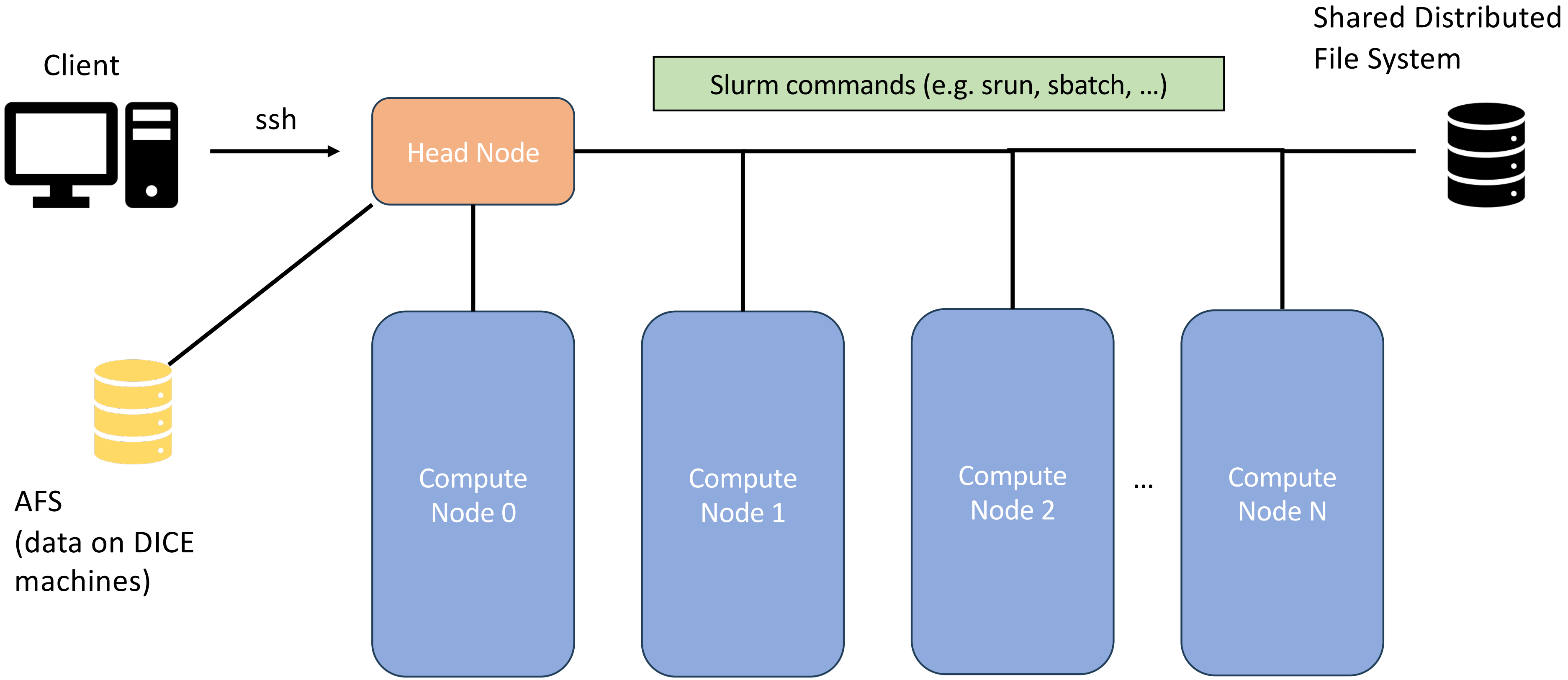
Teaching GPU Cluster

Teaching Cluster Overview



Teaching GPU Cluster

Basic Info:

The teaching cluster has over 120 GPUs in 20 servers (landonia[01-25]) with:

- 100 NVIDIA GTX 1060 6GB GPUs,
- 8 NVIDIA RTX A6000 48GB GPUs,
- a small number of NVIDIA TITAN-X 12GB GPUs.

Usage:

Access through **ssh** and use **SLURM** commands.

1. Access the cluster's head node via:

   - Connect to the Informatics VPN or use the DICE machine and then run:

     ```
     ssh <YOUR_UUN>@mlp.inf.ed.ac.uk
     ```

   - You can remotely connect to your DICE machine using the command **below** if you are connected to the Informatics VPN or the eduroam Wi-Fi. Once you have accessed the DICE machine, run the command mentioned **above**.

