Managing Python Environments at OSC

Shameema Oottikkal

Ohio Supercomputer Center
An OH·TECH Consortium Member

Outline

- How to access Python at OSC
- Module files for Python versions
- Python environment
- Anaconda vs miniconda
- Pip vs conda
- Install, uninstall and upgrade packages
- Interactive Python with Jupyter

How to access python on OSC clusters

On Pitzer cluster:

```
[soottikkaltest@pitzer-login01 ~]$ python
Python 2.7.5 (default, May 27 2022, 11:27:32)
[GCC 4.8.5 20150623 (Red Hat 4.8.5-44)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> exit()
[soottikkaltest@pitzer-login01 ~]$ which python
/usr/bin/python
```

On Ascend cluster:

```
[soottikkaltest@ascend-login01 ~]$ which python
/usr/bin/python
[soottikkaltest@ascend-login01 ~]$ python
Python 3.6.8 (default, Apr 28 2022, 06:08:06)
[GCC 8.4.1 20200928 (Red Hat 8.4.1-1)] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

- These are installed by system group
- Older versions and cannot be updated

How to get newer versions of python on OSC systems

```
[soottikkaltest@pitzer-login01 ~]$ module avail python

------------------------------------------------------------------------ /apps/lmodfiles/Core --
   python/2.7-conda5.2    python/3.6-conda5.2 (D)    python/3.7-2019.10    python/3.9-2022.05

  Where:
   D:  Default Module

Use "module spider" to find all possible modules.
Use "module keyword key1 key2 ..." to search for all possible modules matching any of the "keys".
```

Versions

Python is available on Pitzer and Owens Clusters. The versions currently available at OSC are:

Version	Owens	Pitzer	Ascend	Notes
2.7	X			
3.5	X			
3.6	X			
2.7-conda5.2	X	X		Anaconda 5.2 with Python 2.7
3.6-conda5.2	X*	X*		Anaconda 5.2 with Python 3.6
3.7-2019.10	X	X		Anaconda 2019.10 with Python 3.7
3.9-2022.05	X	X		Anaconda 2022.05 with Python 3.9
3.9			X*	

* Current default version; A = installed as an integrated package Anaconda

Loading Python module

`module load python` or `module load python/3.9-2022.05`

```
[soottikkaltest@pitzer-login02 ~]$ ml python/3.9-2022.05
[soottikkaltest@pitzer-login02 ~]$ python
Python 3.9.12 (main, Apr  5 2022, 06:56:58)
[GCC 7.5.0] :: Anaconda, Inc. on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import numpy
>>> print(numpy.__file__)
/apps/python/3.9-2022.05/lib/python3.9/site-packages/numpy/__init__.py
```

```
[soottikkaltest@pitzer-login01 ~]$ module show python/3.9-2022.05
-------------------------------------------------------------------
   /apps/lmodfiles/Core/python/3.9-2022.05.lua:
-------------------------------------------------------------------
whatis("loads python")
help([[This module loads python
Configured and installed with modules:
No modules loaded]])
setenv("OSC_PYTHON_DIR","/apps/python/3.9-2022.05")
family("python")
setenv("PYTHON_HOME","/apps/python/3.9-2022.05")
prepend_path("PATH","/apps/python/3.9-2022.05/bin")
prepend_path("LD_LIBRARY_PATH","/apps/python/3.9-2022.05/lib")
prepend_path("MANPATH","/apps/python/3.9-2022.05/share/man")
setenv("TCL_LIBRARY","/apps/python/3.9-2022.05/lib/tcl8.6")
setenv("TK_LIBRARY","/apps/python/3.9-2022.05/lib/tk8.6")
```

If you need a package that is not available in the module

```
[soottikkaltest@pitzer-login02 ~]$ python
Python 3.9.12 (main, Apr  5 2022, 06:56:58)
[GCC 7.5.0] :: Anaconda, Inc. on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import yt
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ModuleNotFoundError: No module named 'yt'
>>> exit()
[soottikkaltest@pitzer-login02 ~]$ pip install yt
```

```
[soottikkaltest@pitzer-login02 ~]$ pip show yt
Name: yt
Version: 4.1.4
Summary: An analysis and visualization toolkit for volumetric data
Home-page: https://github.com/yt-project/yt
Author: The yt project
Author-email: yt-dev@python.org
License: BSD 3-Clause
Location: /users/PZS1118/soottikkaltest/.local/lib/python3.9/site-packages
Requires: ipywidgets, more-itertools, tomli, numpy, cmyt, unyt, pillow, pypa
Required-by:
```

- If you install yt/4.1.0, yt/4.1.4 will be uninstalled
- Create a Python environment to avoid conflicts

Installing Python packages locally

While our Python installations come with many popular packages installed, users may have a case in which they need an additional package that is not installed.

Step 1: Load proper Python module

```
module load python/3.9-2022.05 or module load miniconda3/4.12.0-py39
```

```
-bash-4.2$ module avail miniconda

------------------------------------------------------------------------------ /apps/lmodfiles/Core --
   miniconda3/4.10.3-py37 (D)    miniconda3/4.12.0-py38    miniconda3/4.12.0-py39

  Where:
   D:  Default Module

Use "module spider" to find all possible modules.
Use "module keyword key1 key2 ..." to search for all possible modules matching any of the "keys".
```

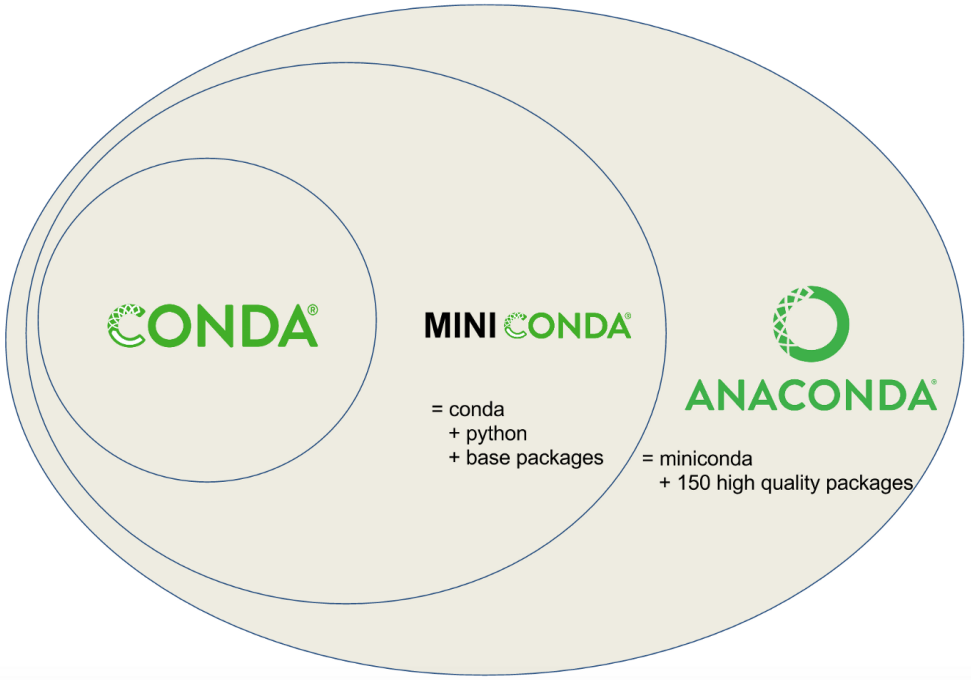
Anaconda vs Miniconda

Anaconda and Miniconda are both Python distributions that come with a package manager called Conda.

Conda is a powerful tool that allows you to install, update, and remove Python packages.

Anaconda is a more comprehensive distribution than Miniconda. It comes with over 150 pre-installed packages, including many popular data science libraries such as NumPy, SciPy, and Pandas. This makes it a good choice for beginners who want to get started with data science quickly.

Miniconda is a smaller, more lightweight distribution than Anaconda. It only comes with Conda and a few other essential packages. This makes it a good choice for experienced users who want to have more control over the packages they install.



Source: Planemo documentation

Choose Anaconda if you:

- Are new to conda or Python
- Like the convenience of having Python and several scientific packages automatically installed at once
- Have the time and disk space (a few minutes and 3 GB), and/or
- Don’t want to install each of the packages you want to use individually.

Choose Miniconda if you:

- Do not mind installing each of the packages you want to use individually.
- Do not have time or disk space to install several packages
- Just want fast access to Python and the conda commands

If you want to create your own python environment, we recommend using miniconda3 module, since you can start with minimal configurations.

