Magic Tools to Install & Manage Software

Part 1: CONDA Virtual Environment

Jason Li
HPC User Services
LSU HPC / LONI
sys-help@loni.org

Louisiana State University, Baton Rouge
Apr 1, 2026

Magic Tools to Install & Manage Software

Part 1: CONDA Virtual Environment

Part 2: Singularity Container

Outlines

1. **Why Conda?**
   1) Problems
   2) Virtual environment & Conda
2. **Basic Usage**
   1) Get Conda
   2) Typical workflow
   3) Creating a virtual environment
   4) Installing software packages
3. **Advanced Tips**
   1) Where to get software?
   2) Change Conda path
   3) Share virtual environment
   4) Migrate / clone virtual environment
   5) Use virtual environment in Open OnDemand
   6) More than Python
   7) Troubleshooting
   8) Story time…

Outlines

1. **Why Conda?**
   1) Problems
   2) Virtual environment & Conda
2. **Basic Usage**
   1) Get Conda
   2) Typical workflow
   3) Creating a virtual environment
   4) Installing software packages
3. **Advanced Tips**
   1) Where to get software?
   2) Change Conda path
   3) Share virtual environment
   4) Migrate / clone virtual environment
   5) Use virtual environment in Open OnDemand
   6) More than Python
   7) Troubleshooting
   8) Story time…

1) Problems

- **Core problem:**

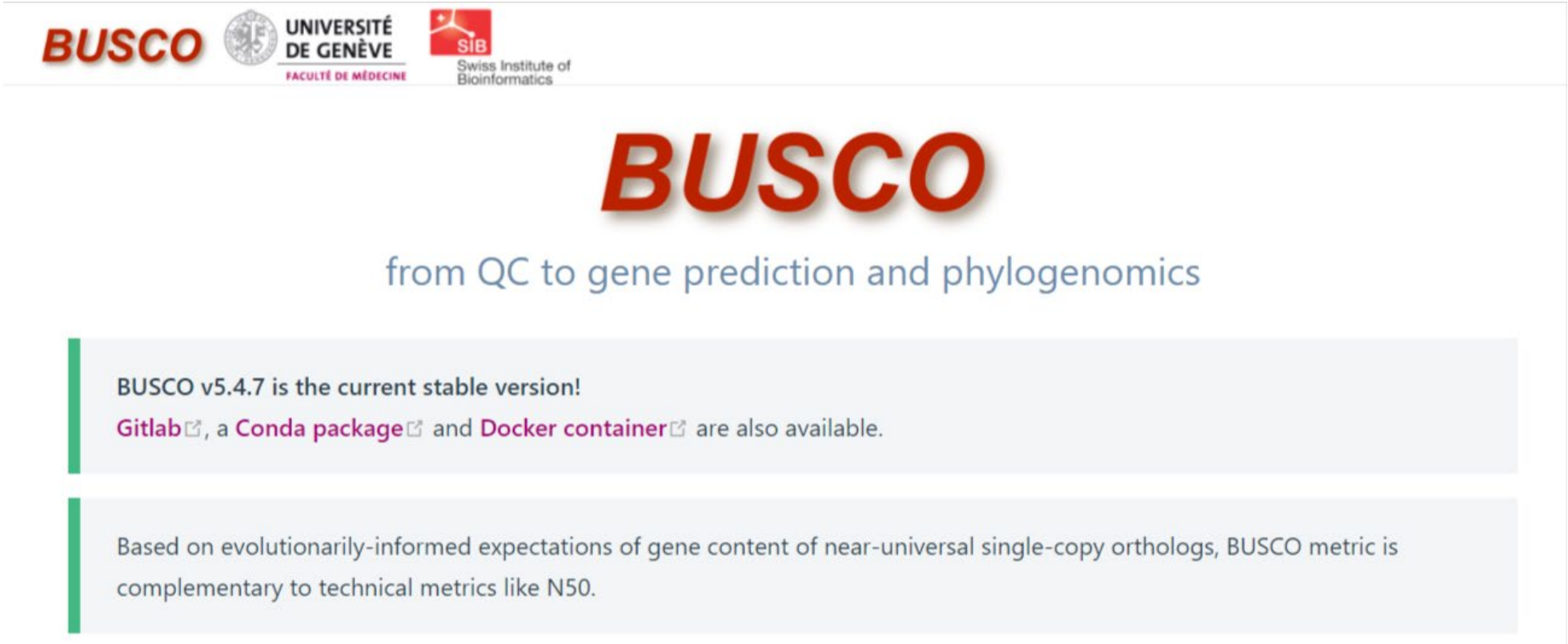
Installing software on an HPC system

1) Problems

- **Traditional Linux solution:**
  - Compiling from source code

1) Problems

a) **Dependencies** (Welcome to Linux!)



1) Problems

a) **Dependencies** (Welcome to Linux!)

Third-party components

A full installation of BUSCO requires *Python 3.3+* (2.7 is not supported from v4 onwards), *BioPython*, *pandas*, *BBMap*, *tBLASTn 2.2+*, *Augustus 3.2+*, *Prodigal*, *Metaeuk*, *HMMER3.1+*, *SEPP*, and *R + ggplot2* for the plotting companion script. Some of these tools are necessary only for analysing certain type of organisms and input data, or for specific run modes.

- https://biopython.org/
- https://pandas.pydata.org/
- https://jgi.doe.gov/data-and-tools/software-tools/bbtools/
- https://ftp.ncbi.nlm.nih.gov/blast/executables/blast+/LATEST
- http://bioinf.uni-greifswald.de/augustus/
- https://github.com/soedinglab/metaeuk
- https://github.com/hyattpd/Prodigal
- http://hmmer.org/
- https://github.com/smirarab/sepp/
- https://www.r-project.org/

> Please make sure that each software package listed above works INDEPENDENTLY of BUSCO before attempting to run any BUSCO assessments.

1) Problems

a) **Dependencies** (Welcome to Linux!)

Third-party components

A full installation of BUSCO requires *Python 3.3+* (2.7 is not supported from v4 onwards), *BioPython*, *pandas*, *BBMap*, *tBLASTn 2.2+*, *Augustus 3.2+*, *Prodigal*, *Metaeuk*, *HMMER3.1+*, *SEPP*, and *R + ggplot2* for the plotting companion script. Some of these tools are necessary only for analysing certain type of organisms and input data, or for specific run modes.

- https://biopython.org/
- https://pandas.pydata.org/
- https://jgi.doe.gov/data-and-tools/software-tools/bbtools/
- https://ftp.ncbi.nlm.nih.gov/blast/executables/blast+/LATEST
- http://bioinf.uni-greifswald.de/augustus/
- https://github.com/soedinglab/metaeuk
- https://github.com/hyattpd/Prodigal
- http://hmmer.org/
- https://github.com/smirarab/sepp/
- https://www.r-project.org/

> Please make sure that each software package listed above works INDEPENDENTLY of BUSCO before attempting to run any BUSCO assessments.

1) Problems

a) **Dependencies** (Welcome to Linux!)

Third-party components

A full installation of BUSCO requires *Python 3.3+* (2.7 is not supported from v4 onwards), *BioPytho*
pandas, *BBMap*, *tBLASTn 2.2+*, *Augustus 3.2+*, *Prodigal*, *Metaeuk*, *HMMER3.1+*, *SEPP*, and *R + ggplo*
the plotting companion script. Some of these tools are necessary only for analysing certain type o
organisms and input data, or for specific run modes.

- https://biopython.org/
- https://pandas.pydata.org/
- https://jgi.doe.gov/data-and-tools/software-tools/bbtools/
- https://ftp.ncbi.nlm.nih.gov/blast/executables/blast+/LATEST
- http://bioinf.uni-greifswald.de/augustus/
- https://github.com/soedinglab/metaeuk
- https://github.com/hyattpd/Prodigal
- http://hmmer.org/
- https://github.com/smirarab/sepp/
- https://www.r-project.org/

> Please make sure that each software package listed above works INDEPENDENTLY of BUSCO
> before attempting to run any BUSCO assessments.

- Dependencies

The following dependencies are required for AUGUSTUS:

- for gzip compressed input: (set ZIPINPUT = false in common.mk if t… re not available)
  - libboost-iostreams-dev
  - zlib1g-dev
- for comparative AUGUSTUS (multi-species, CGP): (set COMPGENEPRED = false in co… raries required by the CGP version are not available. Augustus can then only be run in single-genome mode, which is what most users need.)
  - libgsl-dev
  - libboost-all-dev
  - libsuitesparse-dev
  - liblpsolve55-dev
  - libsqlite3-dev (add SQLITE = false to common.mk if this feature is not required or the required library is not available)
  - libmysql++-dev (add MYSQL = false to common.mk if this feature is not required or the required library is not available)
- for compiling utilities bam2hints and filterBam:
  - libbamtools-dev zlib1g-dev
- for compiling utility utrrnaseq:
  - libboost-all-dev (version must be >Boost_1_49_0)
- for compiling utility bam2wig:
  - Follow these instructions. Note that it shouldn't be a problem to compile AUGUSTUS without bam2wig. In practice, you can simply use `bamToWig.py` to accomplish the same task.
- For compiling homgenemapping (set BOOST = FALSE in auxprogs/homgenemapping/src/Makefile if the option --printHomologs is not required or the required libraries are not available)
  - libboost-all-dev
- for scripts:
  - Perl
  - Python3
- for the python3 script bamToWig.py:
  - twoBitInfo and faToTwoBit from http://hgdownload.soe.ucsc.edu/admin/exe . bamToWig.py will automatically download these tools to the working directory during execution if they are not in your $PATH.
  - SAMtools (available e.g. via package managers or here - see notes below)

1) Problems

b) Permission denied (Welcome to HPC!)

```
[jasonli3@mike4 ~]$ module load python
[jasonli3@mike4 ~]$ pip install gdal
```

1) Problems

b) Permission denied (Welcome to HPC!)

```
Using numpy 2.0.2
running egg_info
writing gdal-utils/GDAL.egg-info/PKG-INFO
writing dependency_links to gdal-utils/GDAL.egg-info/dependency_links.txt
writing entry points to gdal-utils/GDAL.egg-info/entry_points.txt
writing requirements to gdal-utils/GDAL.egg-info/requires.txt
writing top-level names to gdal-utils/GDAL.egg-info/top_level.txt
Traceback (most recent call last):
  File "<string>", line 91, in fetch_config
  File "/usr/local/packages/python/3.9.7-anaconda/lib/python3.9/subprocess.p
    self._execute_child(args, executable, preexec_fn, close_fds,
  File "/usr/local/packages/python/3.9.7-anaconda/lib/python3.9/subprocess.p
    raise child_exception_type(errno_num, err_msg, err_filename)
FileNotFoundError: [Errno 2] No such file or directory: 'gdal-config'
```

1) Problems

b) Permission denied (Welcome to HPC!)

```
Using numpy 2.0.2
running egg_info
writing gdal-utils/GDAL.egg-info/PKG-INFO
writing dependency_links to gdal-utils/GDAL.egg-info/dependency_links.txt
writing entry points to gdal-utils/GDAL.egg-info/entry_points.txt
writing requirements to gdal-utils/GDAL.egg-info/requires.txt
writing top-level names to gdal-utils/GDAL.egg-info/top_level.txt
Traceback (most recent call last):
  File "<string>", line 91, in fetch_config
  File "/usr/local/packages/python/3.9.7-anaconda/lib/python3.9/subprocess.p
    self._execute_child(args, executable, preexec_fn, close_fds,
  File "/usr/local/packages/python/3.9.7-anaconda/lib/python3.9/subprocess
    raise child_exception_type(errno_num, err_msg, err_filename)
FileNotFoundError: [Errno 2] No such file or directory: 'gdal-c
```

libgdal

1) Problems

b) Permission denied (Welcome to HPC!)