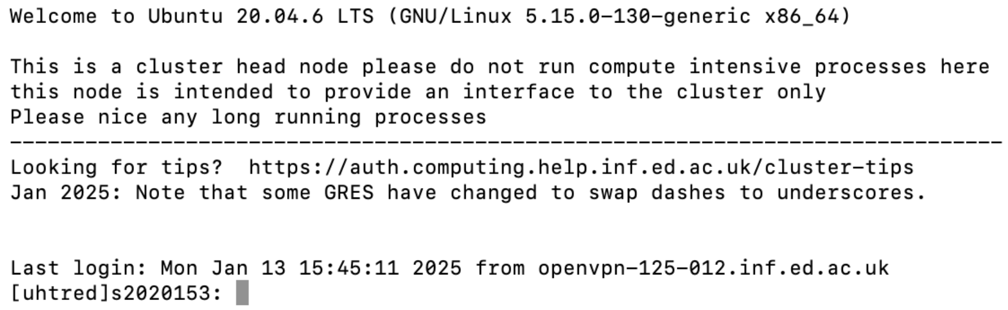
     ```
     ssh <YOUR_UUN>@ssh.inf.ed.ac.uk
     ```

   - You can find information about how to connect to Informatics VPN at: https://computing.help.inf.ed.ac.uk/openvpn

Teaching GPU Cluster

Usage (continue):

After successfully connecting to the head node, your terminal will display the following:



```
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.15.0-130-generic x86_64)

This is a cluster head node please do not run compute intensive processes here
this node is intended to provide an interface to the cluster only
Please nice any long running processes
--------------------------------------------------------------------------------
Looking for tips?  https://auth.computing.help.inf.ed.ac.uk/cluster-tips
Jan 2025: Note that some GRES have changed to swap dashes to underscores.


Last login: Mon Jan 13 15:45:11 2025 from openvpn-125-012.inf.ed.ac.uk
[uhtred]s2020153: 
```

2. Run your code with SLURM command:

There are two ways you can run your code:

- Interactive job - allow you to interact directly with the allocated compute resources, good for debugging and testing code.

- Batch job - run the task automatically based on a pre-written script (.sh file) without interaction with the compute node and constant user attention, good for long-running tasks.

Teaching GPU Cluster

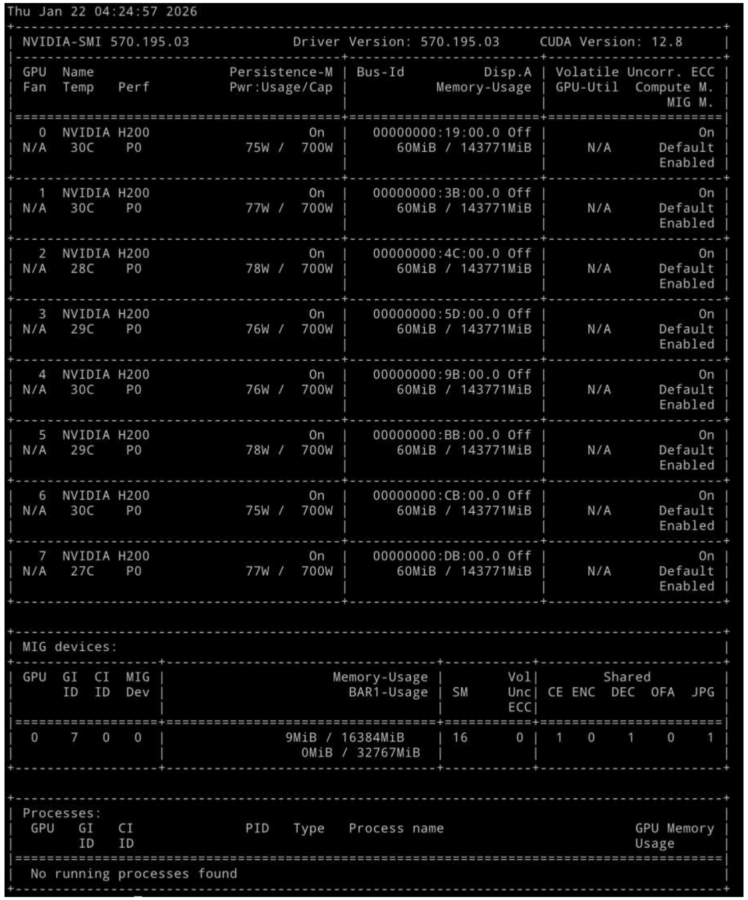
Usage (continue):

E.g. Let's say you want to run the code utilizing 1 GPU.