```
Module load miniconda3/4.12.0-py39
```

Step 2: Create new environment

The following will create a minimal Python installation without any extraneous packages:

```
conda create -n myenv
```

If you want to clone the full base Python environment from the system, you may use the following create command:

```
conda create -n myenv --clone base
```

You can augment the command above by listing specific packages you would like installed into the environment. For example, the following will create a minimal Python installation with only the specified packages (in this case, numpy and babel):

```
conda create -n myenv  numpy babel
```

Specific versions can be specified by adding =<version> after the package name. For example, the following will create a Python installation with Python version 2.7 and NumPy version 1.16:

```
conda create -n myenv python=3.6 numpy=1.16
```

```
-bash-4.2$ conda create -n myenv
Collecting package metadata (current_repodata.json): done
Solving environment: done


==> WARNING: A newer version of conda exists. <==
  current version: 4.12.0
  latest version: 23.3.1

Please update conda by running

    $ conda update -n base -c defaults conda



## Package Plan ##

  environment location: /users/PZS0680/soottikkal/.conda/envs/myenv



Proceed ([y]/n)? y

Preparing transaction: done
Verifying transaction: done
Executing transaction: done
#
# To activate this environment, use
#
#     $ conda activate myenv
#
# To deactivate an active environment, use
#
#     $ conda deactivate
```

Do not use this, but use source activate instead

```
-bash-4.2$ conda remove -n myenv --all

Remove all packages in environment /users/PZS0680/soottikkal/.conda/envs/myenv:

No packages found in /users/PZS0680/soottikkal/.conda/envs/myenv. Continuing environment removal
```

To verify that a clone has been created, use the command

```
conda info -e
```

Step 3: Activate environment

Before the created environment can be used, it must be activated

```
-bash-4.2$ source activate myenv
(myenv) -bash-4.2$
```

Step 4: Install packages

To install additional packages, use the conda install command. For example, to install the numpy package:

```
conda install numpy
```

```
>>> import numpy
>>> print(numpy.__file__)
/users/PZS0680/soottikkal/.conda/envs/myenv/lib/python3.10/site-packages/numpy/__init__.py
```

By default, conda will install the latest version if the package that it can find. Specific versions can be specified by adding =<version> after the package name. For example, to install version 1.16 of the NumPy package:

```
conda install numpy=1.16
```

Searching for packages

To see installed packages:

```
conda list
```

To see if a specific package, such as SciPy, is available for installation:

```
conda search scipy
```

```
# Name                       Version           Build               Channel
scipy                         0.17.1 np110py27_blas_openblas_200  conda-forge
scipy                         0.17.1 np110py27_blas_openblas_201  conda-forge
scipy                          1.1.0  py35h9b217d5_1               pkgs/main
scipy                          1.1.0  py35hd20e5f9_0               pkgs/main
scipy                          1.1.0  py35he2b7bc3_1               pkgs/main
```

To see if a specific package, such as SciPy, is available for installation from Anaconda.org:

```
conda search --override-channels --channel defaults scipy
```

To see if a specific package, such as scipy, exists in a specific channel, such as http://conda.anaconda.org/intel, and is available for installation:

```
conda search --override-channels --channel http://conda.anaconda.org/intel  scipy
```

To install multiple packages at once, such as SciPy and cURL:

```
conda install scipy curl
```

To install a package without activating an environment

```
conda install --name myenv scipy
```

Updating packages

Use `conda update` command to check to see if a new update is available. If conda tells you an update is available, you can then choose whether or not to install it.

- To update a specific package:
  `conda update scipy`
- To update Python:
  `conda update python`
- To update conda itself:
  `source deactivate -n myenv`
  `conda update conda`

Removing Packages

Use the terminal or an Anaconda Prompt for the following steps.

- To remove a package such as SciPy in an environment such as myenv:
```
conda remove -n myenv scipy
```

- To remove a package such as SciPy in the current environment:
```
conda remove scipy
```

- To remove multiple packages at once, such as SciPy and cURL:
```
conda remove scipy curl
```

- To confirm that a package has been removed:
```
conda list
```

pip: python package manager

pip is a package manager for Python. That means it's a tool that allows you to install and manage libraries and dependencies that aren't distributed as part of the standard library.

