Python (Conda/Jupyter/Etc.)

- Miniconda
- Miniforge
- JupyterHub

Miniconda

> The default installation of Miniconda may have restrictive licensing on some packages. The drop-in replacement "miniforge" will set up a similar minimal conda environment that defaults to the "conda-forge" channel of open packages.

As the packages in Python distributions can change rapidly, it is difficult to have a single, system-wide installation that is useful to everyone. We will continue to install some basic ones in /sopt, but they are not likely to change after initial build.

To let users customize a minimal Python environment that won't disturb others, we suggest installing the "Miniconda" version of the Anaconda Python distro. (See

https://conda.io/miniconda.html for additional info.)

Example Install 1) Python3-based conda installed in the default $HOME/miniconda3/ directory:

```
cd
wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh
chmod u+x Miniconda3-latest-Linux-x86_64.sh
./Miniconda3-latest-Linux-x86_64.sh -b
```

Example Install A) Same as option1, but installed to /DFS-L/DATA/$group/$user/miniconda3/ directory, with link in home directory (allows installing very large numbers of packages that would make your home directory go over quota):

```
cd
wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh
chmod u+x Miniconda3-latest-Linux-x86_64.sh
mkdir /DFS-L/DATA/$(id -gn)/$USER/miniconda3
ln -s /DFS-L/DATA/$(id -gn)/$USER/miniconda3 miniconda3
./Miniconda3-latest-Linux-x86_64.sh -bu
```

As of 23-November-2020, this installs Miniconda3 4.9.2, which uses Python 3.8.5. It also installs the minimal set of packages:

```
_libgcc_mutex-0.1-main
brotlipy-0.7.0-py38h27cfd23_1003
ca-certificates-2020.10.14-0
certifi-2020.6.20-pyhd3eb1b0_3
```

```
cffi-1.14.3-py38h261ae71_2
chardet-3.0.4-py38h06a4308_1003
conda-4.9.2-py38h06a4308_0
conda-package-handling-1.7.2-py38h03888b9_0
cryptography-3.2.1-py38h3c74f83_1
idna-2.10-py_0
ld_impl_linux-64-2.33.1-h53a641e_7
libedit-3.1.20191231-h14c3975_1
libffi-3.3-he6710b0_2
libgcc-ng-9.1.0-hdf63c60_0
libstdcxx-ng-9.1.0-hdf63c60_0
ncurses-6.2-he6710b0_1
openssl-1.1.1h-h7b6447c_0
pip-20.2.4-py38h06a4308_0
pycosat-0.6.3-py38h7b6447c_1
pycparser-2.20-py_2
pyopenssl-19.1.0-pyhd3eb1b0_1
pysocks-1.7.1-py38h06a4308_0
python-3.8.5-h7579374_1
readline-8.0-h7b6447c_0
requests-2.24.0-py_0
ruamel_yaml-0.15.87-py38h7b6447c_1
setuptools-50.3.1-py38h06a4308_1
six-1.15.0-py38h06a4308_0
sqlite-3.33.0-h62c20be_0
tk-8.6.10-hbc83047_0
tqdm-4.51.0-pyhd3eb1b0_0
urllib3-1.25.11-py_0
wheel-0.35.1-pyhd3eb1b0_0
xz-5.2.5-h7b6447c_0
yaml-0.2.5-h7b6447c_0
zlib-1.2.11-h7b6447c_3
```

This uses about 323MB of disk space.

Once installed, you can set up the environment paths to your private version (with either install option) using:

```
ml miniconda/3/own
```

Installing new packages (e.g. numpy) within your miniconda3 directory is as simple as:

```
conda install numpy
```

Miniconda2 can be installed in a similar way with:

```
cd
wget https://repo.continuum.io/miniconda/Miniconda2-latest-Linux-x86_64.sh
chmod u+x Miniconda2-latest-Linux-x86_64.sh
./Miniconda2-latest-Linux-x86_64.sh -b
```

There is also the corresponding miniconda/2/own module. Only one python module (miniconda/anaconda/python/Intel-python, all with versions 2 or 3) can be loaded at a time.

Miniforge

> The default installation of Miniconda may have restrictive licensing on some packages. The drop-in replacement "miniforge" will set up a similar minimal conda environment that defaults to the "conda-forge" channel of open packages.