Interactive jobs:

```
srun -p Teaching -w saxa --gres gpu:1 --pty bash
```

After executing this command, run “nvidia-smi”, and you will see output similar to the following:



```
Thu Jan 22 04:24:57 2026
+-----------------------------------------------------------------------------------------+
| NVIDIA-SMI 570.195.03             Driver Version: 570.195.03     CUDA Version: 12.8     |
|-----------------------------------------+------------------------+----------------------+
| GPU  Name                 Persistence-M | Bus-Id          Disp.A | Volatile Uncorr. ECC |
| Fan  Temp   Perf          Pwr:Usage/Cap |           Memory-Usage | GPU-Util  Compute M. |
|                                         |                        |               MIG M. |
|=========================================+========================+======================|
|   0  NVIDIA H200                    On  |   00000000:19:00.0 Off |                   On |
| N/A   30C    P0             75W /  700W |      60MiB / 143771MiB |     N/A      Default |
|                                         |                        |              Enabled |
+-----------------------------------------+------------------------+----------------------+
|   1  NVIDIA H200                    On  |   00000000:3B:00.0 Off |                   On |
| N/A   30C    P0             77W /  700W |      60MiB / 143771MiB |     N/A      Default |
|                                         |                        |              Enabled |
+-----------------------------------------+------------------------+----------------------+
|   2  NVIDIA H200                    On  |   00000000:4C:00.0 Off |                   On |
| N/A   28C    P0             78W /  700W |      60MiB / 143771MiB |     N/A      Default |
|                                         |                        |              Enabled |
+-----------------------------------------+------------------------+----------------------+
|   3  NVIDIA H200                    On  |   00000000:5D:00.0 Off |                   On |
| N/A   29C    P0             76W /  700W |      60MiB / 143771MiB |     N/A      Default |
|                                         |                        |              Enabled |
+-----------------------------------------+------------------------+----------------------+
|   4  NVIDIA H200                    On  |   00000000:9B:00.0 Off |                   On |
| N/A   30C    P0             76W /  700W |      60MiB / 143771MiB |     N/A      Default |
|                                         |                        |              Enabled |
+-----------------------------------------+------------------------+----------------------+
|   5  NVIDIA H200                    On  |   00000000:BB:00.0 Off |                   On |
| N/A   29C    P0             78W /  700W |      60MiB / 143771MiB |     N/A      Default |
|                                         |                        |              Enabled |
+-----------------------------------------+------------------------+----------------------+
|   6  NVIDIA H200                    On  |   00000000:CB:00.0 Off |                   On |
| N/A   30C    P0             75W /  700W |      60MiB / 143771MiB |     N/A      Default |
|                                         |                        |              Enabled |
+-----------------------------------------+------------------------+----------------------+
|   7  NVIDIA H200                    On  |   00000000:DB:00.0 Off |                   On |
| N/A   27C    P0             77W /  700W |      60MiB / 143771MiB |     N/A      Default |
|                                         |                        |              Enabled |
+-----------------------------------------+------------------------+----------------------+

+-----------------------------------------------------------------------------------------+
| MIG devices:                                                                            |
+------------------+----------------------------------+-----------+-----------------------+
| GPU  GI  CI  MIG |                     Memory-Usage |        Vol|        Shared         |
|      ID  ID  Dev |                       BAR1-Usage | SM     Unc| CE ENC  DEC  OFA  JPG |
|                  |                                  |        ECC|                       |
|==================+==================================+===========+=======================|
|  0    7   0   0  |               9MiB / 16384MiB    | 16      0 |  1   0    1    0    1 |
|                  |               0MiB / 32767MiB    |           |                       |
+------------------+----------------------------------+-----------+-----------------------+

+-----------------------------------------------------------------------------------------+
| Processes:                                                                              |
|  GPU   GI   CI              PID   Type   Process name                        GPU Memory |
|        ID   ID                                                               Usage      |
|=========================================================================================|
|  No running processes found                                                             |
+-----------------------------------------------------------------------------------------+
```

Teaching GPU Cluster

Usage (continue):

E.g. Let's say you want to run the code utilizing 1 GPU.

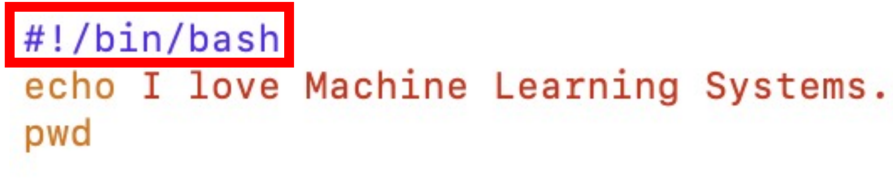
Batch jobs: `sbatch --gres=gpu:1 test.sh`

After executing this command, you will see “Submitted batch job [ID]”.