```
(myenv) -bash-4.2$ pip install camelcase
Collecting camelcase
  Using cached camelcase-0.2.tar.gz (1.3 kB)
  Preparing metadata (setup.py) ... done
Building wheels for collected packages: camelcase
  Building wheel for camelcase (setup.py) ... done
  Created wheel for camelcase: filename=camelcase-0.2-py3-none-any.whl size=1772 sha256=c6d134159a
  Stored in directory: /users/PZS0680/soottikkal/.cache/pip/wheels/89/69/2b/bb16b6766a74942bfbd1e0
Successfully built camelcase
Installing collected packages: camelcase
Successfully installed camelcase-0.2
(myenv) -bash-4.2$ python
Python 3.10.8 (main, Mar 21 2023, 00:22:10) [GCC 11.2.0] :: Intel Corporation on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import camelcase
>>> print(camelcase.__file__)
/users/PZS0680/soottikkal/.conda/envs/myenv/lib/python3.10/site-packages/camelcase/__init__.py
>>>
```

The pip install <package> command always looks for the latest version of the package and installs it. It also searches for dependencies listed in the package metadata and installs them to ensure that the package has all the requirements that it needs.

To uninstall a package

```
pip uninstall camelcase
```

If pipis not available in conda environment, install it as

```
conda install pip
```

Point of Difference	pip	Conda
Multi-Language Dependency	Not Supported	Supported
Package Installation	Build on wheels	Download binary
Package Availability	235,000 packages	1,500+ pacakges
Dependency Management	No SAT test	performs SAT test
Virtual Environment Management	No in-built Virutal management	In-built virtual management system
Minimalistic	yes	no

Pip in virtual environment

Currently, there are two common tools for creating Python virtual environments:

- venv is available by default in Python 3.3 and later, and installs pip and setuptools into created virtual environments in Python 3.4 and later.
- virtualenv needs to be installed separately, but supports Python 2.7+ and Python 3.3+, and pip, setuptools and wheel are always installed into created virtual environments by default (regardless of Python version).

Example: create an environment using venv

```
python -m venv pipenv
```

Activate the environment

```
source pipenv/bin/activate
```

Using pip in virtual environments

To see pip commands:

```
pip help
```

```
Usage:
  pip <command> [options]

Commands:
  install                     Install packages.
  download                    Download packages.
  uninstall                   Uninstall packages.
  freeze                      Output installed packages in requirements format.
  list                        List installed packages.
  show                        Show information about installed packages.
  check                       Verify installed packages have compatible dependencies.
  config                      Manage local and global configuration.
  search                      Search PyPI for packages.
  cache                       Inspect and manage pip's wheel cache.
  wheel                       Build wheels from your requirements.
  hash                        Compute hashes of package archives.
  completion                  A helper command used for command completion.
  debug                       Show information useful for debugging.
  help                        Show help for commands.
```

Python environment using virtualenv

```
-bash-4.2$ ml miniconda3/4.12.0-py39
-bash-4.2$ pip install virtualenv
Defaulting to user installation because normal site-packages is not writeable
Collecting virtualenv
  Downloading virtualenv-20.22.0-py3-none-any.whl (3.2 MB)
     |████████████████████████████████| 3.2 MB 5.9 MB/s
Collecting distlib<1,>=0.3.6
  Downloading distlib-0.3.6-py2.py3-none-any.whl (468 kB)
     |████████████████████████████████| 468 kB 43.2 MB/s
Collecting platformdirs<4,>=3.2
  Downloading platformdirs-3.3.0-py3-none-any.whl (15 kB)
Collecting filelock<4,>=3.11
  Downloading filelock-3.12.0-py3-none-any.whl (10 kB)
Installing collected packages: platformdirs, filelock, distlib, virtualenv
  WARNING: The script virtualenv is installed in '/users/PZS0680/soottikkal/.local/bin' which is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
Successfully installed distlib-0.3.6 filelock-3.12.0 platformdirs-3.3.0 virtualenv-20.22.0
-bash-4.2$ virtualenv mynewenv
New python executable in /users/PZS0680/soottikkal/mynewenv/bin/python2
Also creating executable in /users/PZS0680/soottikkal/mynewenv/bin/python
Installing setuptools, pip, wheel...done.
-bash-4.2$ source mynewenv/bin/activate
(mynewenv) -bash-4.2$ pip list
DEPRECATION: The default format will switch to columns in the future. You can use --format=(legacy|columns) (or define a fo
 section) to disable this warning.
pip (9.0.1)
setuptools (28.8.0)
wheel (0.29.0)
You are using pip version 9.0.1, however version 23.1.1 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.
(mynewenv) -bash-4.2$ pip install camelcase
Collecting camelcase
  Cache entry deserialization failed, entry ignored
  Cache entry deserialization failed, entry ignored
  Downloading https://files.pythonhosted.org/packages/24/54/6bc20bf371c1c78193e2e4179097a7b779e56f420d0da41222a3b7d87890/ca
Building wheels for collected packages: camelcase
  Running setup.py bdist_wheel for camelcase ... done
  Stored in directory: /users/PZS0680/soottikkal/.cache/pip/wheels/b1/fe/08/84d2143069bc44c20127c38cc1bf202332319b3da7315ca
Successfully built camelcase
Installing collected packages: camelcase
```

Creating reproducible environment with requirements.txt

The requirements.txt is a simple text file that allows you to keep track of the Python modules installed and enabled in a given environment.

This file keeps a list of modules and packages required in a given project. Hence, if you want to replicate the project in a new environment, you can reference this file to install the dependencies instead of manually tracking them down.

Create requirements.txt

Start by navigating to the environment where your project is located. Then use the pip freeze command to export your packages to the requirements.txt file

```
pip freeze > requirments.txt
```

```
conda list -e > requirements.txt
```

Install required pkgs

```
pip install -r requirements.txt
```

```
conda install --file requirements.txt
```

How to run a Python code

test.py

```
name = input("What is your name? ")
print("Hello, " + name + "!")
```

python test.py

```
(mynewenv) -bash-4.2$ python test.py
What is your name? "shameema"
Hello, shameema!
(mynewenv) -bash-4.2$
```

Running on a compute node:

```
[soottikkaltest@pitzer-login03 ~]$ sinteractive -A PZS1118 -n 2 -t 1:00:00
salloc: Pending job allocation 16568567
salloc: job 16568567 queued and waiting for resources
salloc: job 16568567 has been allocated resources
salloc: Granted job allocation 16568567
salloc: Waiting for resource configuration
salloc: Nodes p0049 are ready for job


Your primary project is the only group quota reported here.
See http://osc.edu/check-quotas to learn how to check the group storage quotas for yo

As of 2023-04-26T10:15:01.000000 userid soottikkaltest on /fs/ess/PZS1124 used 0 GiB
As of 2023-04-26T10:15:01.000000 userid soottikkaltest on /fs/ess/PZS1118 used 0 GiB
As of 2023-04-26T10:15:01.000000 project/group PZS1118 on /fs/ess used 12 GiB of quot
As of 2023-04-26T10:12:13.000000 userid soottikkaltest on /users/PZS1118 used 661.77

[soottikkaltest@p0049 ~]$ ml miniconda3/4.12.0-py39
[soottikkaltest@p0049 ~]$ python test.py
What is your name? "shameema"
Hello, "shameema"!
```

Submitting Batch Jobs

test.py

```
numbers = [1, 2, 3, 4, 5]
average = sum(numbers) / len(numbers)
print("The average of the numbers is:", average)
```

test.sbatch

```
#!/bin/bash
#SBATCH --job-name Python_ExampleJob
#SBATCH --nodes=1 --ntasks-per-node=4
#SBATCH --time=01:00:00
#SBATCH --account PZS1118

ml miniconda3/4.12.0-py39
source activate myenv

cp test.dat test.py $TMPDIR
cd $TMPDIR

python test.py> test.out

cp test.out $SLURM_SUBMIT_DIR
```

```
[soottikkaltest@pitzer-login03 ~]$ sbatch test.sbatch
Submitted batch job 16568740
[soottikkaltest@pitzer-login03 ~]$ qstat -u `whoami`

pitzer-login03.hpc.osc.edu:

                                                                  Req'd  Req'd   Elap
Job id               Username Queue    Name                SessID NDS   TSK   Memory Time Use S Time
-------------------- -------- -------- ------------------- ------ ----- ----- ------ ----- - -----
16568740             soottikk serial-4 Python_ExampleJob    --        1     4     -- 01:00 Q 00:00
[soottikkaltest@pitzer-login03 ~]$ qstat -u `whoami`

pitzer-login03.hpc.osc.edu:

                                                                  Req'd  Req'd   Elap
Job id               Username Queue    Name                SessID NDS   TSK   Memory Time Use S Time
-------------------- -------- -------- ------------------- ------ ----- ----- ------ ----- - -----
16568740             soottikk condo-os Python_ExampleJob    --        1     4     -- 01:00 R 00:00
[soottikkaltest@pitzer-login03 ~]$ qstat -u `whoami`

pitzer-login03.hpc.osc.edu:

                                                                  Req'd  Req'd   Elap
Job id               Username Queue    Name                SessID NDS   TSK   Memory Time Use S Time
-------------------- -------- -------- ------------------- ------ ----- ----- ------ ----- - -----
16568740             soottikk condo-os Python_ExampleJob    --        1     4     -- 01:00 C 00:00
[soottikkaltest@pitzer-login03 ~]$
```

Once the job is completed:

