing into the virtual environment you created, in this case, change venv to the name of your virtual env.

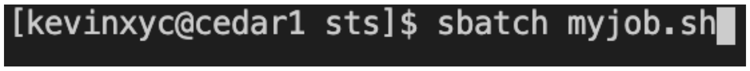
Line 8 is the line that execute your python script.

***If you want to output your results to an external file (say result.txt),** change line 8 to be:

```
python job.py --data train.txt > result.txt
```

Step 3

Lastly, execute the script file using sbatch and your job will be in queue for running



```
[kevinxyc@cedar1 sts]$ sbatch myjob.sh
```

After sbatching, you can use sq to check the status.

Once the job is done, sbatch will output some training results to slurm-{your job id}.out in the same directory such as slurm-1234566.out. We are done!

For more information, take a look at:

https://github.com/castorini/onboarding/blob/master/docs/cc-guide.md

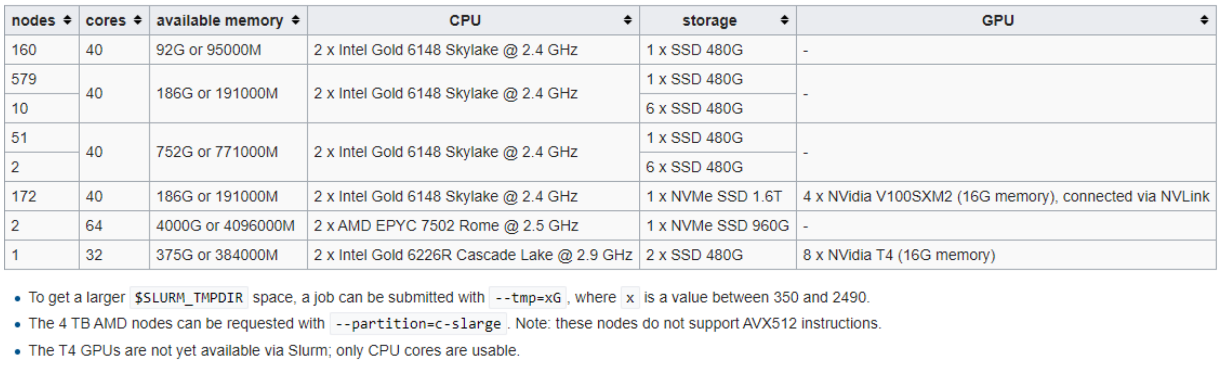
Appendix I

The followings are some of the commands you may want to use for cluster Beluga, Graham and Cedar:

Command	**Description**	**Example**
salloc	This command allows you to request for interactive jobs on the cluster.	
--time=<hh:m:s>	Mandatory	--time=10:0:0 Note that your job will get shut down whether the run-job is finished or not.
--account=def-sekarshr	Mandatory	
--nodes=<#nodes>	This will specify the number of nodes that you wish to use. For interactive jobs, this will always be 1 unless using multi-node configured GPU-Accelerating jobs	--nodes=1
--mem=<mem space>	This will specify the memory that you need.	--mem=180G Note that it is the best that the memory you request does not exceeds the memory of the node that you have requested for. (Requesting for only 100G will satisfy any node)
--constraint=<device name>	This will specify the type of cpu/gpu that you would like to use.	Using only Intel Skylake cpus: --constraint=skylake Using only Intel Broadwell cpus: --constraint=broadwell Specifying the constraint may take you a long time to wait.
--gres=gpu:<device name>:<#GPUs>	Specify the gpus that you request.	--gres=gpu:v100:4 --gres=gpu:p100:4 --gres=gpu:t4:4 Only use this command on Nodes that have GPU

		For GPU jobs, please prioritize the Mist Cluster.
srun $VIRTUAL_ENV/bin/notebook.sh	This will run the notebook.sh executable script in the bin. For interactive jobs on Begula, Cedar and Graham, you will have to always include this command at the end of the salloc request.	