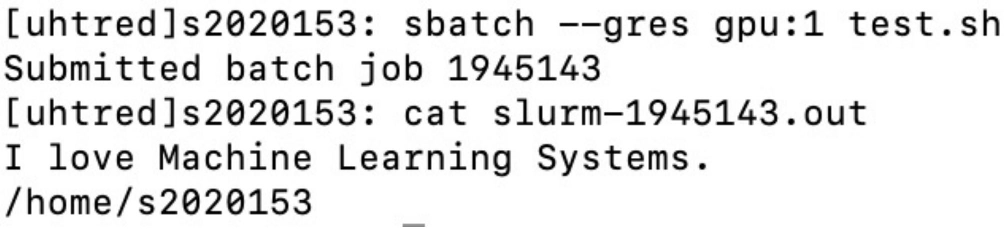
The output file will be saved as `slurm-[ID].out` by default in the directory where you call sbatch. You can set the output directory and file names by using the `--output` option with the `sbatch` command.

test.sh:

You must include this



```
#!/bin/bash
echo I love Machine Learning Systems.
pwd
```



```
[uhtred]s2020153: sbatch --gres gpu:1 test.sh
Submitted batch job 1945143
[uhtred]s2020153: cat slurm-1945143.out
I love Machine Learning Systems.
/home/s2020153
```

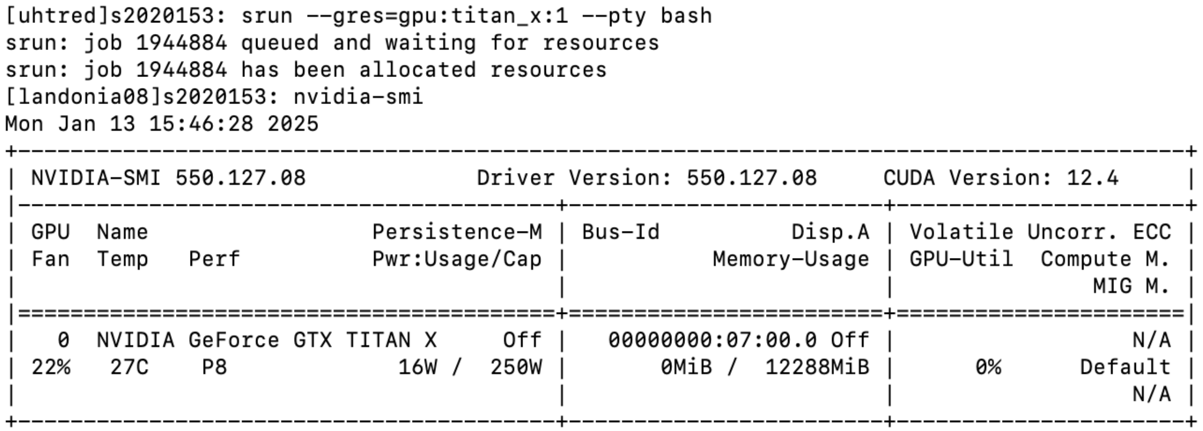
Teaching GPU Cluster

Usage (continue):

Since servers are equipped with different types of GPUs, you can specify the type of GPU you wish to use:

E.g. You want to have 1 NVIDIA Titan X GPU:

```
srun --gres=gpu:titan_x:1 --pty bash
sbatch --gres=gpu:titan_x:1 test.sh
```



```
[uhtred]s2020153: srun --gres=gpu:titan_x:1 --pty bash
srun: job 1944884 queued and waiting for resources
srun: job 1944884 has been allocated resources
[landonia08]s2020153: nvidia-smi
Mon Jan 13 15:46:28 2025
+-----------------------------------------------------------------------------------------+
| NVIDIA-SMI 550.127.08             Driver Version: 550.127.08     CUDA Version: 12.4     |
|-----------------------------------------+------------------------+----------------------+
| GPU  Name                 Persistence-M | Bus-Id          Disp.A | Volatile Uncorr. ECC |
| Fan  Temp   Perf          Pwr:Usage/Cap |           Memory-Usage | GPU-Util  Compute M. |
|                                         |                        |               MIG M. |
|=========================================+========================+======================|
|   0  NVIDIA GeForce GTX TITAN X     Off |   00000000:07:00.0 Off |                  N/A |
| 22%   27C    P8             16W /  250W |       0MiB /  12288MiB |      0%      Default |
|                                         |                        |                  N/A |
+-----------------------------------------+------------------------+----------------------+
```

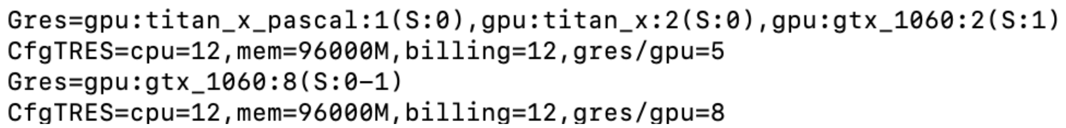
You are allowed to request a maximum of 8 GTX 1060/4 Titan X/1 Titan X Pascal GPU/2 A6000 GPUs at a time.
This year we have 8 NVIDIA H200 for you! (so I guess nobody want to use 1060 any more)
You can get access with `-p Teaching -w saxa` after srun/sbatch command.
NOTE: Please allocate resources according to your specific requirements. Avoid over-allocating to ensure resources are available for others!