```
[soottikkaltest@pitzer-login03 ~]$ cat test.out
The average of the numbers is: 3.0
[soottikkaltest@pitzer-login03 ~]$
```

OSC OnDemand
ondemand.osc.edu

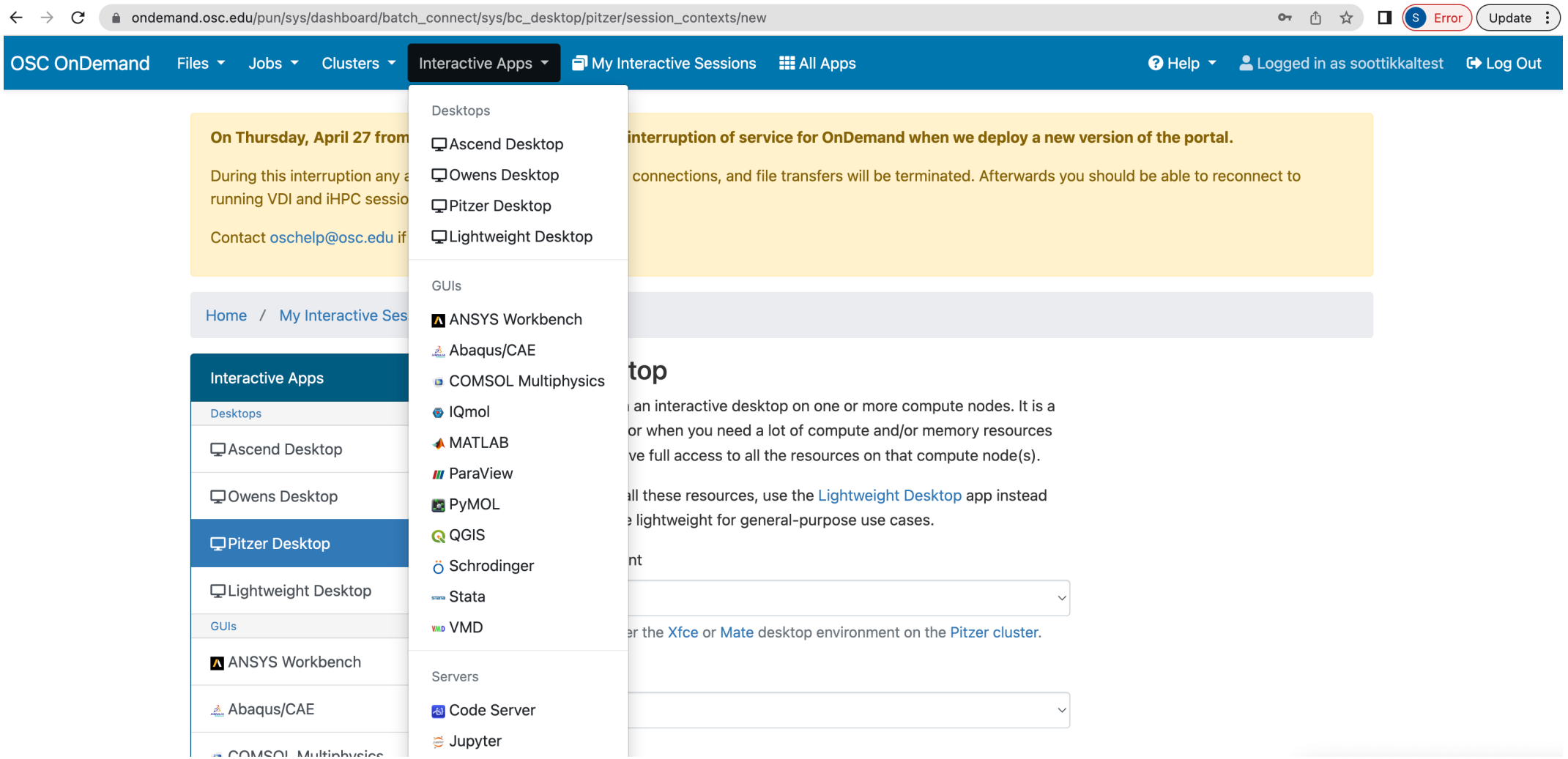
- 1: User Interface
  - Web based
    - Usable from computers, tablets, smartphones
    - Zero installation
  - Single point of entry
    - User needs three things
      - ondemand.osc.edu
      - OSC Username
      - OSC Password
    - Connected to all resources at OSC

- 2: Interactive Services
  - File Access
  - Job Management
  - Visualization Apps
    - Desktop access
    - Single-click apps (Abaqus, Ansys, Comsol, Paraview)
  - Terminal Access

Tutorial available at

osc.edu/ondemand

Interactive Python with Jupyter



Home / My Interactive Sessions / Jupyter

Interactive Apps

Desktops

- Ascend Desktop
- Owens Desktop
- Pitzer Desktop
- Lightweight Desktop

GUIs

- ANSYS Workbench
- Abaqus/CAE
- COMSOL Multiphysics
- IQmol
- MATLAB
- ParaView
- PyMOL
- QGIS

Jupyter version: v0.25.0

This app will launch a Jupyter server using Python on the Ascend, Owens or Pitzer clusters.

Cluster

pitzer

Project

PZS1118

Mode

- (•) Jupyter Lab
- ( ) Jupyter Notebook

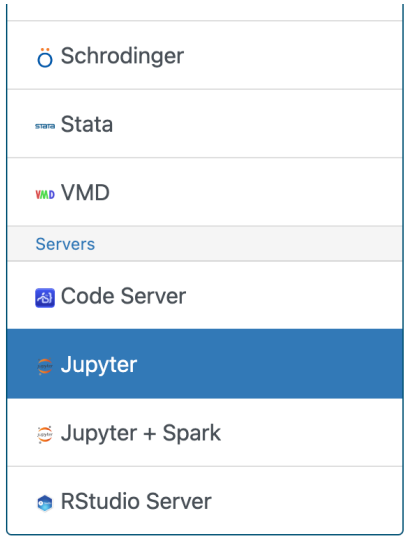
Number of hours

1

Node type

any

- **Standard Compute**
  These are standard HPC machines. Owens has 648 of these nodes with 40 cores and 128 GB of memory. Pitzer has 224 of these nodes with 40 cores and 340 of these nodes with 48 cores. All pitzer nodes have 192 GB of RAM. Chosing "any" as the node type will decrease your wait time.



- **GPU Enabled**
  These are HPC machines with GPUs. Owens has 160 nodes with 1 NVIDIA Tesla P100 GPU and Pitzer has 74 nodes with 2 NVIDIA Tesla V100 GPUs. They have the same CPU and memory characteristics of standard compute. However, Pitzer's 40 core machines have 2 GPUs with 16 GB of RAM; and Pitzer's 48 core machines have 2 GPUs with 32 GB of RAM. Dense GPU types have 4 GPUs with 16 GB of RAM.
- **Large Memory**
  These are HPC machines with very large amounts of memory. Owens has 16 hugemem nodes with 48 cores and 1.5 TB of RAM. Pitzer has 4 hugemem nodes with 3 TB of RAM and 80 cores. Pitzer also has 12 Largmem nodes which have 48 cores with 768 GB of RAM.

Number of cores

1

Number of cores on node type (4 GB per core unless requesting whole node). Leave blank if requesting full node.

JupyterLab Version

3.1

I would like to receive an email when the session starts

Launch

* The Jupyter session data for this session can be accessed under the data root directory.

Session was successfully created.

Home / My Interactive Sessions

Interactive Apps

Desktops

- Ascend Desktop
- Owens Desktop
- Pitzer Desktop
- Lightweight Desktop

GUIs

- ANSYS Workbench

Jupyter (16569985)

1 node | 1 core | Running

Host: >_p0085.ten.osc.edu

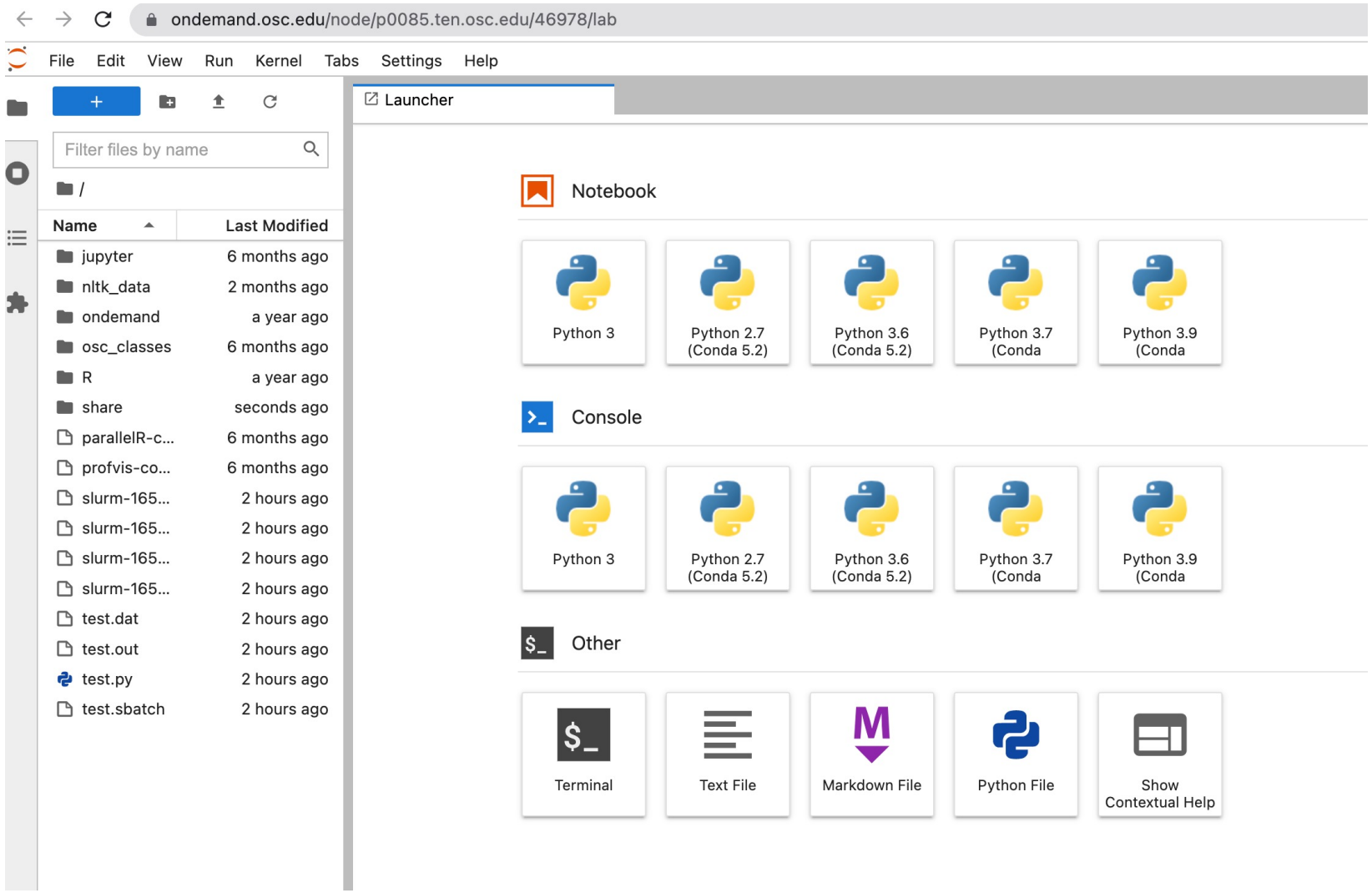
Delete

Created at: 2023-04-26 12:41:39 EDT

Time Remaining: 59 minutes

Session ID: f66f39f7-633c-4b2d-b722-62e31e0e9099

Connect to Jupyter


ondemand.osc.edu/node/p0085.ten.osc.edu/46978/lab
File Edit View Run Kernel Tabs Settings Help
Filter files by name
/
Name
Last Modified
jupyter 6 months ago
nltk_data 2 months ago
ondemand a year ago
osc_classes 6 months ago
R a year ago
share seconds ago
parallelR-c... 6 months ago
profvis-co... 6 months ago
slurm-165... 2 hours ago
slurm-165... 2 hours ago
slurm-165... 2 hours ago
slurm-165... 2 hours ago
test.dat 2 hours ago
test.out 2 hours ago
test.py 2 hours ago
test.sbatch 2 hours ago
Launcher
Notebook
Python 3
Python 2.7 (Conda 5.2)
Python 3.6 (Conda 5.2)
Python 3.7 (Conda
Python 3.9 (Conda
Console
Python 3
Python 2.7 (Conda 5.2)
Python 3.6 (Conda 5.2)
Python 3.7 (Conda
Python 3.9 (Conda
Other
Terminal
Text File
Markdown File
Python File
Show Contextual Help

Kernel name

Untitled.ipynb

Code

Python 3