- If you ask Google / ChatGPT…

```
$ sudo yum install libgdal-devel        # On Red Hat

$ sudo apt-get install libgdal-dev      # On Ubuntu
```

1) Problems

b) Permission denied (Welcome to HPC!)

- If you ask Google / AI…

```
$ sudo yum install libgdal-devel        # On Red Hat

$ sudo apt-get install libgdal-dev      # On Ubuntu
```

1) Problems

b) Permission denied (Welcome to HPC!)

- If you ask Google / AI…

```
$ sudo yum install libgdal-devel        # On Red Hat

$ sudo apt-get install libgdal-dev      # On Ubuntu
```

1) Problems

b) Permission denied (Welcome to HPC!)

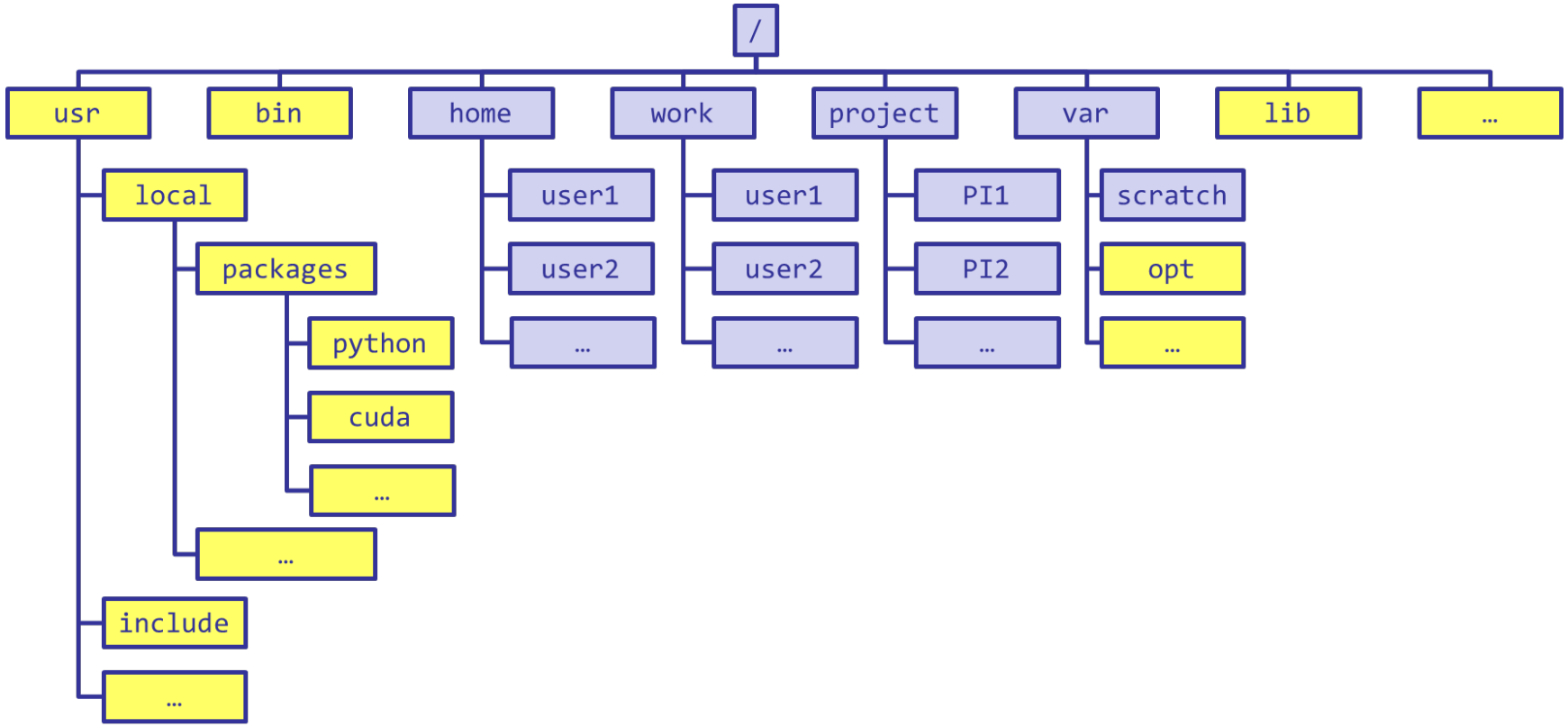
- If you ask Google / AI…

```
$ sudo yum install libgdal-devel        # On Red Hat

$ sudo apt-get install libgdal-dev      # On Ubuntu
```

1) Problems

b) Permission denied (Welcome to HPC!)



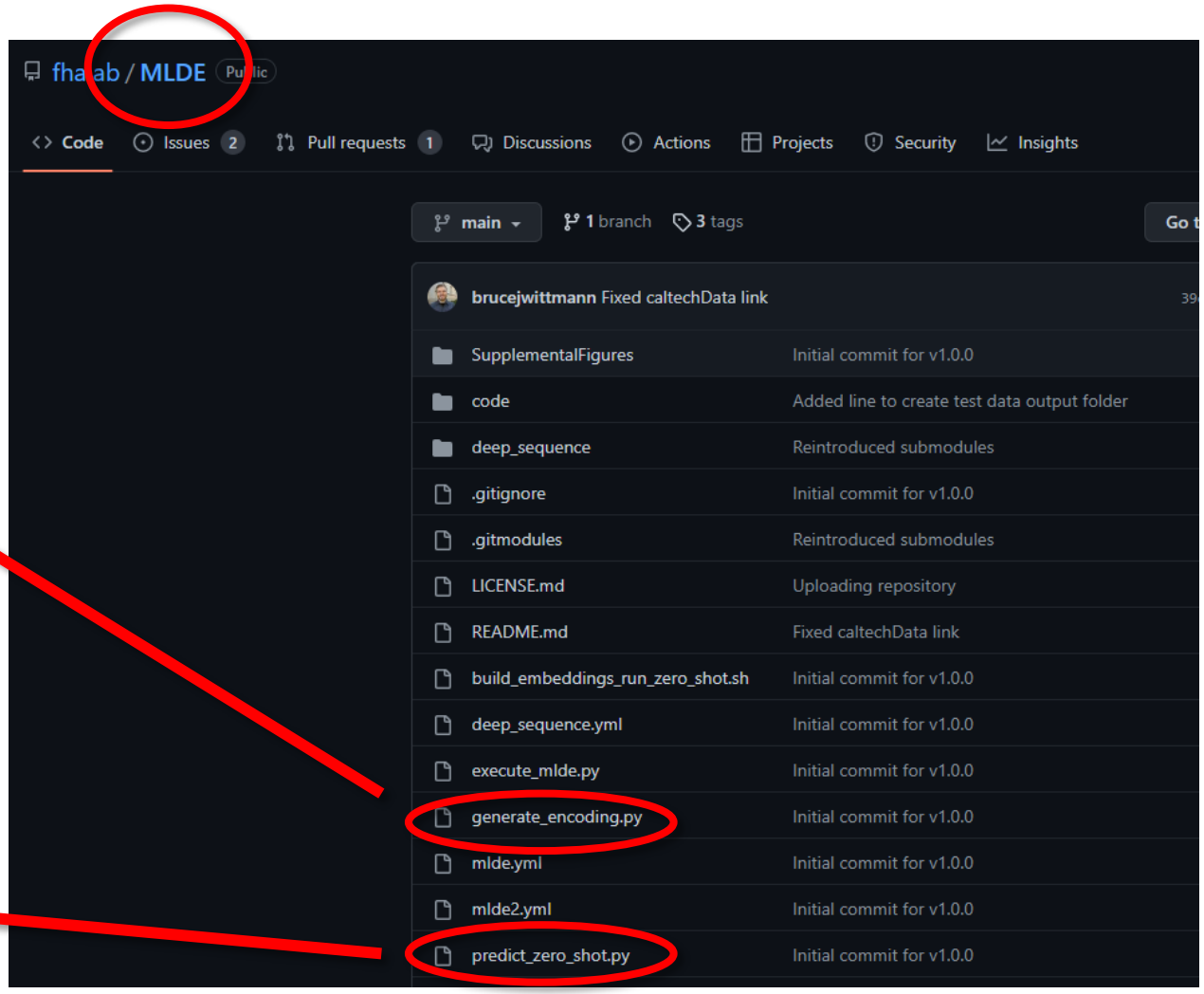
1) Problems

c) Conflicted packages

- What if I need two packages w/ conflicted dependencies?

Tensorflow 1.13

PyTorch > 1.5



d) Sharing / Migrating your software

- Huge effort & large disk quota to install

  - What if my colleagues want to use?
  - What if I want to migrate a different cluster?

1) Problems

Any of those apply to you?

