

Jupyter Notebooks on Mistral

An easy way - useful if your data is not too large

1. login to mistral using the following command (replacing XXXXXX by your UserID):

```
ssh -X -L 8888:localhost:8888 mXXXXXX@mistral.dkrz.de
```

2. load anaconda using the following two commands:

```
module unload netcdf_c  
module load anaconda3/bleeding_edge
```

Note that the first of the two commands is necessary as netcdf_c otherwise causes a conflict with anaconda.

Note that you can also run these two commands automatically by adding them to your .bashrc file.

3. to start a jupyter notebook type the following command:

```
jupyter notebook --port=8888
```

This will cause a message to appear and to finally start your notebook you have to **copy and paste the ULR** to your browser.

If that doesn't work, try

```
jupyter notebook --port=8888 --no-browser
```

Note that this procedure causes problems if you have notebooks running locally on the same port, so you should close all other notebooks first. If you want to use local and remote notebooks at the same time, you can change one of the ports. For example: -port:8887 when you start your local notebook server.

What if you need to install additional packages?

1. first you need to create a virtual environment:

```
conda create -n yourenvname python=x.x anaconda
```

Note that you need to do this only once.

Note that yourenvname is the name you want to call your environment and X.X is the python version you want to use.

2. enter the virtual environment:

```
source activate yourenvname
```

3. install the package:

You can either use the standard distribution

```
conda install package_name
```

where package_name is the name of the package you want to install (e.g. numpy, scipy, etc.).

If the package is not available in the standard distribution, you can try to install the package from other sources using

```
conda install -c conda-forge package_name
```

For more detailed information see

<https://uoa-ereseach.github.io/ereseach-cookbook/recipe/2014/11/20/conda/>

A more comprehensive way - useful if your data is large

This follows the above steps where you have created your python environment (based primarily on instructions on the DKRZ website

<https://www.dkrz.de/up/systems/mistral/programming/jupyter-notebook>)

1. Create a file (jupyter_preload) which can activate this environment in your home directory of mistral.

```
export PATH = /work/.../miniconda3/bin:$PATH  
source activate yourenvname
```

2. download start-jupyter file from

<https://www.dkrz.de/up/systems/mistral/programming/jupyter-notebook>

3. store it on your local machine and change the file attribute to executable

```
chmod a+x start-jupyter
```

4. Execute on your local machine to start jupyter lab, then jupyter will pop up automatically on the local browser.

```
./start-jupyter.txt -u mXXXXXX -i jupyter_preload -c lab
```

This will run jupyter on the shared partition with one task.

You can also request a specific partition (e.g. compute) which is counted on a project when using jupyter, this is usually needed if you want to run big jobs.

```
./start-jupyter.txt -u mXXXXXX -i jupyter_preload -c lab -A project_id -p  
compute
```

From:

<https://wiki.mpimet.mpg.de/> - **MPI Wiki**

Permanent link:

https://wiki.mpimet.mpg.de/doku.php?id=analysis:pot_pourri:python:jupyter

Last update: **2020/10/20 16:54**