```
[ ]: name = input("What is your name? ")
     print("Hello, " + name + "!")
```

```
What is your name?  shameema
Hello, shameema!
```

```
[5]: !which python
```

```
/apps/project/ondemand/app_jupyter/3.1.18/bin/python
```

```
[6]: pip list --verbose
```

Package	Version	Location	Installer
anyio	3.6.1	/apps/project/ondemand/app_jupyter/3.1.18/lib/python3.6/site-packages	pip
argon2-cffi	21.3.0	/apps/project/ondemand/app_jupyter/3.1.18/lib/python3.6/site-packages	pip
argon2-cffi-bindings	21.2.0	/apps/project/ondemand/app_jupyter/3.1.18/lib/python3.6/site-packages	pip
async-generator	1.10	/apps/project/ondemand/app_jupyter/3.1.18/lib/python3.6/site-packages	pip
attrs	21.4.0	/apps/project/ondemand/app_jupyter/3.1.18/lib/python3.6/site-packages	pip
Babel	2.10.3	/apps/project/ondemand/app_jupyter/3.1.18/lib/python3.6/site-packages	pip
backcall	0.2.0	/apps/project/ondemand/app_jupyter/3.1.18/lib/python3.6/site-packages	pip
bleach	4.1.0	/apps/project/ondemand/app_jupyter/3.1.18/lib/python3.6/site-packages	pip
certifi	2022.6.15	/apps/project/ondemand/app_jupyter/3.1.18/lib/python3.6/site-packages	pip
cffi	1.15.0	/apps/project/ondemand/app_jupyter/3.1.18/lib/python3.6/site-packages	pip
charset-normalizer	2.0.12	/apps/project/ondemand/app_jupyter/3.1.18/lib/python3.6/site-packages	pip
contextvars	2.4	/apps/project/ondemand/app_jupyter/3.1.18/lib/python3.6/site-packages	pip
dataclasses	0.8	/apps/project/ondemand/app_jupyter/3.1.18/lib/python3.6/site-packages	pip
decorator	5.1.1	/apps/project/ondemand/app_jupyter/3.1.18/lib/python3.6/site-packages	pip
defusedxml	0.7.1	/apps/project/ondemand/app_jupyter/3.1.18/lib/python3.6/site-packages	pip
entrypoints	0.4	/apps/project/ondemand/app_jupyter/3.1.18/lib/python3.6/site-packages	pip
gfal2_util	1.8.0	/usr/lib/python3.6/site-packages	
idna	3.3	/apps/project/ondemand/app_jupyter/3.1.18/lib/python3.6/site-packages	pip
immutables	0.18	/apps/project/ondemand/app_jupyter/3.1.18/lib/python3.6/site-packages	pip
importlib-metadata	4.8.3	/apps/project/ondemand/app_jupyter/3.1.18/lib/python3.6/site-packages	pip
ipykernel	5.5.6	/apps/project/ondemand/app_jupyter/3.1.18/lib/python3.6/site-packages	pip

[1]:
```
name = input("What is your name? ")
print("Hello, " + name + "!")
```

```
What is your name?  shameema
Hello, shameema!
```

[2]:
```
!which python
```

```
/apps/python/3.9-2022.05/bin/python
```

[3]:
```
pip list --verbose
```

```
Package                       Version            Location                                                      Installer
----------------------------- ------------------ ------------------------------------------------------------- ---------
aiohttp                       3.8.1              /apps/python/3.9-2022.05/lib/python3.9/site-packages          conda
aiosignal                     1.2.0              /apps/python/3.9-2022.05/lib/python3.9/site-packages          conda
alabaster                     0.7.12             /apps/python/3.9-2022.05/lib/python3.9/site-packages          conda
anaconda-client               1.9.0              /apps/python/3.9-2022.05/lib/python3.9/site-packages          conda
anaconda-navigator            2.1.4              /apps/python/3.9-2022.05/lib/python3.9/site-packages
anaconda-project              0.10.2             /apps/python/3.9-2022.05/lib/python3.9/site-packages          conda
anyio                         3.5.0              /apps/python/3.9-2022.05/lib/python3.9/site-packages          conda
appdirs                       1.4.4              /apps/python/3.9-2022.05/lib/python3.9/site-packages
argon2-cffi                   21.3.0             /apps/python/3.9-2022.05/lib/python3.9/site-packages          conda
argon2-cffi-bindings          21.2.0             /apps/python/3.9-2022.05/lib/python3.9/site-packages          conda
arrow                         1.2.2              /apps/python/3.9-2022.05/lib/python3.9/site-packages          conda
astroid                       2.6.6              /apps/python/3.9-2022.05/lib/python3.9/site-packages          conda
astropy                       5.0.4              /apps/python/3.9-2022.05/lib/python3.9/site-packages          conda
asttokens                     2.0.5              /apps/python/3.9-2022.05/lib/python3.9/site-packages          conda
async-timeout                 4.0.1              /apps/python/3.9-2022.05/lib/python3.9/site-packages          conda
atomicwrites                  1.4.0              /apps/python/3.9-2022.05/lib/python3.9/site-packages          conda
attrs                         21.4.0             /apps/python/3.9-2022.05/lib/python3.9/site-packages          conda
Automat                       20.2.0             /apps/python/3.9-2022.05/lib/python3.9/site-packages          conda
autopep8                      1.6.0              /apps/python/3.9-2022.05/lib/python3.9/site-packages          conda
Babel                         2.9.1              /apps/python/3.9-2022.05/lib/python3.9/site-packages          conda
backcall                      0.2.0              /apps/python/3.9-2022.05/lib/python3.9/site-packages          conda
backports.functools-lru-cache 1.6.4              /apps/python/3.9-2022.05/lib/python3.9/site-packages          conda
```

Installing packages

[2]:
```
pip install camelcase
```

```
Defaulting to user installation because normal site-packages is not writeable
Collecting camelcase
  Using cached camelcase-0.2-py3-none-any.whl
Installing collected packages: camelcase
Successfully installed camelcase-0.2
Note: you may need to restart the kernel to use updated packages.
```

[1]:
```
import camelcase
print(camelcase.__file__)
```