Magic Tools to Install / Manage Software

Part 1: CONDA Virtual Environment

Part 2: Singularity Container

Outlines

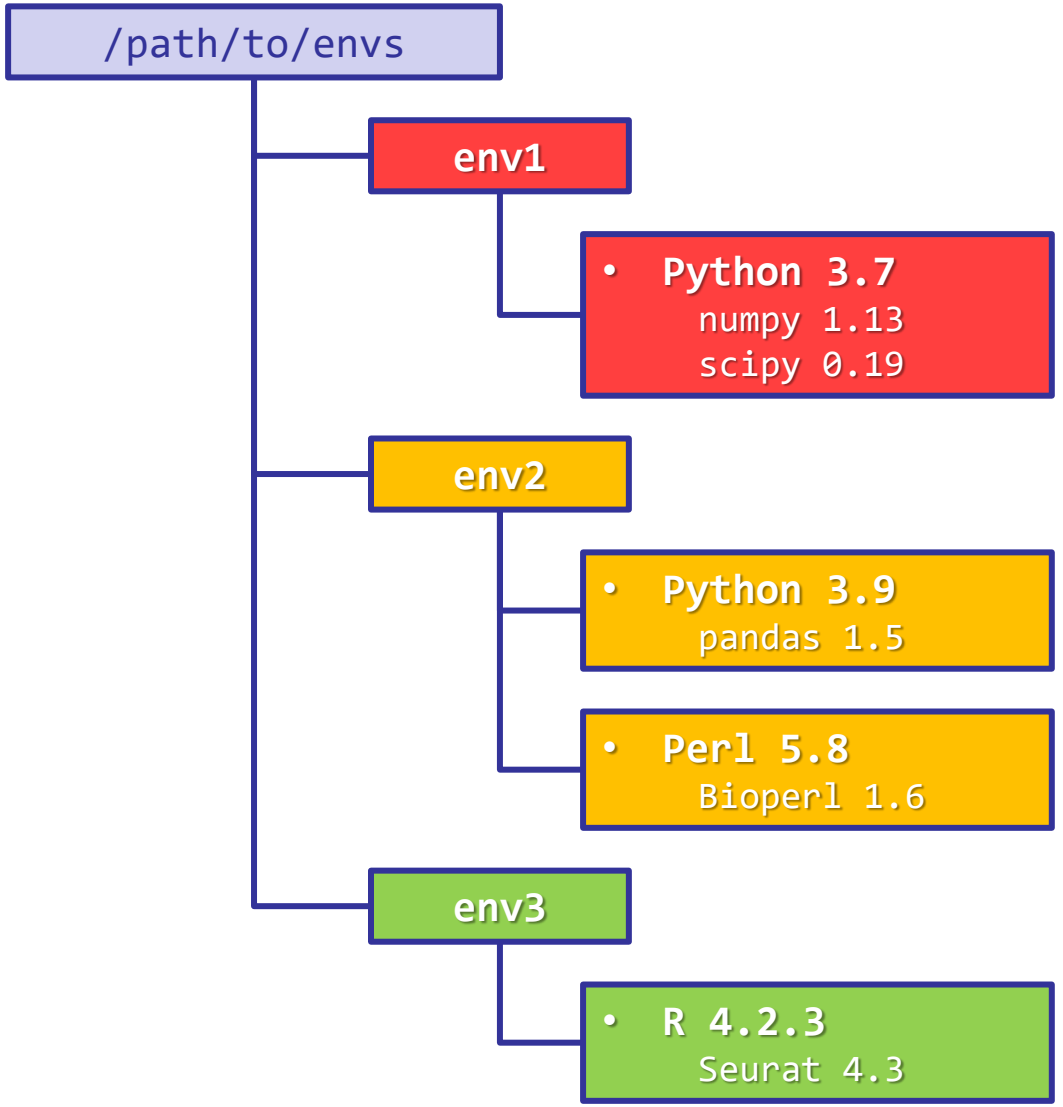
1. **Why Conda?**
   1) Problems
   2) Virtual environment & Conda
2. **Basic Usage**
   1) Get Conda
   2) Typical workflow
   3) Creating a virtual environment
   4) Installing software packages
3. **Advanced Tips**
   1) Where to get software?
   2) Change Conda path
   3) Share virtual environment
   4) Migrate / clone virtual environment
   5) Use virtual environment in Open OnDemand
   6) More than Python
   7) Troubleshooting
   8) Story time…

Virtual Environment

2) Virtual environment & Conda

a) What is a virtual environment?

- **A comprehensive software framework, (usually) consists of:**
  - A **single directory** contains all files (e.g., executables, dependencies, …)
  - Proper **configurations** (e.g., environment variables)



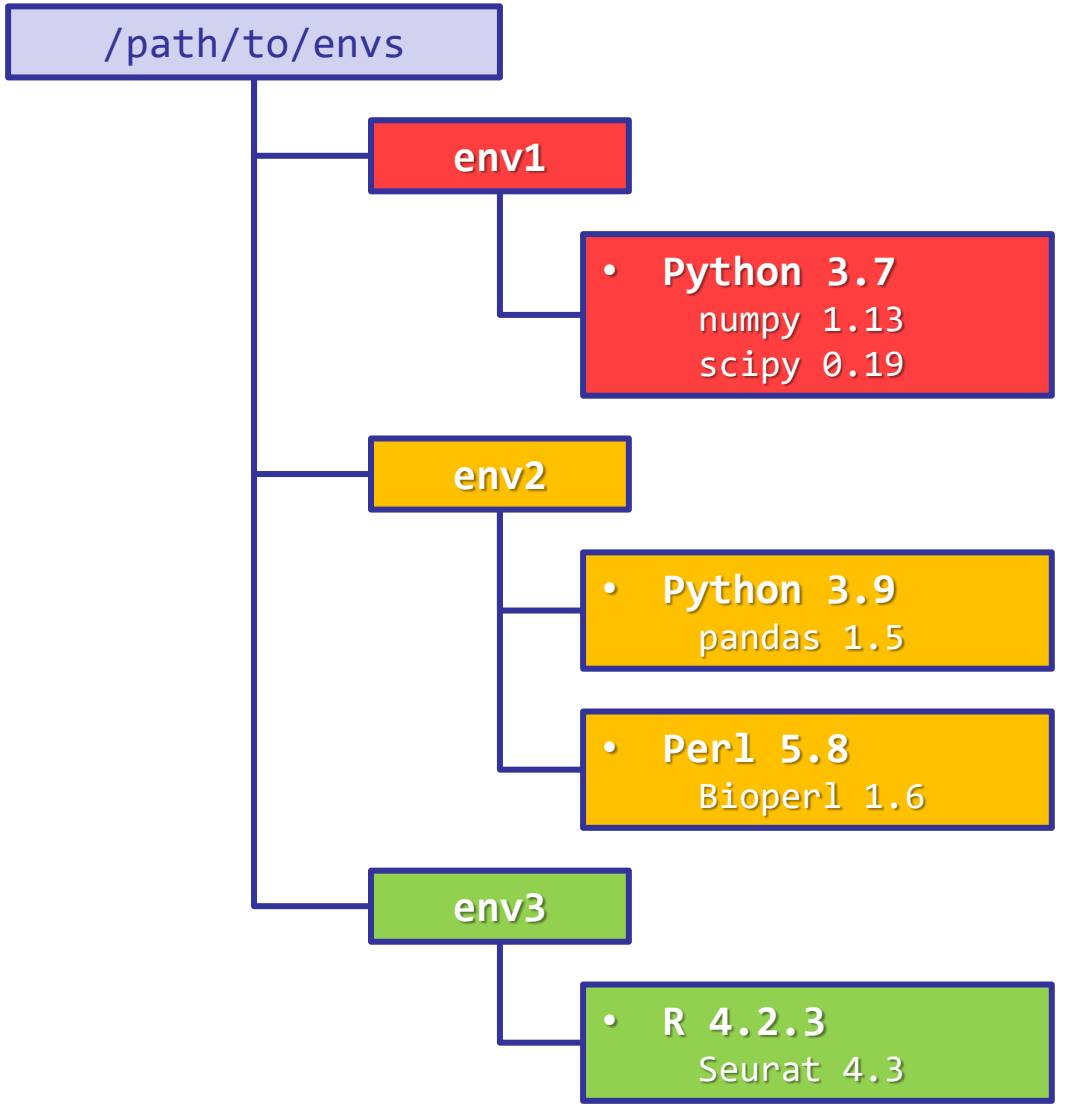
2) Virtual environment & Conda

a) What is a virtual environment?

- **Properties**
  - **Self-contained**

    All dependencies are installed within the VE
  - **Isolated**

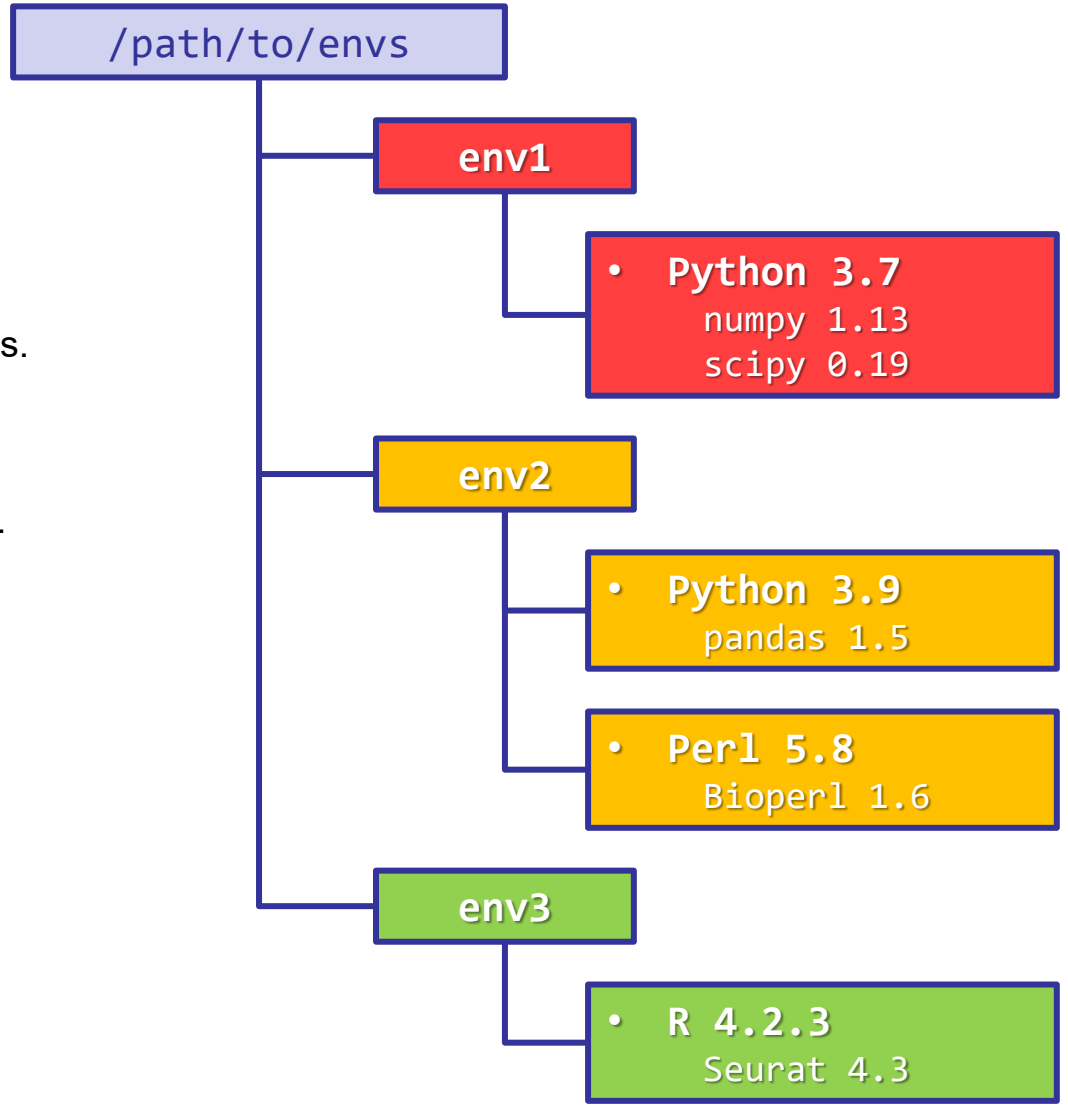
    Whatever happens in a VE stays in that VE…



