# The Conda Module

[Anaconda Python](https://www.anaconda.com/) is a very popular Python environment/distribution, which is made available via the centrally managed anaconda module. Typical usage of conda is to use it to manage/access multiple virtual environments, allowing the install of Python tools and packages that would otherwise conflict and be impossible to intall into the same location.

After module load anaconda you will have access to conda installed environments which are centrally maintained.

[john.hanks@cluster ~]$ module load anaconda  
[john.hanks@cluster ~]$ conda env list  
# conda environments:  
#  
base \* /hpc/apps/x86\_64/anaconda/2021\_09\_16  
griznog\_testing /hpc/apps/x86\_64/anaconda/2021\_09\_16/envs/griznog\_testing

!!! warning “Warning: conda init breaks things” Running conda init will add code to your ${HOME}/.bashrc file which will place conda installed python and dependent libraries in your ${PATH}. This will potetially break other applications which depend on the system intsalled python and libraries in many subtle ways.

# Self-managed conda envs

In addition to the centrally maintained virtual environments, it’s also possible to use the central anaconda module to self-manage personal or group virtual environments. While it’s possible and sometimes necessary to install the full Anaconda package, many times it’s sufficient to just create a personal env using the conda module described above. Example:

[john.hanks@cluster ~]$ module load anaconda  
[john.hanks@cluster ~]$ conda env list  
# conda environments:  
#  
base \* /hpc/apps/x86\_64/anaconda/2021\_09\_16  
griznog\_testing /hpc/apps/x86\_64/anaconda/2021\_09\_16/envs/griznog\_testing  
  
[john.hanks@cluster ~]$ conda create -n mycondaenv python=3.6  
Collecting package metadata (current\_repodata.json): done  
Solving environment: done  
  
## Package Plan ##  
  
 environment location: /home/john.hanks/.conda/envs/mycondaenv  
  
 added / updated specs:  
 - python=3.6  
  
  
The following packages will be downloaded:  
  
 package | build  
 ---------------------------|-----------------  
 ca-certificates-2021.9.30 | h06a4308\_1 116 KB  
   
 [...]  
   
 tk-8.6.11 | h1ccaba5\_0 3.0 MB  
 ------------------------------------------------------------  
 Total: 38.3 MB  
  
The following NEW packages will be INSTALLED:  
  
 \_libgcc\_mutex pkgs/main/linux-64::\_libgcc\_mutex-0.1-main  
   
 [...]  
   
 zlib pkgs/main/linux-64::zlib-1.2.11-h7b6447c\_3  
  
  
Proceed ([y]/n)? y  
  
  
Downloading and Extracting Packages  
tk-8.6.11 | 3.0 MB | ######################################################################################### | 100%   
  
[...]  
  
ca-certificates-2021 | 116 KB | ######################################################################################### | 100%   
Preparing transaction: done  
Verifying transaction: done  
Executing transaction: done  
#  
# To activate this environment, use  
#  
# $ conda activate mycondaenv  
#  
# To deactivate an active environment, use  
#  
# $ conda deactivate  
  
[john.hanks@cluster ~]$ conda env list  
# conda environments:  
#  
mycondaenv /home/john.hanks/.conda/envs/mycondaenv  
base \* /hpc/apps/x86\_64/anaconda/2021\_09\_16  
griznog\_testing /hpc/apps/x86\_64/anaconda/2021\_09\_16/envs/griznog\_testing

As you can see I now have a conda env of my own based off the centrally installed anaconda. To make this visible in the OnDemand Jupyter Notebook/Lab environments, I simply need to add the nb\_conda package to it:

[john.hanks@cluster ~]$ conda activate mycondaenv  
(mycondaenv) [john.hanks@cluster ~]$ conda install nb\_conda  
Collecting package metadata (current\_repodata.json): done  
Solving environment: failed with initial frozen solve. Retrying with flexible solve.  
Solving environment: failed with repodata from current\_repodata.json, will retry with next repodata source.  
Collecting package metadata (repodata.json): done  
Solving environment: done  
  
## Package Plan ##  
  
 environment location: /home/john.hanks/.conda/envs/mycondaenv  
  
 added / updated specs:  
 - nb\_conda  
  
  
The following packages will be downloaded:  
  
 package | build  
 ---------------------------|-----------------  
 argon2-cffi-20.1.0 | py36h27cfd23\_1 46 KB  
   
 [...]  
   
 zipp-3.6.0 | pyhd3eb1b0\_0 17 KB  
 ------------------------------------------------------------  
 Total: 8.8 MB  
  
The following NEW packages will be INSTALLED:  
  
 argon2-cffi pkgs/main/linux-64::argon2-cffi-20.1.0-py36h27cfd23\_1  
   
 [...]  
   
 zipp pkgs/main/noarch::zipp-3.6.0-pyhd3eb1b0\_0  
  
The following packages will be UPDATED:  
  
 certifi 2020.12.5-py36h06a4308\_0 --> 2021.5.30-py36h06a4308\_0  
  
  
Proceed ([y]/n)? y  
  
  
Downloading and Extracting Packages  
notebook-6.4.3 | 4.2 MB | ######################################################################################### | 100%   
  
[...]  
  
markupsafe-2.0.1 | 21 KB | ######################################################################################### | 100%   
Preparing transaction: done  
Verifying transaction: done  
Executing transaction: / Enabling nb\_conda\_kernels...  
CONDA\_PREFIX: /home/john.hanks/.conda/envs/mycondaenv  
Status: enabled  
  
- Config option `kernel\_spec\_manager\_class` not recognized by `EnableNBExtensionApp`.  
Enabling notebook extension nb\_conda/main...  
 - Validating: OK  
Enabling tree extension nb\_conda/tree...  
 - Validating: OK  
Config option `kernel\_spec\_manager\_class` not recognized by `EnableServerExtensionApp`.  
Enabling: nb\_conda  
- Writing config: /home/john.hanks/.conda/envs/mycondaenv/etc/jupyter  
 - Validating...  
 nb\_conda 2.2.1 OK  
  
done  
(mycondaenv) [john.hanks@cluster ~]$