Teaching GPU Cluster

Usage (continue):

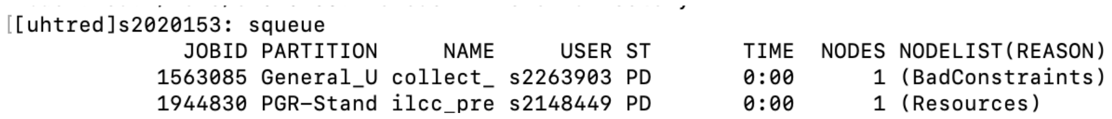
Other useful SLURM commands:

Check all available GPU types: `scontrol show node | grep gpu`



```
Gres=gpu:titan_x_pascal:1(S:0),gpu:titan_x:2(S:0),gpu:gtx_1060:2(S:1)
CfgTRES=cpu=12,mem=96000M,billing=12,gres/gpu=5
Gres=gpu:gtx_1060:8(S:0-1)
CfgTRES=cpu=12,mem=96000M,billing=12,gres/gpu=8
```

Check current SLURM job status: `squeue`



```
[uhtred]s2020153: squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
           1563085 General_U collect_ s2263903 PD       0:00      1 (BadConstraints)
           1944830 PGR-Stand ilcc_pre s2148449 PD       0:00      1 (Resources)
```

Cancel a job: `scancel <job_id>`

For more details about the school cluster and running SLURM commands, please refer to: https://computing.help.inf.ed.ac.uk/teaching-cluster.

IMPORTANT: Please read the Files and Backups section and GPU cluster tips in this link carefully, as it contains crucial information about file storage and can help prevent data loss.

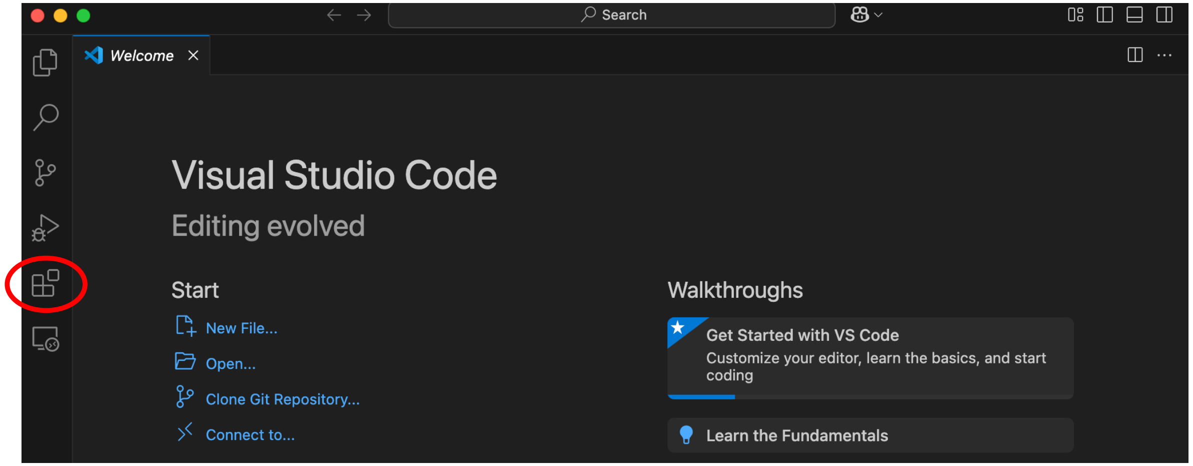
Teaching GPU Cluster

How to write codes in the teaching cluster?

VSCode!

