# JGI server training

James Thomas · j.thomas@bristol.ac.uk

# Specifications

- see JGI data science handbook
  - 12 processor cores
  - 256 GB memory
  - NVIDIA GPU
  - ~1 TB SSD + ~4 TB HDD storage
- don’t assume it will be faster
- fair-use

# Logging in via SSH

- from the command line `ssh username@IT160062.users.bris.ac.uk`
- setting an SSH host alias and default user `~/.ssh/config`
- prevent lost sessions with `byobu-enable`
- EDGE F5 VPN when off-campus
- *advanced topics* - SSH keys, SEIS jump host

# Physical access

- monitor - switch input
- startup decryption passphrase
- GUI
- shutdown/restarting - takes a long time!

# Working with code and data

- use `/work` sub-directories (was `/data`)
  - shared first, this is backed up (to be tested!)
- transfer to/from local machine `scp -r`
- alternative tools: `rsync`, mount folder, FileZilla, VS Code
- data/code on the RDSF (also for `data.bris`)
  - see also JGI Training Project
  - set up in advance with machine user
- code on JGIBristol GitHub organisation

# Python environments

- conda environments (micromamba)
  - per-user (uses `~/.conda/NAME`)

```
conda env create -n NAME python=3.11 pandas
conda env create -n NAME -f environment.yml
conda activate NAME
```

  - shared (uses `/conda/envs/NAME`)

```
conda env create -p /conda/envs/NAME -f environment.yml
conda activate NAME
conda activate -p /conda/envs/NAME
```

# Python environments (2)

- testing the GPU

```
conda activate test_gpu
python -c "import tensorflow; print(tensorflow.config.list_physical_devices
/conda/gpu-test
```

# Jupyter notebooks

- using `jupyterlab`
- port forwarding:

```
ssh -L 8000:localhost:8000
...
conda activate NAME
jupyter lab --port=8000
```

# VS Code

- extensions:
  - Remote - SSH
  - Python and Jupyter (‘installed’ in remote)
- required Python packages:
  - `conda install ipykernel` (if not already installed)

# VS Code (2)

- opening a remote/folder
- finding your interpreter (`which python`)
- selecting an interpreter (first) and then kernel (second)
- check environment when using built-in terminal