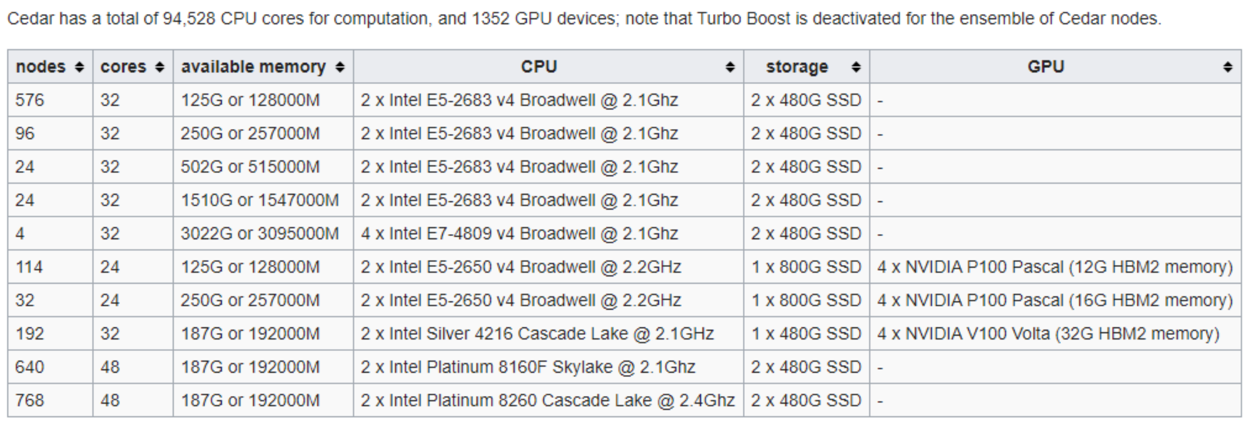
Appendix II



nodes	cores	available memory	CPU	storage	GPU
160	40	92G or 95000M	2 x Intel Gold 6148 Skylake @ 2.4 GHz	1 x SSD 480G	-
579	40	186G or 191000M	2 x Intel Gold 6148 Skylake @ 2.4 GHz	1 x SSD 480G	-
10				6 x SSD 480G	
51	40	752G or 771000M	2 x Intel Gold 6148 Skylake @ 2.4 GHz	1 x SSD 480G	-
2				6 x SSD 480G	
172	40	186G or 191000M	2 x Intel Gold 6148 Skylake @ 2.4 GHz	1 x NVMe SSD 1.6T	4 x NVidia V100SXM2 (16G memory), connected via NVLink
2	64	4000G or 4096000M	2 x AMD EPYC 7502 Rome @ 2.5 GHz	1 x NVMe SSD 960G	-
1	32	375G or 384000M	2 x Intel Gold 6226R Cascade Lake @ 2.9 GHz	2 x SSD 480G	8 x NVidia T4 (16G memory)

- To get a larger `$SLURM_TMPDIR` space, a job can be submitted with `--tmp=xG`, where `x` is a value between 350 and 2490.
- The 4 TB AMD nodes can be requested with `--partition=c-slarge`. Note: these nodes do not support AVX512 instructions.
- The T4 GPUs are not yet available via Slurm; only CPU cores are usable.

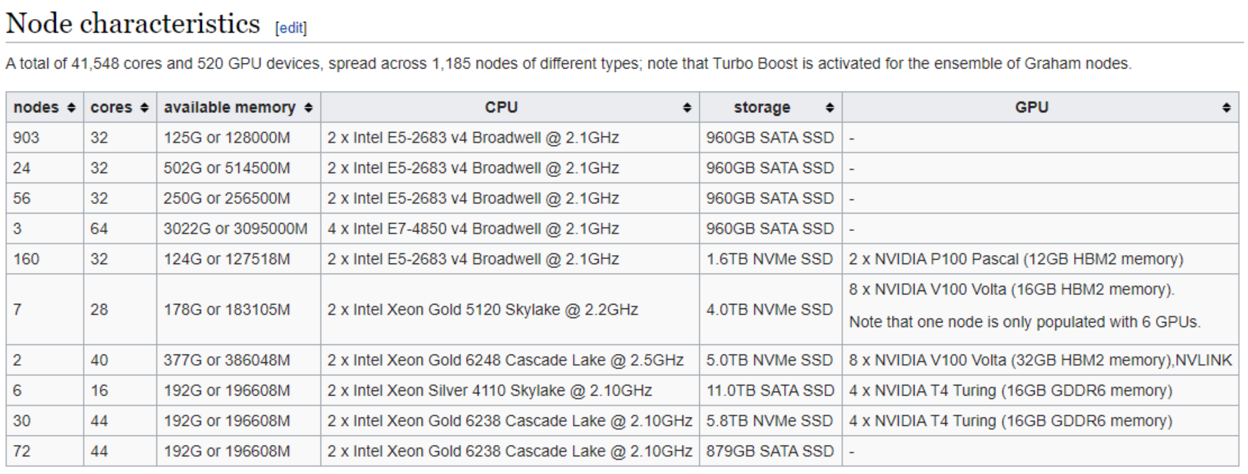
Cluster 1Begula



Cedar has a total of 94,528 CPU cores for computation, and 1352 GPU devices; note that Turbo Boost is deactivated for the ensemble of Cedar nodes.

