

# Jupyter

Um Jupyter-Notebooks effizient und ohne Port-Forwarding im Cluster zu nutzen gibt es einen JupyterHub unter <https://jupyter.hpc.rz.uni-duesseldorf.de>

Dort können Jobs mit vordefinierten Ressourcen abgeschickt werden, innerhalb derer dann das Notebook gestartet wird.

Derzeit werden dort nur Shells und Python 3 - Notebooks angeboten, aber die Liste soll erweitert werden.

## Install packages

<https://jakevdp.github.io/blog/2017/12/05/installing-python-packages-from-jupyter/#How-to-use-Pip-from-the-Jupyter-Notebook>

## Install individual Python kernel

Jupyter allows you to work with your own environment. You can use e.g. conda for this task. Start by creating a new conda environment:

```
module purge    ##just in case you are using jupyter terminal
module load Miniconda/3.1
conda create -p /gpfs/project/$USER/py310 python=3.10
conda activate /gpfs/project/$USER/py310
```

Hint: this only works if you have defined a .condarc with channels pointing to our repo server (see also [Conda](#))

Install the programs that you need with conda install, at least ipykernel must be installed:

```
conda install ipykernel
```

Create a new file "kernel.sh" in the main directory of your environment and make it executable

```
cd /gpfs/project/$USER/py310
vi kernel.sh
```

Contents of the file kernel.sh:

```
#!/bin/bash
export PYTHONPATH=/gpfs/project/$USER/py310/lib/python3.10/site-packages
export PATH=/gpfs/project/$USER/py310/bin:$PATH
exec python -m ipykernel $@
```

Make the file executable with

```
chmod a+x kernel.sh
```

Create a new directory for your kernel in your /home/.local/share folder

```
mkdir -p /home/$USER/.local/share/jupyter/kernels/py310
cd /home/$USER/.local/share/jupyter/kernels/py310
```

Create a new file "kernel.json" with contents (!!replace \$USER with your explicit username here!!)

```
{
  "argv": [
    "/gpfs/project/$USER/py310/kernel.sh",
    "-f",
    "{connection_file}"
  ],
  "display_name": "Python 3.10 (conda)",
  "language": "python",
  "metadata": {
    "debugger": true
  }
}
```

In your next jupyterhub session a new kernel with the name "Python 3.10 (conda)" will then be available.

Hint: This seems to only work with Python versions < 3.11 !