2) Virtual environment & Conda

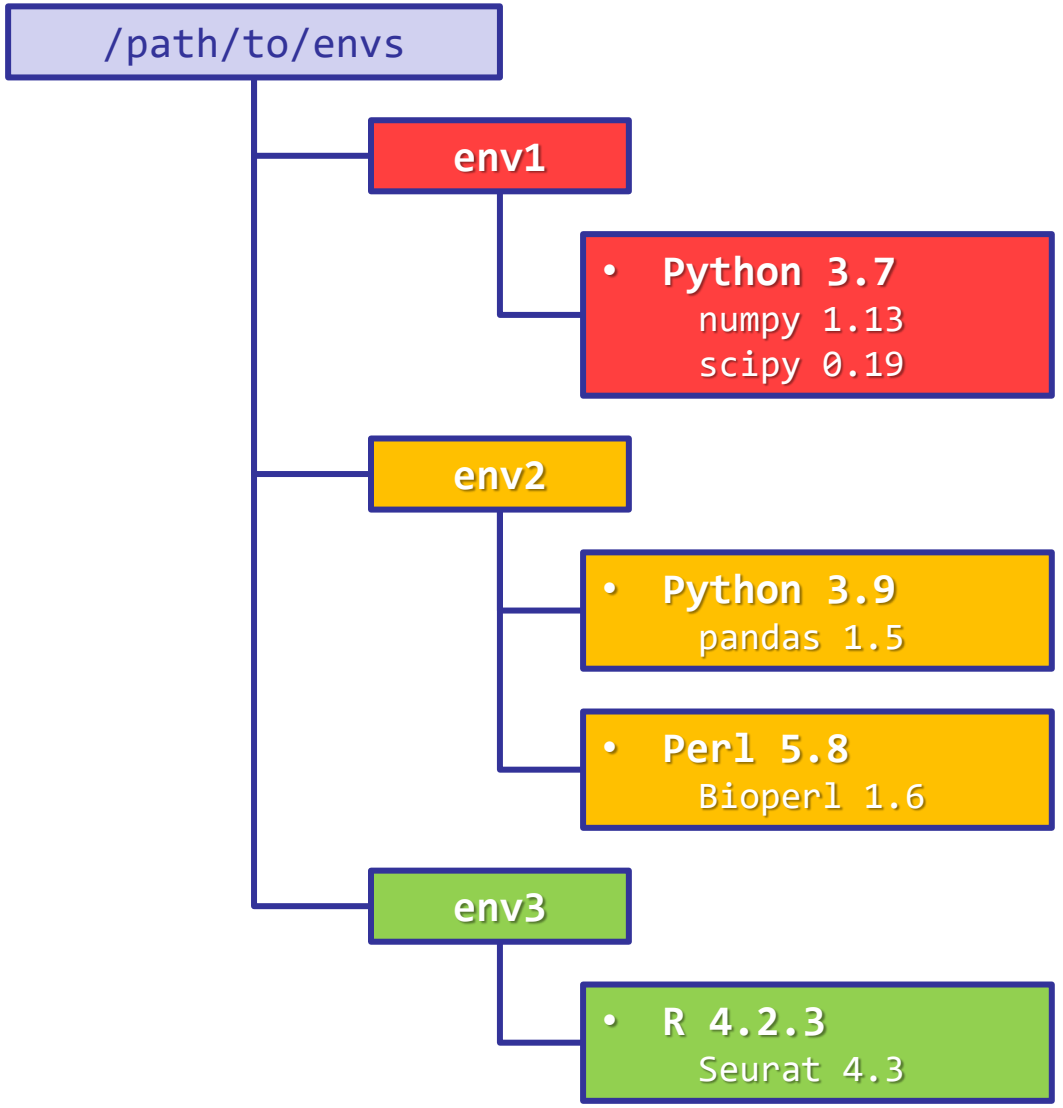
b) How does it solve my problems?

- **Dependency issue**
  - Pack all dependencies in the VE. No need to rely on the system.
  - Tools like **Conda** can help install dependencies.
- **Permission issue**
  - Create a VE where you have write permission.
  - Does not need **sudo** permission
- **Conflicted packages**
  - Install in different VEs.
- **Share / Migrate**
  - Create VE in /project and share w/ group
  - Export recipe and build on a different system

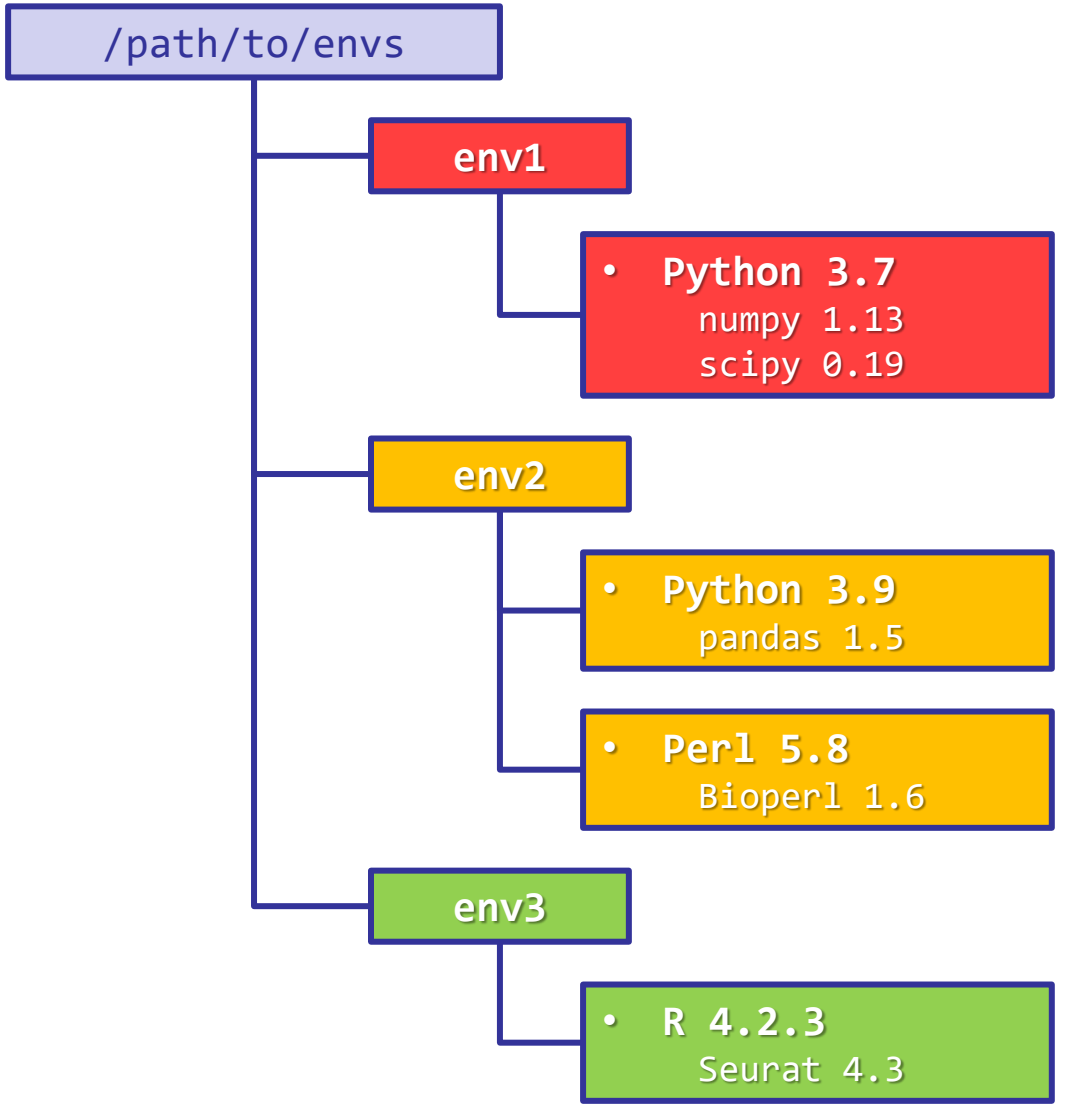


c) What is Conda?

Technology →

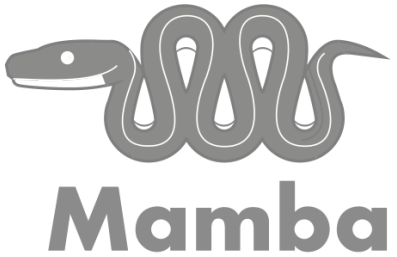


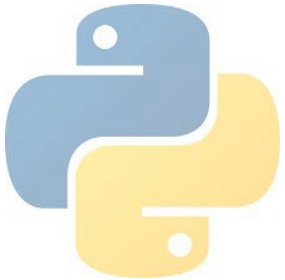
c) What is Conda?



↑ **Software** system that implements the technology

c) What is Conda?





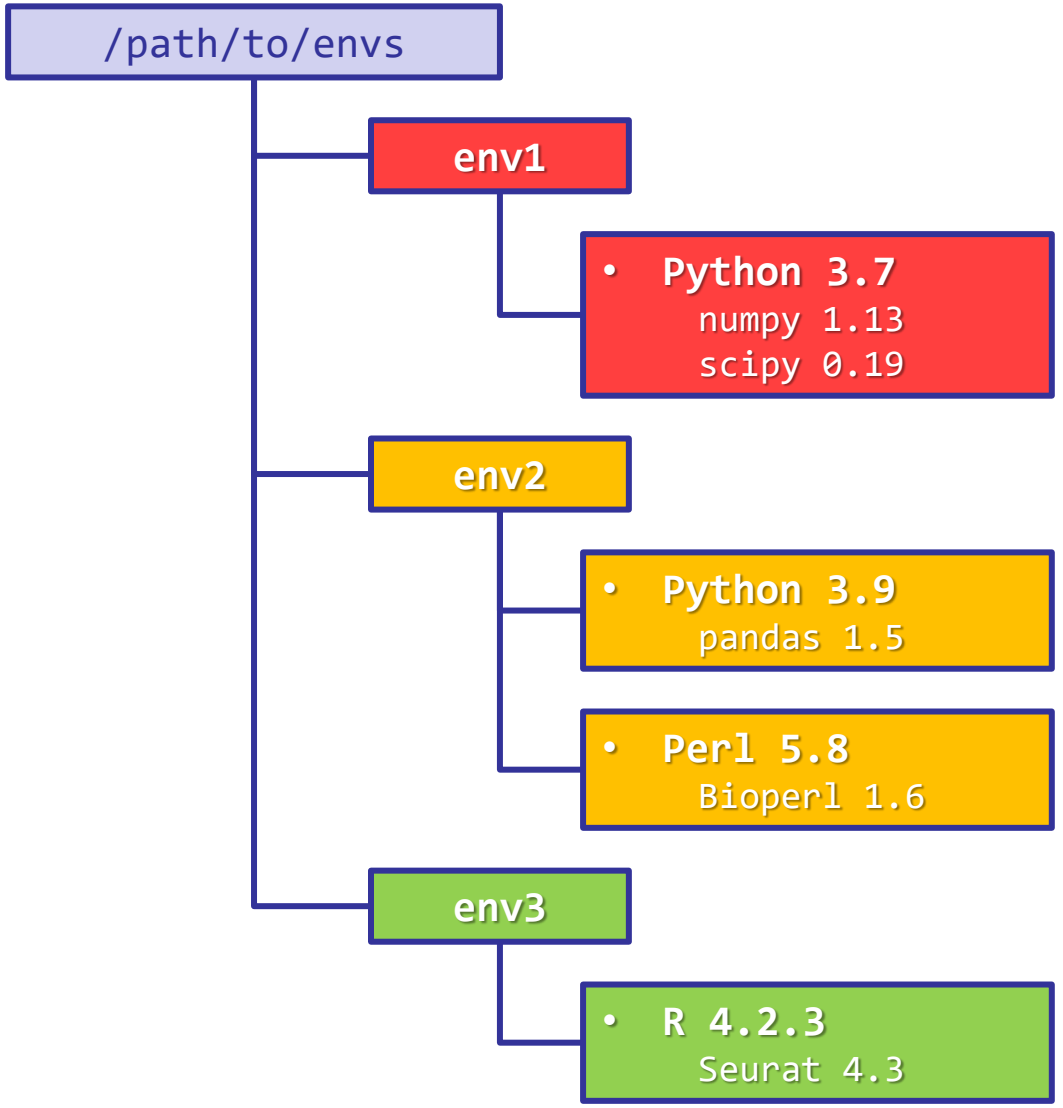
c) What is Conda?