nodes	cores	available memory	CPU	storage	GPU
576	32	125G or 128000M	2 x Intel E5-2683 v4 Broadwell @ 2.1Ghz	2 x 480G SSD	-
96	32	250G or 257000M	2 x Intel E5-2683 v4 Broadwell @ 2.1Ghz	2 x 480G SSD	-
24	32	502G or 515000M	2 x Intel E5-2683 v4 Broadwell @ 2.1Ghz	2 x 480G SSD	-
24	32	1510G or 1547000M	2 x Intel E5-2683 v4 Broadwell @ 2.1Ghz	2 x 480G SSD	-
4	32	3022G or 3095000M	4 x Intel E7-4809 v4 Broadwell @ 2.1Ghz	2 x 480G SSD	-
114	24	125G or 128000M	2 x Intel E5-2650 v4 Broadwell @ 2.2GHz	1 x 800G SSD	4 x NVIDIA P100 Pascal (12G HBM2 memory)
32	24	250G or 257000M	2 x Intel E5-2650 v4 Broadwell @ 2.2GHz	1 x 800G SSD	4 x NVIDIA P100 Pascal (16G HBM2 memory)
192	32	187G or 192000M	2 x Intel Silver 4216 Cascade Lake @ 2.1GHz	1 x 480G SSD	4 x NVIDIA V100 Volta (32G HBM2 memory)
640	48	187G or 192000M	2 x Intel Platinum 8160F Skylake @ 2.1Ghz	2 x 480G SSD	-
768	48	187G or 192000M	2 x Intel Platinum 8260 Cascade Lake @ 2.4Ghz	2 x 480G SSD	-

Cluster 2Cedar

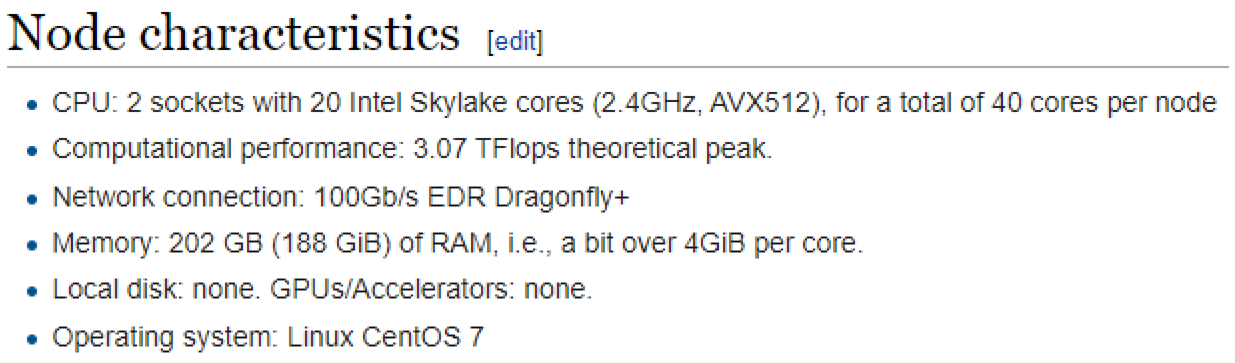


Node characteristics [edit]

A total of 41,548 cores and 520 GPU devices, spread across 1,185 nodes of different types; note that Turbo Boost is activated for the ensemble of Graham nodes.