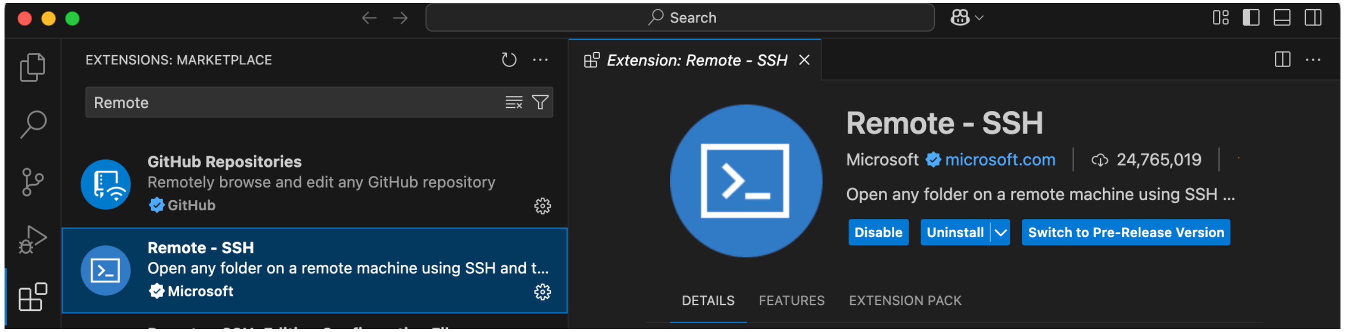
1. Open your VScode

2. Go to Extensions



Customize your editor, learn the basics, and start coding

3. Search for the keyword "Remote" and install the "Remote - SSH" extension.



Remotely browse and edit any GitHub repository

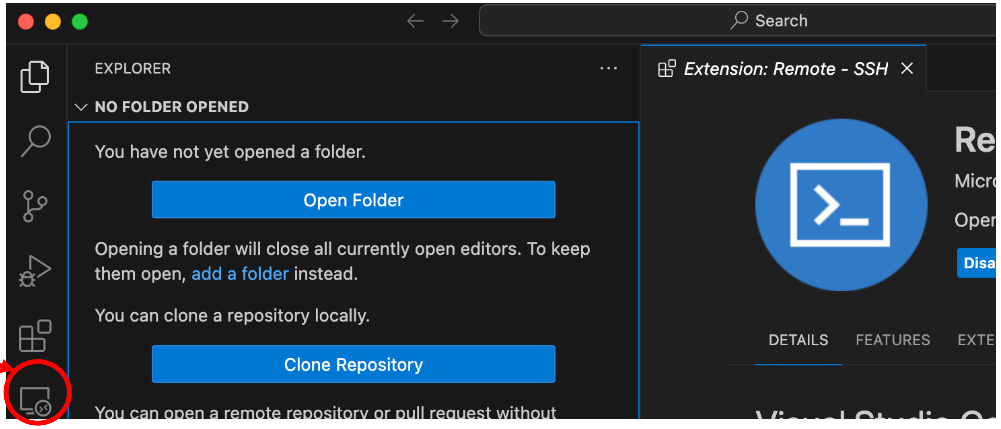
Open any folder on a remote machine using SSH and t...

Open any folder on a remote machine using SSH ...

Teaching GPU Cluster

How to write codes in the teaching cluster? (Continue)

4. Once the extension is installed, the REMOTES icon will appear on the left sidebar.



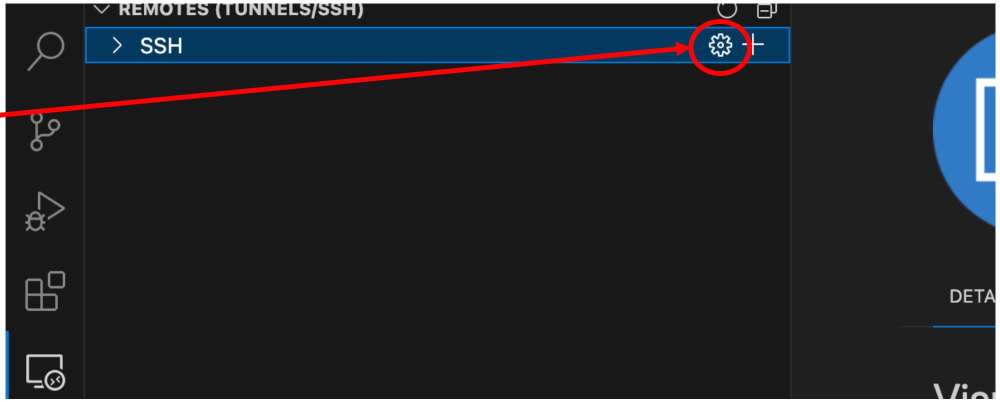
You have not yet opened a folder.

Opening a folder will close all currently open editors. To keep them open, add a folder instead.

You can clone a repository locally.

You can open a remote repository or pull request without

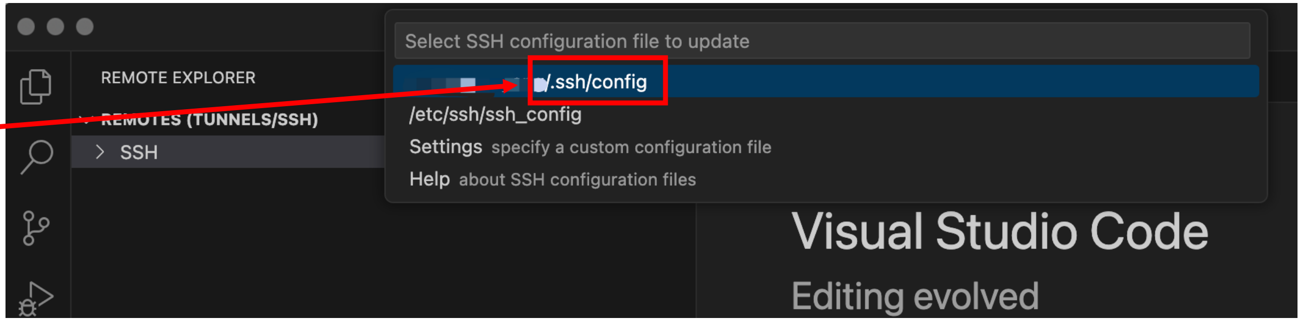
5. Click on the REMOTES icon, then select the Config icon located to the right of "SSH."