Summary

Technology that helps
with software installation →

↓ **Software** system that
implements the technology

CONDA



Outlines

1. Why Conda?
   1) Problems
   2) Virtual environment & Conda

2. **Basic Usage**
   1) Get Conda
   2) Typical workflow
   3) Creating a virtual environment
   4) Installing software packages

3. Advanced Tips
   1) Where to get software?
   2) Change Conda path
   3) Share virtual environment
   4) Migrate / clone virtual environment
   5) Use virtual environment in Open OnDemand
   6) More than Python
   7) Troubleshooting
   8) Story time…

Outlines

1. Why Conda?
   1) Problems
   2) Virtual environment & Conda
2. **Basic Usage**
   1) Get Conda
   2) Typical workflow
   3) Creating a virtual environment
   4) Installing software packages
3. Advanced Tips
   1) Where to get software?
   2) Change Conda path
   3) Share virtual environment
   4) Migrate / clone virtual environment
   5) Use virtual environment in Open OnDemand
   6) More than Python
   7) Troubleshooting
   8) Story time…

1) Get Conda

a) Use Conda module [Recommended]

- No installation / disk quota required.
- Sufficient for most user cases.

```
$ module load conda
```

Step 1: Can use Conda

```
$ conda init
```

Step 2: Can use Conda later without loading conda module, and avoid Conda initiation error (recommended)

b) Install your own Conda (only if needed)

– Conda distributions

Popular Distribution	Description
Anaconda	Full size Conda + Python, w/ a LOT of Python packages. Supported by Anaconda Inc.
Miniconda	Minimum size Conda + Python only. Supported by Anaconda Inc.
Miniforge	Minimum size Conda + Python only. Community supported.

1) Get Conda

b) Install your own Conda (only if needed)

– Conda distributions

Popular Distribution	Description
Anaconda	Full size Conda + Python, w/ a LOT of Python packages. Support by Anaconda Inc.
Miniconda	Minimum size Conda + Python only. Support by Anaconda Inc.
Miniforge	Minimum size Conda + Python only. Community supported.

1) Get Conda

b) Install your own Conda (only if needed)

– Latest Miniforge: https://github.com/conda-forge/miniforge/releases/latest/download/Miniforge3-Linux-x86_64.sh

```
$ wget https://github.com/conda-forge/miniforge/releases/latest/download/Miniforge3-
  Linux-x86_64.sh                                    Step 1: Download Miniforge

$ chmod u+x Miniforge3-Linux-x86_64.sh               Step 2: Enable execution

$ ./Miniforge3-Linux-x86_64.sh                       Step 3: Run and follow prompts
```

Outlines

1. Why Conda?
   1) Problems
   2) Virtual environment & Conda
2. **Basic Usage**
   1) Get Conda
   2) Typical workflow
   3) Creating a virtual environment
   4) Installing software packages
3. Advanced Tips
   1) Where to get software?
   2) Change Conda path
   3) Share virtual environment
   4) Migrate / clone virtual environment
   5) Use virtual environment in Open OnDemand
   6) More than Python
   7) Troubleshooting
   8) Story time…

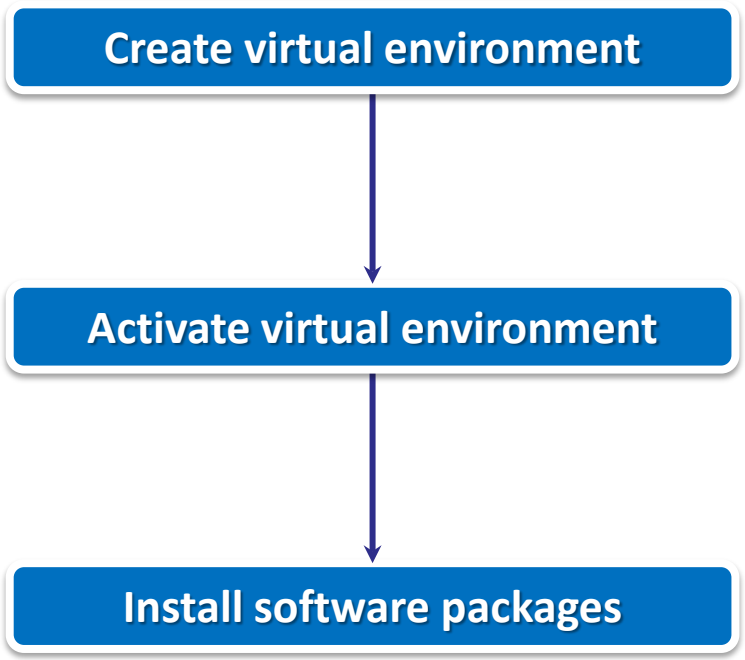
2) Typical workflow

- **Rule of thumb:**

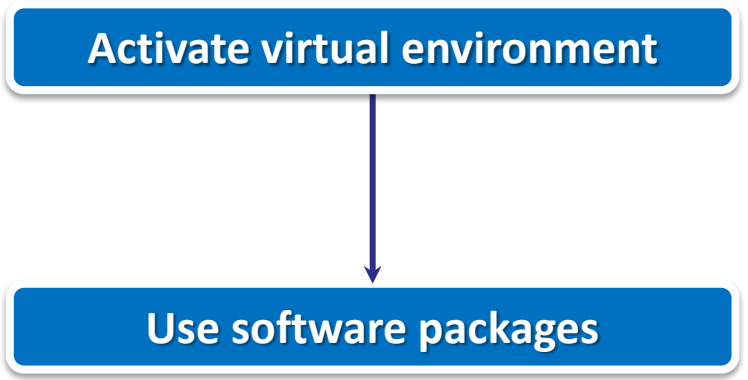
Always use a virtual environment with Conda!

2) Typical workflow

To install …



To use …



Outlines

1. Why Conda?
   1) Problems
   2) Virtual environment & Conda
2. **Basic Usage**
   1) Get Conda
   2) Typical workflow
   3) Creating a virtual environment
   4) Installing software packages
3. Advanced Tips
   1) Where to get software?
   2) Change Conda path
   3) Share virtual environment
   4) Migrate / clone virtual environment
   5) Use virtual environment in Open OnDemand
   6) More than Python
   7) Troubleshooting
   8) Story time…

3) Creating a virtual environment

- **Most frequently used commands**

Command	
`conda create -n ENVIRONMENT`	Create

```
(base) [jasonli3@mike4 ~]$ conda create -n myenv
Retrieving notices: ...working... done
Channels:
 - defaults
Platform: linux-64
Collecting package metadata (repodata.json): done
Solving environment: done

## Package Plan ##

  environment location: /work/jasonli3/.conda/envs/myenv



Proceed ([y]/n)?

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

(base) [jasonli3@mike4 ~]$
```

[1] https://docs.conda.io/projects/conda/en/latest/commands.html

3) Creating a virtual environment

- **Most frequently used commands**

Command	Description
`conda create -n ENVIRONMENT`	Create a virtual environment
`source activate ENVIRONMENT`	Activate a virtual environment

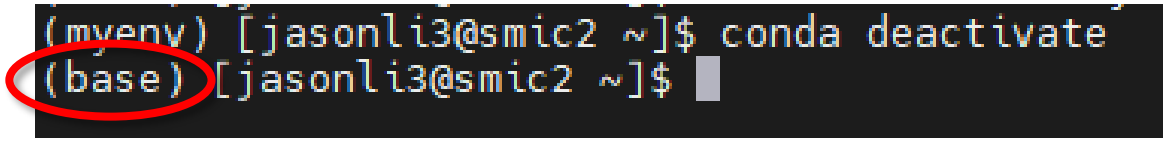
```
(base) [jasonli3@smic2 ~]$ source activate myenv
(myenv) [jasonli3@smic2 ~]$
```

[1] https://docs.conda.io/projects/conda/en/latest/commands.html

3) Creating a virtual environment

- **Most frequently used commands**

Command	Description
`conda create -n ENVIRONMENT`	Create a virtual environment
`source activate ENVIRONMENT`	Activate a virtual environment
`conda deactivate`	Deactivate a virtual environment



[1] https://docs.conda.io/projects/conda/en/latest/commands.html

3) Creating a virtual environment

- **Most frequently used commands**

Command	Description
`conda create -n ENVIRONMENT`	Create a virtual environment
`source activate ENVIRONMENT`	Activate a virtual environment
`conda deactivate`	Deactivate a virtual environment
`conda env list`	List all virtual environments

```
(base) [jasonli3@smic2 ~]$ conda env list
# conda environments:
#
myenv                    /home/jasonli3/.conda/envs/myenv
base                  *  /usr/local/packages/python/3.8.5-anaconda
```

[1] https://docs.conda.io/projects/conda/en/latest/commands.html

3) Creating a virtual environment

- **Most frequently used commands**

Command	Description
`conda create -n ENVIRONMENT`	Create a virtual environment
`source activate ENVIRONMENT`	Activate a virtual environment
`conda deactivate`	Deactivate a virtual environment
`conda env list`	List all virtual environments
`conda env remove -n ENVIRONMENT`	Remove a virtual environment and all installed packages

CAUTION! NO CONFIRMATION! IRREVOCABLE!

[1] https://docs.conda.io/projects/conda/en/latest/commands.html

3) Creating a virtual environment

- **Most frequently used commands**

Command	Description
`conda create -n ENVIRONMENT`	Create a virtual environment
`source activate ENVIRONMENT`	Activate a virtual environment
`conda deactivate`	Deactivate a virtual environment
`conda env list`	List all virtual environments
`conda env remove -n ENVIRONMENT`	Remove a virtual environment and all installed packages

[1] https://docs.conda.io/projects/conda/en/latest/commands.html

Outlines

1. Why Conda?
   1) Problems
   2) Virtual environment & Conda
2. **Basic Usage**
   1) Get Conda
   2) Typical workflow
   3) Creating a virtual environment
   4) Installing software packages
3. Advanced Tips
   1) Where to get software?
   2) Change Conda path
   3) Share virtual environment
   4) Migrate / clone virtual environment
   5) Use virtual environment in Open OnDemand
   6) More than Python
   7) Troubleshooting
   8) Story time…

4) Installing software packages

- **Rule of thumb:**

Always use a virtual environment with Conda!