```
/users/PZS1118/soottikkaltest/.local/lib/python3.6/site-packages/camelcase/__init__.py
```

How to load custom Python environment in Jupyter

if the Conda environment is created via `conda create -n MYENV`

```
~support/classroom/tools/create_jupyter_kernel conda MYENV
```

if the Conda environment is created via `conda create -p /path/to/MYENV`

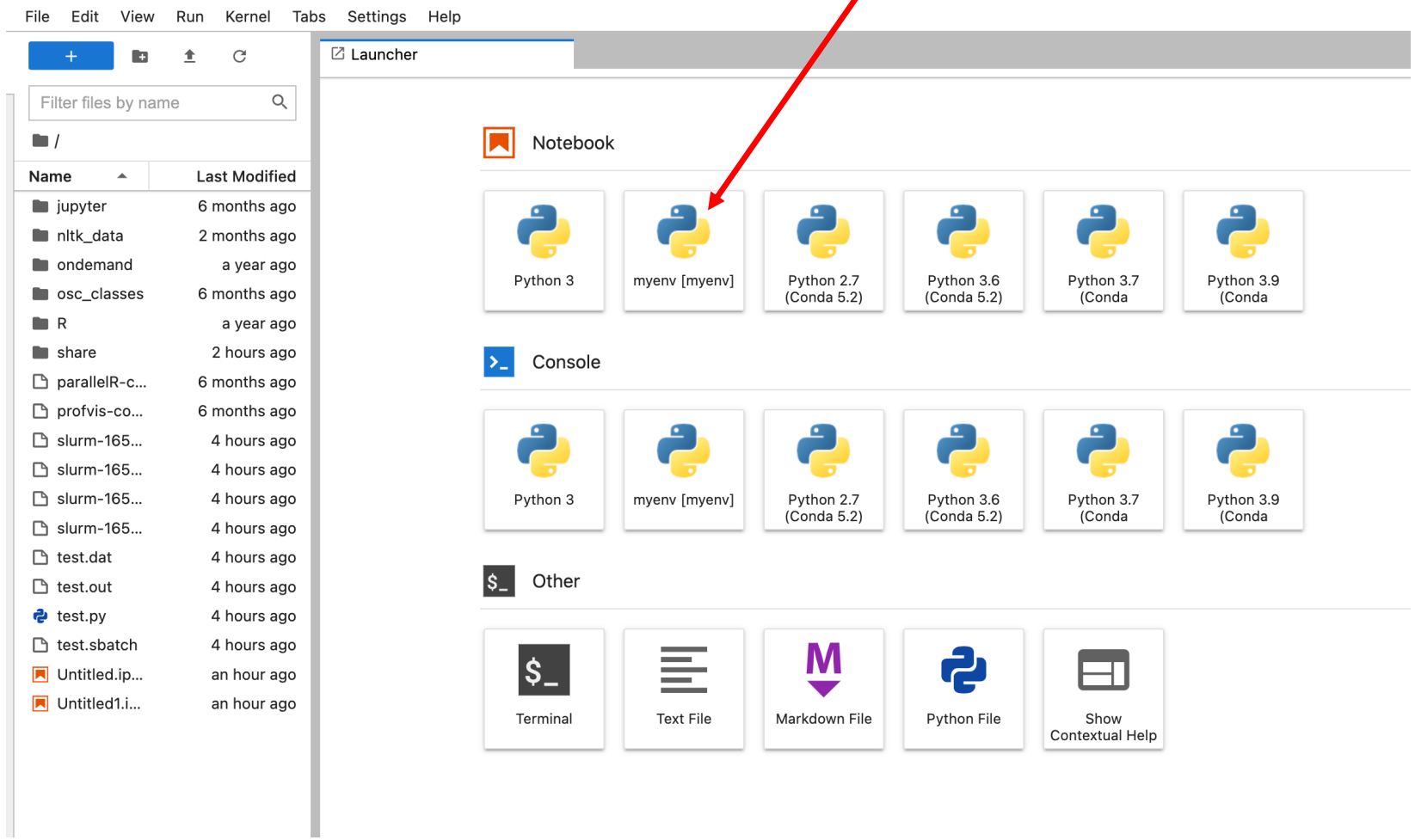
```
~support/classroom/tools/create_jupyter_kernel conda /path/to/MYENV
```

if the Python virtual environment is created via `python3 -m venv /path/to/MYENV`

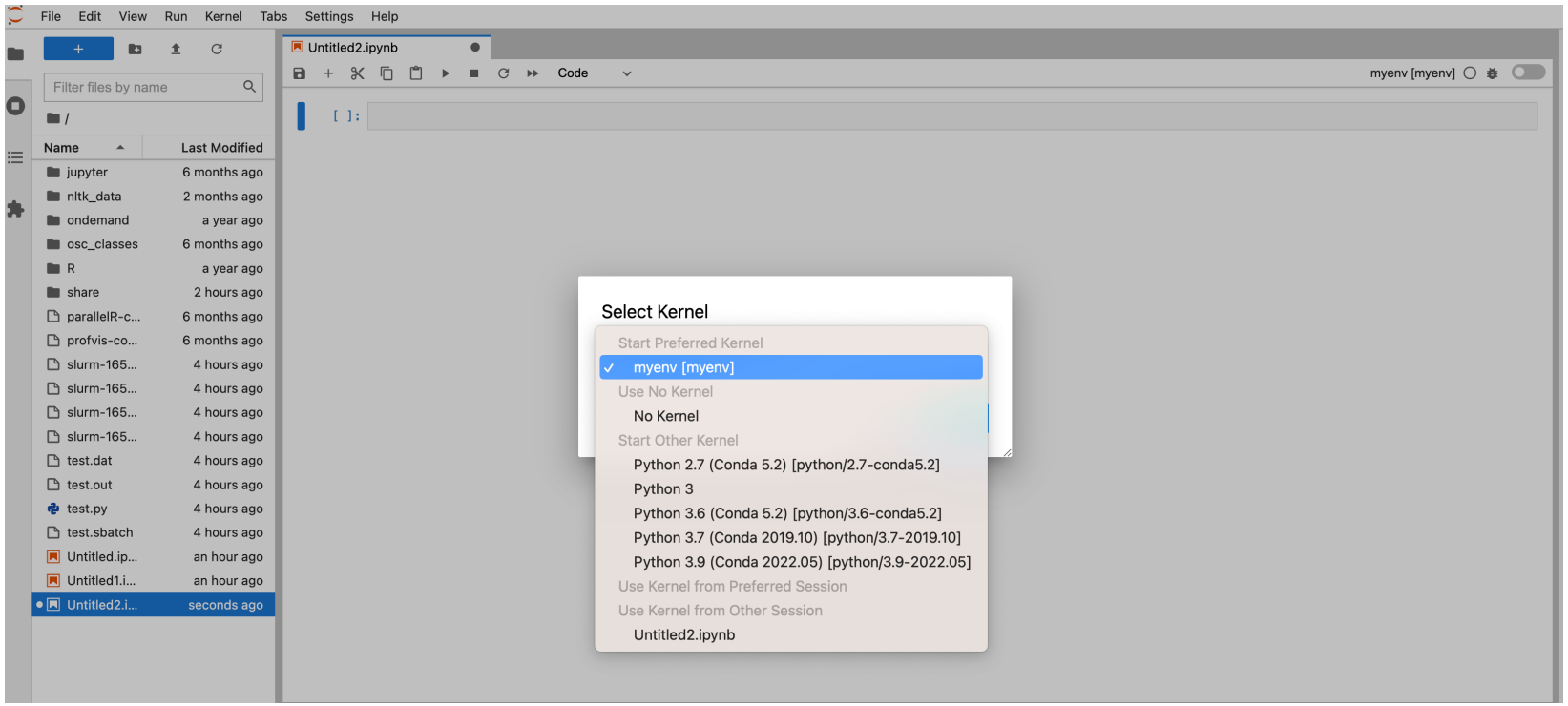
```
~support/classroom/tools/create_jupyter_kernel venv /path/to/MYENV
```