Teaching GPU Cluster

How to write codes in the teaching cluster? (Continue)

6. After clicking the Config icon, choose the configuration file with a path ending in .ssh/config.



Select SSH configuration file to update

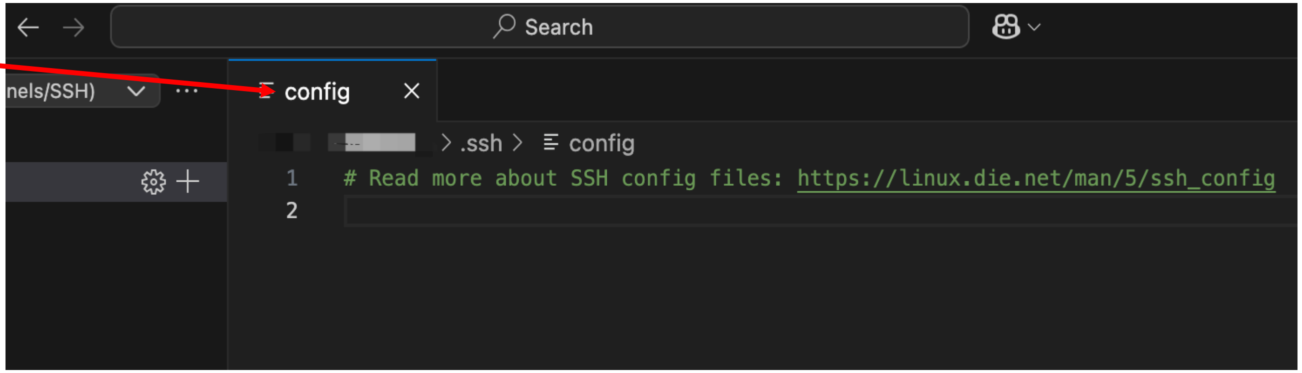
[redacted]/.ssh/config

/etc/ssh/ssh_config

Settings specify a custom configuration file

Help about SSH configuration files

7. You will see a page like this.



[redacted] > .ssh > config

```
# Read more about SSH config files: https://linux.die.net/man/5/ssh_config

```

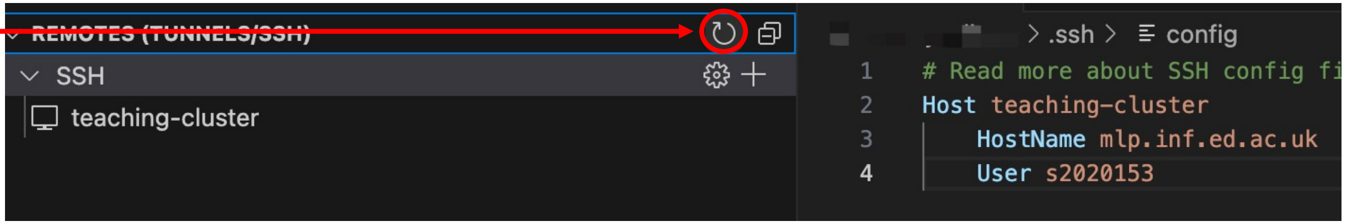
Teaching GPU Cluster

How to write codes in the teaching cluster? (Continue)

8. In this page, write the SSH configuration text and save it:

```
Host teaching-cluster
    HostName mlp.inf.ed.ac.uk
    User <YOUR_UUN>
```

9. Refresh the REMOTES page, and a screen similar to this will appear.

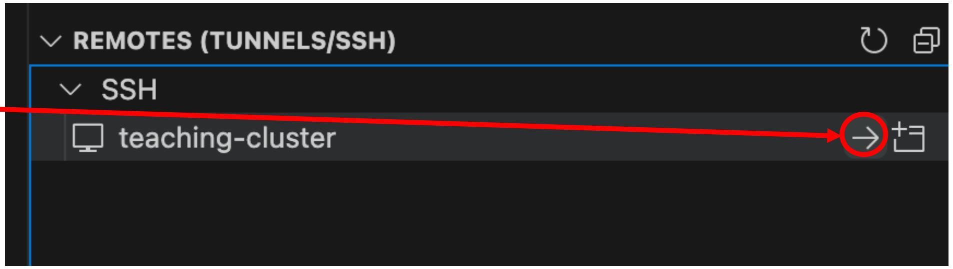


```
# Read more about SSH config fi
Host teaching-cluster
    HostName mlp.inf.ed.ac.uk
    User s2020153
```