4) Installing software packages

a) Most frequently used commands

Command	Description
`conda install NAME`	Install a software package
`conda install NAME=VERSION`	Install a specific version
`conda install NAME -c CHANNEL`	Install from a specific channel (e.g., conda-forge, bioconda, nvidia, …)
`conda install NAME1 NAME2 …`	Install multiple packages at once (let conda work out dependencies)
`conda list`	List all installed software package

[1] https://docs.conda.io/projects/conda/en/latest/commands.html

b) Other useful commands

Command	Description
`conda search NAME`	Search available package versions

[1] https://docs.conda.io/projects/conda/en/latest/commands.html

4) Installing software packages

b) Other useful commands

Command	Description
`conda search NAME`	Search available package versions
`conda search NAME -c CHANNEL`	Search available package versions in a specific channel

[1] https://docs.conda.io/projects/conda/en/latest/commands.html

4) Installing software packages

b) Other useful commands

Command	Description
`conda search NAME`	Search available package versions
`conda search NAME -c CHANNEL`	Search available package versions in a specific channel
`conda search NAME --info`	Search available package versions with details

[1] https://docs.conda.io/projects/conda/en/latest/commands.html

b) Other useful commands

Command	Description
`conda search NAME`	Search available package versions
`conda search NAME -c CHANNEL`	Search available package versions in a specific channel
`conda search NAME --info`	Search available package versions with details
`conda update/upgrade NAME`	Update a package to the latest available version

[1] https://docs.conda.io/projects/conda/en/latest/commands.html

4) Installing software packages

b) Other useful commands

Command	Description
`conda search NAME`	Search available package versions
`conda search NAME -c CHANNEL`	Search available package versions in a specific channel
`conda search NAME --info`	Search available package versions with details
`conda update/upgrade NAME`	Update a package to the latest available version
`conda uninstall/remove NAME`	Remove a package

[1] https://docs.conda.io/projects/conda/en/latest/commands.html

4) Installing software packages

c) Bonus: Hot packages!

i. PyTorch (2.11.0, w/ GPU support)

```
$ conda create -n torch
$ source activate torch
$ conda install python=3.12
$ pip3 install torch==2.11.0 torchvision --index-url https://download.pytorch.org/whl/cu128
```

[1] https://pytorch.org/get-started/locally/

4) Installing software packages

c) Bonus: Hot packages!

ii. Tensorflow (2.21.0, w/ GPU support)

```
$ conda create -n tf
$ source activate tf
$ conda install python=3.12
$ pip install tensorflow[and-cuda]==2.21.0
```

[1] https://anaconda.org/anaconda/tensorflow-gpu

Summary

- **Rule of thumb:**

Always use a virtual environment with Conda!

Summary

- **Your workflow should mostly look like…**

To install …

```
$ conda create …

$ source activate …

$ conda install …
```

To use …

```
$ source activate …

$ # Do whatever you need
  to do with the packages
```

Exercise

- Create a virtual environment
- Search for SciPy version and install the second-latest version (as well as dependencies)
- After you are done, type in chat the installed SciPy and Python version

Outlines

1. Why Conda?
   1) Problems
   2) Virtual environment & Conda
2. Basic Usage
   1) Get Conda
   2) Typical workflow
   3) Creating a virtual environment
   4) Installing software packages
3. **Advanced Tips**
   1) Where to get software?
   2) Change Conda path
   3) Share virtual environment
   4) Migrate / clone virtual environment
   5) Use virtual environment in Open OnDemand
   6) More than Python
   7) Troubleshooting
   8) Story time…

A little more than the basics…

Outlines

1. Why Conda?
   1) Problems
   2) Virtual environment & Conda
2. Basic Usage
   1) Get Conda
   2) Typical workflow
   3) Creating a virtual environment
   4) Installing software packages
3. **Advanced Tips**
   1) Where to get software?
   2) Change Conda path
   3) Share virtual environment
   4) Migrate / clone virtual environment
   5) Use virtual environment in Open OnDemand
   6) More than Python
   7) Troubleshooting
   8) Story time…

1) Where to get software?

- **You can get software from a lot of places using Conda**
  - ➢ **Not that you should!**
- **Concerns?**
  - **Reliability** (some third-party or untested channels may not work)
  - **Security risk** (some untrustworthy publishers may pack something you don't know about)
- **Solution**
  - Always get from a source that **you can trust**

1) Where to get software?

- **Tier 1: Developer release (official release)**
  - On software's official website, look for "**Conda**".
  - E.g., PyTorch, Spyder, CudaToolKit

- **Tier 2: Trustworthy channels**

Name	Notes
conda-forge (default)	• Community supported, rule-enforced and generally trustworthy
bioconda	• Community supported for bioinformatics
nvidia / cuda	• Nvidia official channel
pytorch	• PyTorch official channel
intel	• Intel official channel
…	
main / anaconda / r	• *Anaconda default, supported by Anaconda Inc.* **(Do NOT recommend)**

Outlines

1. Why Conda?
   1) Problems
   2) Virtual environment & Conda
2. Basic Usage
   1) Get Conda
   2) Typical workflow
   3) Creating a virtual environment
   4) Installing software packages
3. **Advanced Tips**
   1) Where to get software?
   2) Change Conda path
   3) Share virtual environment
   4) Migrate / clone virtual environment
   5) Use virtual environment in Open OnDemand
   6) More than Python
   7) Troubleshooting
   8) Story time…

2) Change Conda path

- **Default Conda path**

	System-wide Conda modules	Customized Conda
Environments	/home/$USER/.conda/envs/	/path/to/conda/envs/
Cache	/work/$USER/.conda/pkgs/	/path/to/conda/pkgs/

```
[jasonli3@smic1 ~]$ balance
User filesystem quotas for jasonli3 (uid 15827):
     Filesystem          MB used    MB quota
     /home                   950        10000
     /work /project       329639            0      6

Storage allocation       MB used    MB quota
```

2) Change Conda path

a) Method 1: Command lines

– Usage: `conda config [options]`

```
$ conda config --add envs_dirs /path/to/envs
                                        Add path to environments
$ conda config --add pkgs_dirs /path/to/pkgs
                                        Add path to cache
```

[1] https://docs.conda.io/projects/conda/en/latest/commands/config.html

2) Change Conda path

b) Method 2: Configuration file

– Use any text editor to open: `~/.condarc`

```
$ vi ~/.condarc
```

```
envs_dirs:
  - /work/jasonli3/.conda/envs/
  - /project/jasonli3/.conda/envs
pkgs_dirs:
  - /work/jasonli3/.conda/pkgs
```

[1] https://docs.conda.io/projects/conda/en/latest/commands/config.html

2) Change Conda path

c) Places to store your virtual environments:

Location	Pros	Cons
/home	• All users have access • No expiration date • Backed up	• Limited quota (**10 GB**)
/project	• Larger quota (x 100 GB) • Valid for one year & renewable • Can be shared among group	• Not all users have access (PI must apply for /project drive)
/work	• All users have access • No quota limit	• Files are subject to **purge**!

2) Change Conda path

c) Places to store your virtual environments:

Location	Pros	Cons
/home	• All users have access • No expiration date • Backed up	• Limited quota (**10 GB**)
/project	• Larger quota (x 100 GB) • Valid for one year & renewable • Can be shared among group	• Not all users have access (PI must apply for /project drive)
/work	• All users have access • No quota limit	• Files are subject to **purge**!

Outlines

1. Why Conda?
   1) Problems
   2) Virtual environment & Conda
2. Basic Usage
   1) Get Conda
   2) Typical workflow
   3) Creating a virtual environment
   4) Installing software packages
3. **Advanced Tips**
   1) Where to get software?
   2) Change Conda path
   3) **Share virtual environment**
   4) Migrate / clone virtual environment
   5) Use virtual environment in Open OnDemand
   6) More than Python
   7) Troubleshooting
   8) Story time…

3) Share virtual environment

- **Scenario:**
  - I made a huge effort to install an extensive collection of software packages for our group's research needs. I don't want to do it all over again for everyone in our group. Is it possible to just share the virtual environment with them?

3) Share virtual environment

- **Solution:**

Step 1: PI

- Apply for a storage allocation (a.k.a. /project, if hasn't)
- Email sys-help@loni.org, request to add User 1 (sharing) and User 2 (shared) to /project

Step 2: User 1 (sharing):

- Set up `envs_dirs` to create a virtual environment in **a /project location**
- Install software in the virtual environment

Step 3: User 2 (shared):

- Set up `envs_dirs` to create a virtual environment in **the same /project location**

Outlines

1. Why Conda?
   1) Problems
   2) Virtual environment & Conda
2. Basic Usage
   1) Get Conda
   2) Typical workflow
   3) Creating a virtual environment
   4) Installing software packages
3. **Advanced Tips**
   1) Where to get software?
   2) Change Conda path
   3) Share virtual environment
   4) Migrate / clone virtual environment
   5) Use virtual environment in Open OnDemand
   6) More than Python
   7) Troubleshooting
   8) Story time…

4) Migrate / clone virtual environment

- **Scenario:**

  - I have been using LSU HPC cluster. But now I want to switch to LONI and run the exactly same virtual environment. How do I do that?

  - I am leaving. But I may continue doing similar research. How can I replicate my environment to a different HPC system in a different institute?