```
[soottikkaltest@pitzer-login03 ~]$ ml miniconda3/4.12.0-py39
[soottikkaltest@pitzer-login03 ~]$ source activate myenv
(myenv) [soottikkaltest@pitzer-login03 ~]$ ~support/classroom/tools/create_jupyter_kernel conda myenv

## Activating Conda environment
source activate myenv

## Checking ipykernel
:: Cannot find ipykernel packages. Attempting to intall ipykernel
:: If it fails, please install it manually

## Installing ipykernel
++ mktemp -d
+ CONDA_PKGS_DIRS=/tmp/tmp.ci8BzYuGiL
+ conda install --yes --quiet --no-update-deps pip
+ local cmd=install
+ case "$cmd" in
+ __conda_exe install --yes --quiet --no-update-deps pip
+ __add_sys_prefix_to_path
+ '[' -n '' ']'
++ dirname /apps/miniconda3/4.12.0-py39/bin/conda
+ SYSP=/apps/miniconda3/4.12.0-py39/bin
++ dirname /apps/miniconda3/4.12.0-py39/bin
+ SYSP=/apps/miniconda3/4.12.0-py39
+ '[' -n '' ']'
+ PATH=/apps/miniconda3/4.12.0-py39/bin:/users/PZS1118/soottikkaltest/.conda/envs/myenv/bin:/apps/minic
/bin:/opt/mvapich2/intel/19.0/2.3.3/bin:/apps/gnu/8.4.0/bin:/opt/intel/19.0.5/itac_latest/bin:/opt/inte
9.0.5/inspector_2019/bin64:/opt/intel/19.0.5/compilers_and_libraries_2019/linux/bin/intel64:/apps/softw
ocal/sbin:/usr/sbin:/opt/ibutils/bin:/opt/puppetlabs/bin
+ export PATH
+ /apps/miniconda3/4.12.0-py39/bin/conda install --yes --quiet --no-update-deps pip
Collecting package metadata (current_repodata.json): ...working... done
Solving environment: ...working... done

## Package Plan ##

  environment location: /users/PZS1118/soottikkaltest/.conda/envs/myenv
0.00s - Debugger warning: It seems that frozen modules are being used, which may
0.00s - make the debugger miss breakpoints. Please pass -Xfrozen_modules=off
0.00s - to python to disable frozen modules.
0.00s - Note: Debugging will proceed. Set PYDEVD_DISABLE_FILE_VALIDATION=1 to disable this validation.
Installed kernelspec conda_myenv in /users/PZS1118/soottikkaltest/.local/share/jupyter/kernels/conda_myenv
+ set +x

## Jupyter kernel is successfully created
```

New custom kernel



Switch kernels

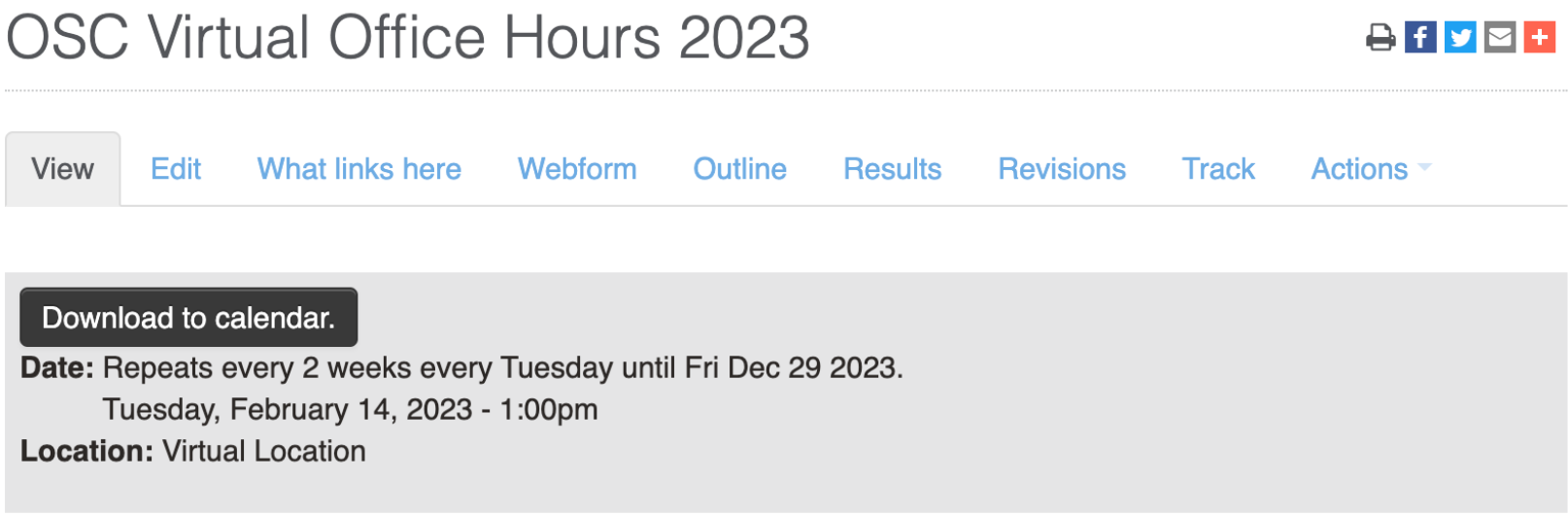


If you need further help:

https://www.osc.edu/resources/available_software/software_list/python

https://www.osc.edu/resources/getting_started/howto/howto_add_python_packages_using_the_conda_package_manager

https://www.osc.edu/resources/getting_started/howto/howto_use_a_condavirtual_environment_with_jupyter



Email: oschelp@osc.edu