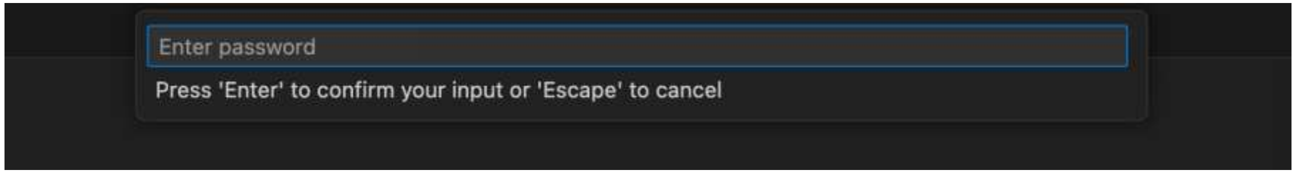
Teaching GPU Cluster

How to write codes in the teaching cluster? (Continue)

10. Connect to the Informatics VPN or Use the DICE machine, then click the right arrow icon to connect to the teaching cluster's head node:



11. Enter the password of your DICE account



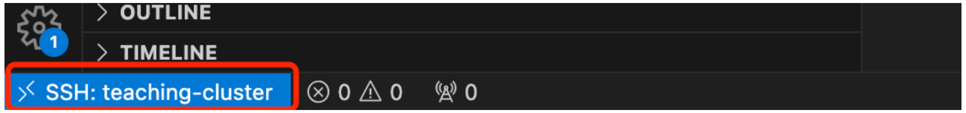
Enter password

Press 'Enter' to confirm your input or 'Escape' to cancel

Teaching GPU Cluster

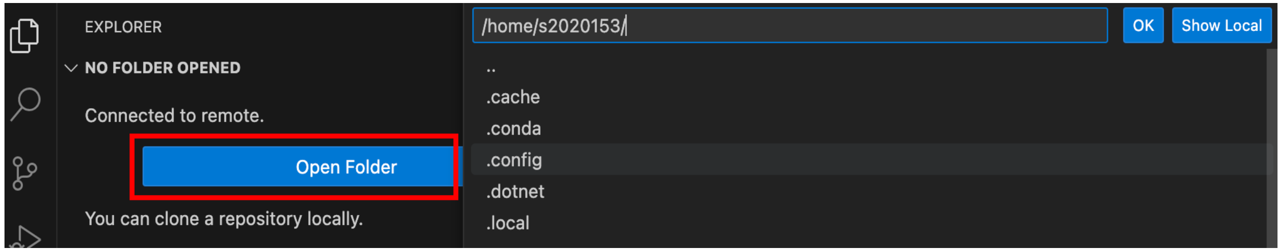
How to write codes in the teaching cluster? (Continue)

12. If this screen appears, congratulations! You have successfully connected to the teaching cluster's head node.



13. Go to your home directory by clicking ”Open Folder”, and select your home directory (/home/<YOUR_UUN>). Click OK.

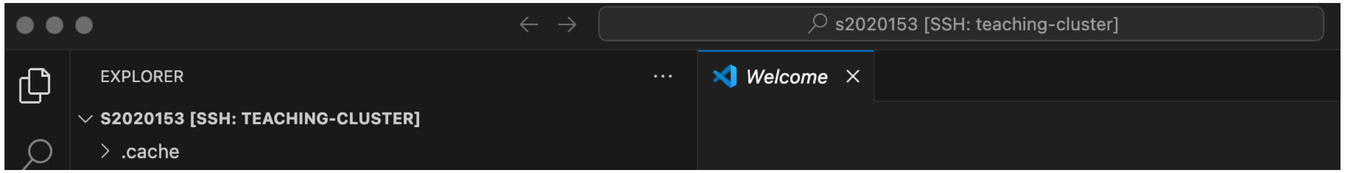
If password required, just type your password of the DICE account again.



Teaching GPU Cluster

How to write codes in the teaching cluster? (Continue)

14. If you can see this screen, you are now ready to write your code in the teaching cluster.


s2020153 [SSH: teaching-cluster]
EXPLORER
S2020153 [SSH: TEACHING-CLUSTER]
.cache
Welcome

IMPORTANT NOTES:

1. Any changes you make in VSCode, such as writing code, creating or deleting files and folders, will automatically synchronize with the cluster. That's why we recommend it ☺. **Do not forget to save your changes**

2. Only install your environment and write your code on the head node. Actions such as Git operations and similar tasks should also be performed on the head node, as they will not work on the compute nodes.

3. Only run your code on the compute nodes using `srun` or `sbatch`, as the head node has limited computing resources and does not have GPUs. Running `torch.cuda.is_available()` on the head node will return `False`.

If you have further issues about accessing the teaching cluster, please contact the Computing Support Team: https://computing.help.inf.ed.ac.uk/