4) Migrate / clone virtual environment

- **Solution**

To …	Run command

[1] https://conda.io/projects/conda/en/latest/user-guide/tasks/manage-environments.html

4) Migrate / clone virtual environment

- **Solution**

To …

Export virtual environment recipe to fi

```
name: spyder
channels:
  - defaults
dependencies:
  - _libgcc_mutex=0.1=main
  - _openmp_mutex=5.1=1_gnu
  - arrow=1.2.3=py310h06a4308_1
  - astroid=2.14.2=py310h06a4308_0
  - attrs=22.1.0=py310h06a4308_0
  - babel=2.11.0=py310h06a4308_0
  - beautifulsoup4=4.11.1=py310h06a4308_0
  - black=22.6.0=py310h06a4308_0
  - blas=1.0=mkl
  - bottleneck=1.3.5=py310ha9d4c09_0
  - brotli=1.0.9=h5eee18b_7
```

[1] https://conda.io/projects/conda/en/latest/user-guide/tasks/manage-environments.html

4) Migrate / clone virtual environment

- **Solution**

To …	Run command
Export virtual environment recipe to file	`conda env export > myenv.yml`
Create a virtual environment from file	`conda env create -f myenv.yml`

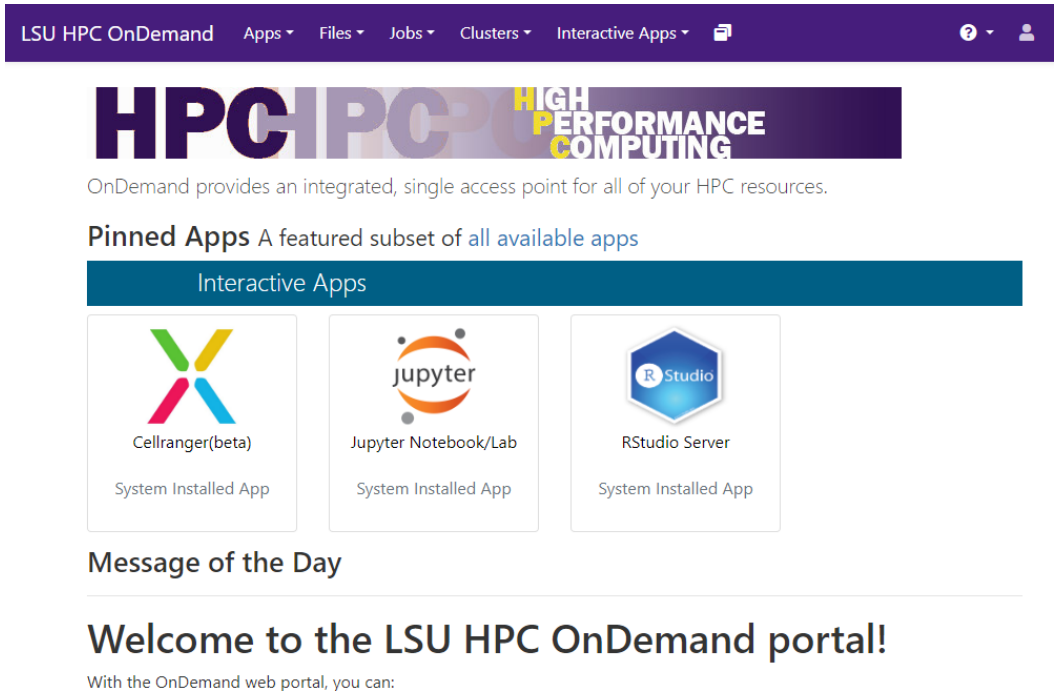
[1] https://conda.io/projects/conda/en/latest/user-guide/tasks/manage-environments.html

Outlines

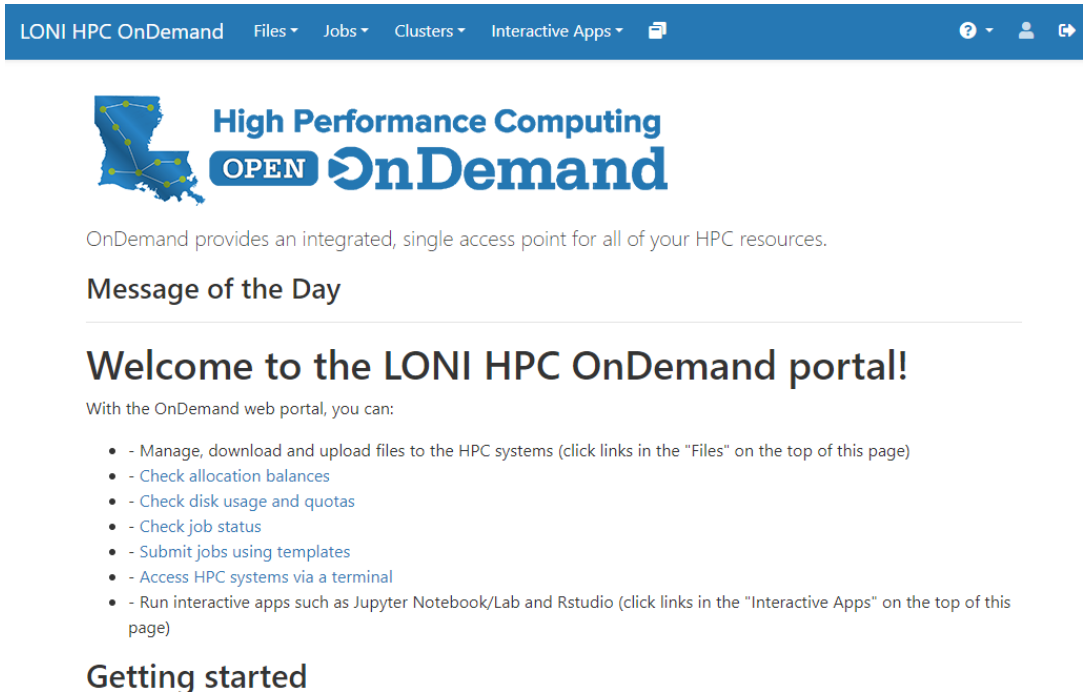
1. Why Conda?
   1) Problems
   2) Virtual environment & Conda
2. Basic Usage
   1) Get Conda
   2) Typical workflow
   3) Creating a virtual environment
   4) Installing software packages
3. Advanced Tips
   1) Where to get software?
   2) Change Conda path
   3) Share virtual environment
   4) Migrate / clone virtual environment
   5) Use virtual environment in Open OnDemand
   6) More than Python
   7) Troubleshooting
   8) Story time…

5) Use virtual environment in Open OnDemand

LSU HPC (SMIC / SuperMike 3)

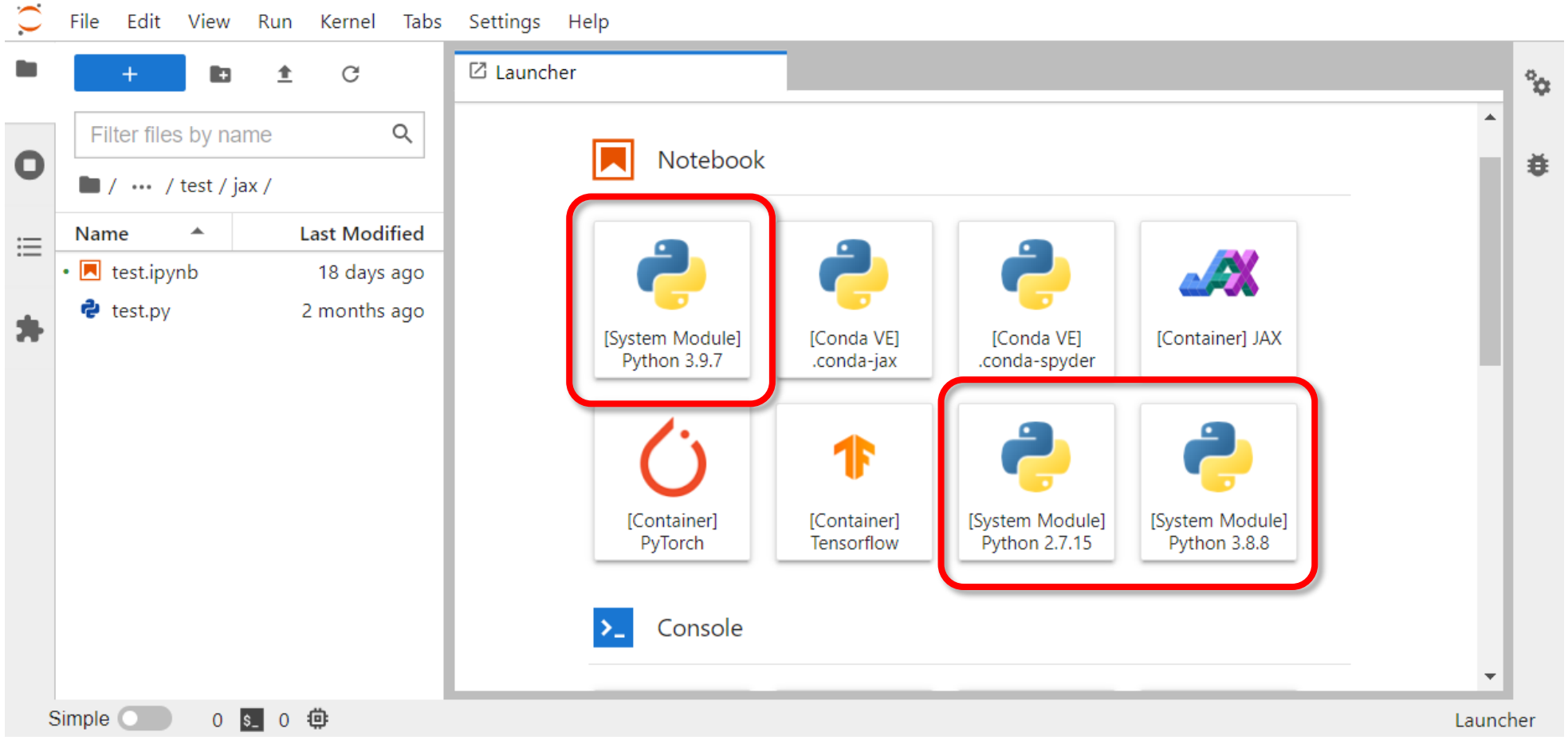


LONI (QB3 / QB4)



