Getting Started with MAGEO

Important information

Your homespace is currently limited to 10gb, while we may increase this in the future it is important to understand that the home directory `/home/jovyan` can be filled very quickly and accidentally in a few conda installs or data uploads. The homespace file system is not backed up, therefore, it is recommended that you regularly save code to GitHub (or your preferred system for version control of code) and keep copies of important data and outputs on your system. Contact NEODAAS if you need assistance transferring data, or if you require more store space on a scratch drive.

Getting started

Once you have confirmation from NEODAAS that your user has been added to MAGEOHub, go to https://mageohub.neodaas.ac.uk and log in using your GitHub credentials. When you enter jupyterlab you will be offered an instance with either 1 or 2 GPUS, we provide 10 cores and 64Gb RAM per GPU so if you know your application will take more memory or CPU time you can decide here. At any time, you can shut down your instance and reboot with larger or smaller resources with no effect on your stored files. Please select a single GPU for this course, to ensure there are enough GPU available for the other course participants.

Once first logged in you will need to set yourself up with some storage space. We limit users home directories (`/home/jovyan`) to 10 Gb to prevent system overload on our GPU nodes, but all nodes are connected to our 500Tb fast access storage. If you need access to the fast access lustre storage, please speak to NEODAAS and we can arrange this and provide a base of 1Tb, as a starting point.

It is important to note that this storage is not backed up. If you require your data to be backed up, please regularly download it, or contact NEODAAS to discuss data management options.

Using JupyterLab

MAGEOHub is built on a JupyterLab instance. If you are not familiar with JupyterLab or Jupyter Notebooks, then we recommend you spend a short time learning about how to use the Jupyter interface. They have a great page talking about the interface and how to effectively use JupyterLab. This is available here: https://jupyterlab.readthedocs.io/en/stable/user/interface.html.

JupyterLab is an environment for writing and developing scientific computing, and prototyping code. There is a basic tutorial available on MAGEOHub, on the storage drive. Copy this notebook over to your local drive and run the notebook. To copy the notebooks that will be used in this course, open a new terminal and run the command

```
cp -r /lustre_scratch/PORTWIMS-summer-school-2022 .
```

This will copy the training material over to your home directory.

Open the first notebook *Introduction to Jupyter* and run through it, learning the basics of JupyterLab.

Extension task with MAGEO

Useful for AI training

Run some GPU code

It's valuable to check what resources you have access to, this can be done using `nvidia-smi` in the terminal, or `!nvidia-smi` in a jupyter code block. You can also confirm access to GPU resources in a jupyter notebook using tensorflow, by first typing `import tensorflow as tf`, and then running `tf.config.list_physical_devices()` which will tell you capabilities and connected resources. Try these commands and have a look at the outputs, to understand what they are doing.

One thing to be aware of, is that you have one GPU attached to your session. This is shared between all notebooks that you have open. TensorFlow defaults to mapping almost all the GPU memory to each individual task, so if you try to work in one notebook, and then move to another notebook, the GPU memory may be fully occupied. This issue will present when running a notebook with some fit or predict command, that would usually use the GPU, and will fail with some difficult to interpret CUDA errors. The easiest way to solve this is to close the kernel of each notebook when you are finished working on that notebook and move onto the next one.

The environment

We have set up the MAGEOhub container with a base conda environment, full details of the contents are available using `conda list`, but some brief highlights are:

- TensorFlow https://www.tensorflow.org 2.4.1
- RAPIDS https://rapids.ai/start.html#get-rapids 0.19 We also provide netCDF, xarray and dask for data processing.

Generally, we would expect you to be able to work within this environment, however if you are unable to perform some action because of a useful missing library it is easy for us to add new modules. You can also create new conda environments by installing them into your home directory. More information on this is available on our GitHub, if this is something you need to do for your own project https://github.com/NEODAAS/MAGEO-docs#getting-started

Getting data onto and off MAGEOhub

As there is a fully featured terminal in MAGEOhub it is possible to use tools like `scp` and `wget` to download data onto your container. It is recommended you put any data in the lustre scratch folder you should have created above.

If these tools will not work, for instance if your data is stored in a location protected by a VPN, we can provide ftp servers to upload and download data. Generally, this would work by you uploading data to the NEODAAS FTP, then downloading using `ftp` on the container terminal, as the ftp server is on the same network as MAGEO you should see near instant speeds for most transfers.

Running code outside of MAGEO

As we have developed the MAGEOhub container ourselves we are able to distribute this as a docker or singularity image to ensure your code continues to operate as expected in new environments. Please ask NEODAAS for further information on this (helpdesk@neodaas.ac.uk).