nodes	cores	available memory	CPU	storage	GPU
903	32	125G or 128000M	2 x Intel E5-2683 v4 Broadwell @ 2.1GHz	960GB SATA SSD	-
24	32	502G or 514500M	2 x Intel E5-2683 v4 Broadwell @ 2.1GHz	960GB SATA SSD	-
56	32	250G or 256500M	2 x Intel E5-2683 v4 Broadwell @ 2.1GHz	960GB SATA SSD	-
3	64	3022G or 3095000M	4 x Intel E7-4850 v4 Broadwell @ 2.1GHz	960GB SATA SSD	-
160	32	124G or 127518M	2 x Intel E5-2683 v4 Broadwell @ 2.1GHz	1.6TB NVMe SSD	2 x NVIDIA P100 Pascal (12GB HBM2 memory)
7	28	178G or 183105M	2 x Intel Xeon Gold 5120 Skylake @ 2.2GHz	4.0TB NVMe SSD	8 x NVIDIA V100 Volta (16GB HBM2 memory). Note that one node is only populated with 6 GPUs.
2	40	377G or 386048M	2 x Intel Xeon Gold 6248 Cascade Lake @ 2.5GHz	5.0TB NVMe SSD	8 x NVIDIA V100 Volta (32GB HBM2 memory),NVLINK
6	16	192G or 196608M	2 x Intel Xeon Silver 4110 Skylake @ 2.10GHz	11.0TB SATA SSD	4 x NVIDIA T4 Turing (16GB GDDR6 memory)
30	44	192G or 196608M	2 x Intel Xeon Gold 6238 Cascade Lake @ 2.10GHz	5.8TB NVMe SSD	4 x NVIDIA T4 Turing (16GB GDDR6 memory)
72	44	192G or 196608M	2 x Intel Xeon Gold 6238 Cascade Lake @ 2.10GHz	879GB SATA SSD	-

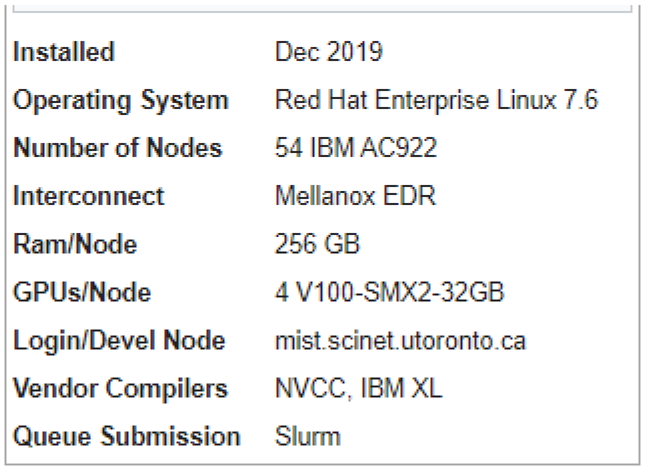
Cluster 3Graham



Node characteristics [edit]

- CPU: 2 sockets with 20 Intel Skylake cores (2.4GHz, AVX512), for a total of 40 cores per node
- Computational performance: 3.07 TFlops theoretical peak.
- Network connection: 100Gb/s EDR Dragonfly+
- Memory: 202 GB (188 GiB) of RAM, i.e., a bit over 4GiB per core.
- Local disk: none. GPUs/Accelerators: none.
- Operating system: Linux CentOS 7

Cluster 4Niagara



Installed	Dec 2019
Operating System	Red Hat Enterprise Linux 7.6
Number of Nodes	54 IBM AC922
Interconnect	Mellanox EDR
Ram/Node	256 GB
GPUs/Node	4 V100-SMX2-32GB
Login/Devel Node	mist.scinet.utoronto.ca
Vendor Compilers	NVCC, IBM XL
Queue Submission	Slurm

Cluster 5Mist

Appendix III

Cluster Begula, Graham and Cedar first time setup (Interactive)

```
module purge
module load openmpi
module load cuda
module load cudnn
module load nccl
module load python/3.8

virtualenv $HOME/alpha_trading
source ~/alpha_trading/bin/activate

pip install --no-index jupyter
echo -e '#!/bin/bash\nunset XDG_RUN TIME_DIR\njupyter notebook --ip $(hostname -f) --no-browser --port=8888 --NotebookApp.port_retries=0' > $VIRTUAL_ENV/bin/notebook.sh
chmod u+x $VIRTUAL_ENV/bin/notebook.sh
pip install jupyterlmod
jupyter nbextension enable --py jupyterlmod --sys-prefix
jupyter serverextension enable --py jupyterlmod --sys-prefix
pip install nbserverproxy
jupyter serverextension enable --py nbserverproxy --sys-prefix
pip install jupyter_http_over_ws
jupyter serverextension enable --py jupyter_http_over_ws
```

```
pip install --no-index tensorflow
pip install --no-index sklearn
pip install --no-index torch
pip install --no-index keras
pip install --no-index pandas
pip install --no-index numpy
pip install --no-index matplotlib
pip install --no-index plotly
pip install --no-index seaborn
pip install --no-index xgboost
pip install --no-index featuretools
```