As the packages in Python distributions can change rapidly, it is difficult to have a single, system-wide installation that is useful to everyone. We will continue to install some basic ones in /sopt, but they are not likely to change after initial build.

To let users customize a minimal Python environment that won't disturb others, we suggest installing the "Miniforge" version of the Conda Python environment. (See https://conda-forge.org/download/ for additional info.)

Example Install 1) Python3-based conda installed in the default $HOME/miniforge3/ directory:

```
cd
wget https://github.com/conda-forge/miniforge/releases/latest/download/Miniforge3-Linux-x86_64.sh
chmod u+x Miniforge3-Linux-x86_64.sh
./Miniforge3-Linux-x86_64.sh -b
```

Example Install A) Same as option1, but installed to /X2/SCRATCH/$group/$user/miniforge3/ directory, with link in home directory (allows installing very large numbers of packages that would make your home directory go over quota):

```
cd
wget https://github.com/conda-forge/miniforge/releases/latest/download/Miniforge3-Linux-x86_64.sh
chmod u+x Miniforge3-Linux-x86_64.sh
mkdir /X2/SCRATCH/$(id -gn)/$USER/miniforge3
ln -s /X2/SCRATCH/$(id -gn)/$USER/miniforge3 miniforge3
./Miniforge3-Linux-x86_64.sh -bu
```

As of 30-Marchr-2026, this installs Conda 26.1.1, which uses Python 3.13.12.

This uses about 552MB of disk space.

Once installed, you can set up the environment paths to your private version (with either install option) using:

```
ml miniforge/3/own
```

Updating will upgrade a few packages

```
conda update --all
```

Installing new packages (e.g. numpy) within your miniforge3 directory is as simple as:

```
conda install numpy
```

You can create a new environment for CPU-based PyTorch:

```
conda create --name torch-CPU pytorch torchvision torchaudio cpuonly -c pytorch
```

Or one for PyRosetta:

```
conda create --name pyrosetta -c https://conda.rosettacommons.org
```

Only one python module (miniforge/miniconda/anaconda/python/Intel-python) can be loaded at a time.

JupyterHub

While Jupyter notebooks can be run on any login node by installing the needed packages via conda or pip, running on compute nodes can be tricky. We have installed JupyterHub servers on gplogin2 and gplogin3 that can spawn JupyterLab sessions as slurm jobs.

Starting

Point your local computer's Web browser to either https://gplogin2.ps.uci.edu:8000 or https://gplogin3.ps.uci.edu:8000 and enter your Greenplanet username and password. Then, for the "simple" configuration:

1. Select which partition to run in
2. Select number of CPUs
3. Select which Jupyter environment to use (more info below)
4. If JupyterLab is desired, click box
5. Select runtime limit
6. Click Start

For "Advanced", you have much more control of each option. See https://github.com/silx-kit/jupyterhub_moss for detailed descriptions.

Jupyter environments

Default

The default Python environment (/sopt/JupyterHub/) is what is used to run the JupyterHub software. It is not user-customizable, but has a lot of general-use conda and pip packages and may just work. We can update and add more packages if needed at any time, so don't count on it having specific package versions.

Miniforge3

One of the pre-configured environments assumes you have miniforge3 set up in your home directory at ~/miniforge3 (this can also be a link to somewhere else, like /X2/SCRATCH/group/user/miniforge3). See https://knowledge.ps.uci.edu/books/greenplanet-software/page/miniforge for details on preparing that.

The miniforge/3 environment will use the base miniforge that is activated by loading the module with "module load miniforge/3". You will need to run "conda install jupyterhub jupyterlab batchspawner" to make sure the JupyterLab prerequisites are installed.

User JupyterHub

If you want to keep your conda base environment clean, you can create a custom environment at ~/jupyterhub through either conda or pip. For example, starting with miniforge3's conda, do this to install a custom environment that is stored in your directory on /X2:

```
mkdir /X2/SCRATCH/$(id -gn)/$USER/jupyterhub
ln -s /X2/SCRATCH/$(id -gn)/$USER/jupyterhub ~/jupyterhub
ml miniforge/3
conda create -y -p ~/jupyterhub batchspawner jupyterhub jupyterlab
conda activate ~/jupyterhub
conda install [list of needed packages for your notebook]
pip install [list of any needed packages not in conda]
```

Other Locations

If you need to run in another arbitrary python environment, you can enter the name, location, and any environment modules you want to load manually in the "Advanced" tab.