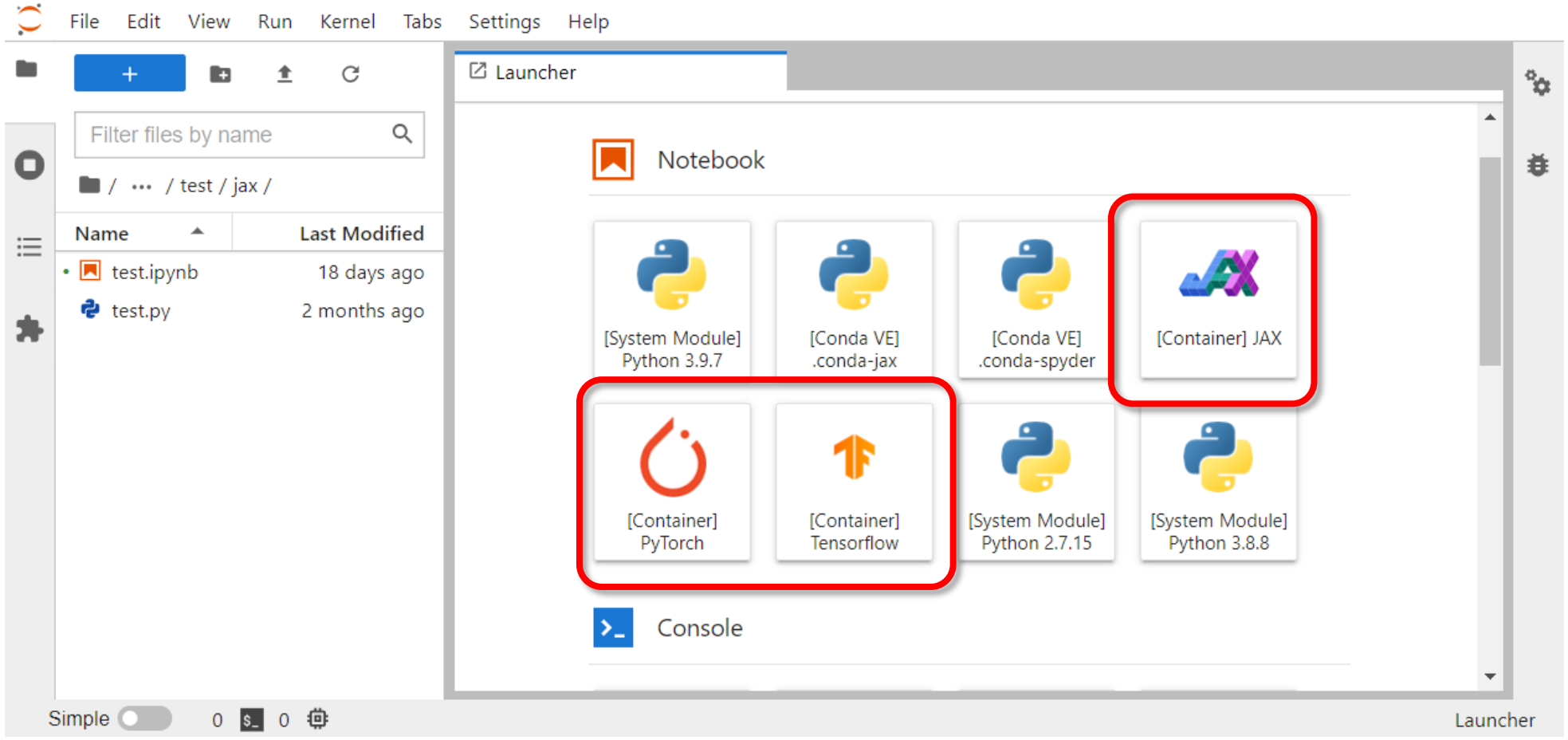
[1] https://youtu.be/pfo-v2BWDMY

5) Use virtual environment in Open OnDemand



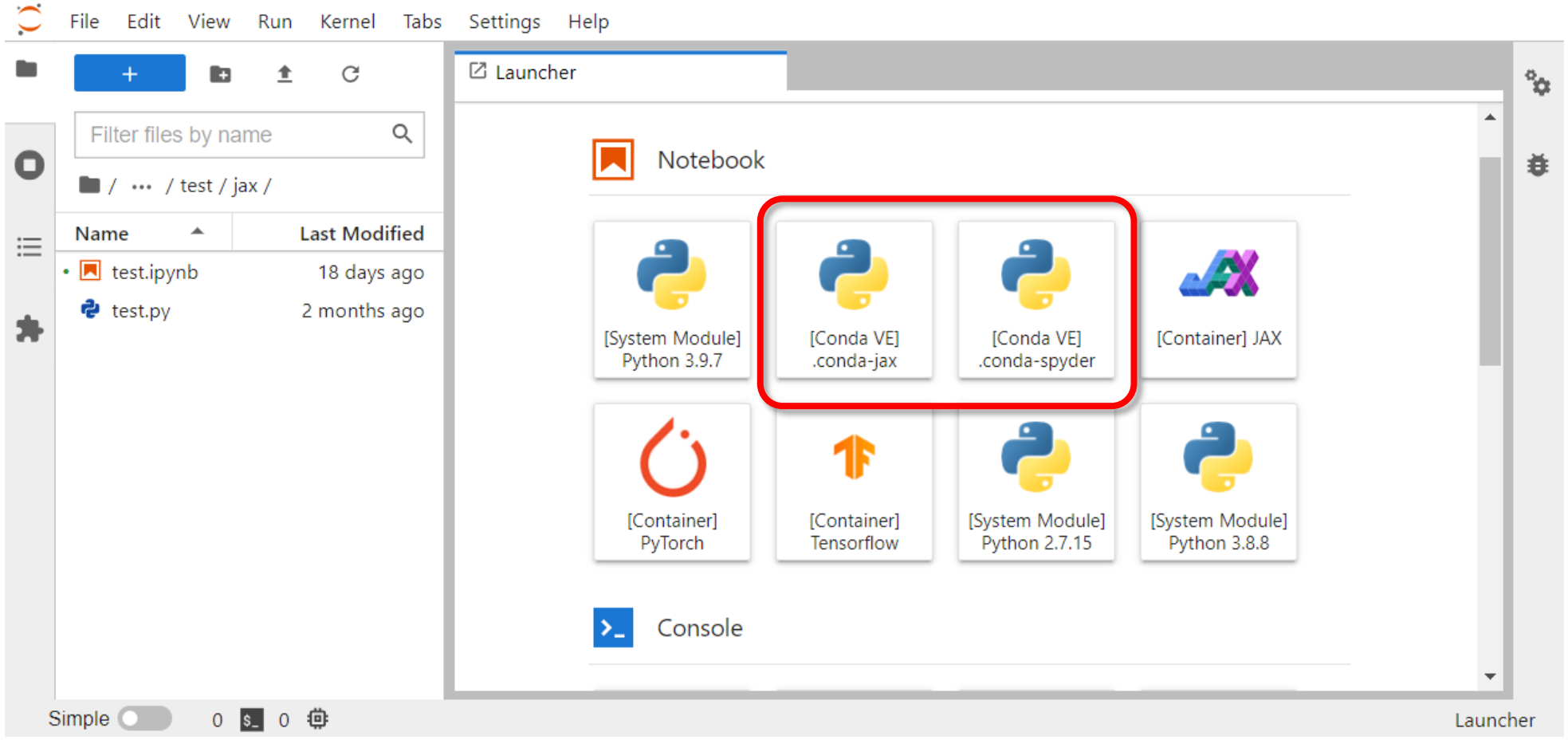
[1] https://youtu.be/pfo-v2BWDMY

5) Use virtual environment in Open OnDemand



[1] https://youtu.be/pfo-v2BWDMY

5) Use virtual environment in Open OnDemand



[1] https://youtu.be/pfo-v2BWDMY

5) Use virtual environment in Open OnDemand

- **How to:**

Step 1: `ssh` to the cluster where OOD is running on

LSU HPC	LONI
Super Mike 3	QB3, QB4

Step 2: Activate the virtual environment you want to use in Jupyter

```
$ source activate ENVIRONMENT
```

Step 3: Install ipykernel

```
$ conda install ipykernel                    # This
  works
$ # pip install ipkykernel                   # This does
  NOT work
```

Step 4: Start a Jupyter session in Open OnDemand, and choose the environment in **kernel**

[1] https://youtu.be/pfo-v2BWDMY

Outlines

1. Why Conda?
   1) Problems
   2) Virtual environment & Conda
2. Basic Usage
   1) Get Conda
   2) Typical workflow
   3) Creating a virtual environment
   4) Installing software packages
3. Advanced Tips
   1) Where to get software?
   2) Change Conda path
   3) Share virtual environment
   4) Migrate / clone virtual environment
   5) Use virtual environment in Open OnDemand
   6) More than Python
   7) Troubleshooting
   8) Story time…

6) More than Python

- **Scenario**
  - I need programming language support other than Python (e.g., R / Perl / Lua / …)
  - I need a different version than the system modules
  - The system-wide installation does not have what I need, or failed at certain things.

6) More than Python

- **Solutions**

Many non-python packages are managed by Conda too!

To install …		Run command …
Languages	R	`conda install R`
	Perl	`conda install perl`
	Julia	`conda install julia`
Dependencies	hdf5	`conda install hdf5`
	netcdf	`conda install libnetcdf`
	FFTW	`conda install fftw`
…		

[1] https://anaconda.org/

6) More than Python

- **It gets even better…**
  - You can use language specific package management tools

Language	Tool
Python	`pip`
R	`install.packages`
Perl	`cpan`
Julia	`Pkg`
…	

  - Packages will be **isolated** in the virtual environment

6) More than Python

- **E.g.: Conda is good at managing R packages, too!**

  – Use system's R module:

```
$ module load r
$ R
> install.packages("Seurat")
```

  – Use Conda (usually faster):

```
$ conda create -n seurat
$ source activate seurat
$ conda install r-seurat
```

Outlines

1. Why Conda?
   1) Problems
   2) Virtual environment & Conda
2. Basic Usage
   1) Get Conda
   2) Typical workflow
   3) Creating a virtual environment
   4) Installing software packages
3. Advanced Tips
   1) Where to get software?
   2) Change Conda path
   3) Share virtual environment
   4) Migrate / clone virtual environment
   5) Use virtual environment in Open OnDemand
   6) More than Python
   7) Troubleshooting
   8) Story time…

7) Troubleshooting

a) Cannot change default conda

– **Issues**:

- I have been using an older version of Conda and ran `conda init` before.
  Now I am stuck with the old one and cannot switch to the new Conda module even if I load it.

– **Solution**:

- Run these commands in terminal:

```
$ unset conda                    # Unset old function "conda"
$ module purge                   # Purge all modules
$ module load conda              # Load the latest conda module
$ conda init                     # Set the new one as default
```

b) Conflict with system module

– **Issues**:

```
(torch) [jasonli3@qbd489 ~]$ module li
Currently Loaded Modulefiles:
 1) python/3.11.5-anaconda
(torch) [jasonli3@qbd489 ~]$ python
Python 3.11.5 (main, Sep 11 2023, 13:54:46) [GCC 11.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import torch
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ModuleNotFoundError: No module named 'torch'
```

Fail!

– **Solution**:

Do NOT load system module if you are using your own installation!

c) Package corrupted?

– **Issues**:

• I got a bunch of these errors trying to install something…

```
Proceed ([y]/n)?


Downloading and Extracting Packages:

Preparing transaction: done
Verifying transaction: failed

CondaVerificationError: The package for python located at /work/jasonli3/.conda/pkgs/python-3.12.5-h2ad013b_0_cpython
appears to be corrupted. The path 'lib/libpython3.12.so'
specified in the package manifest cannot be found.

CondaVerificationError: The package for python located at /work/jasonli3/.conda/pkgs/python-3.12.5-h2ad013b_0_cpython
appears to be corrupted. The path 'lib/libpython3.12.so.1.0'
specified in the package manifest cannot be found.
```

c) Package corrupted?

– **Reason**:

- Cache files under `/work/$USER/` were partially purged.

– **Solution**:

```
$ conda clean -f
```

** This will clean all cache files. It is highly unlikely to cause problems nowadays. But still, beware.*

d) What if I made a mess?

– **Issues**:

- I mixed conda / pip back and forth, and broke the environment…
- I tried to add a package in my existing environment, but Conda failed at solving the environment…
- I tried to `conda upgrade` a package in my environment, but Conda failed at solving the environment…

– **Solution**:

- It may be easier to create a new virtual environment and start fresh…

Outlines

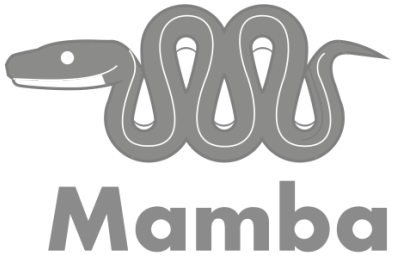
1. Why Conda?
   1) Problems
   2) Virtual environment & Conda
2. Basic Usage
   1) Get Conda
   2) Typical workflow
   3) Creating a virtual environment
   4) Installing software packages
3. **Advanced Tips**
   1) Where to get software?
   2) Change Conda path
   3) Share virtual environment
   4) Migrate / clone virtual environment
   5) Use virtual environment in Open OnDemand
   6) More than Python
   7) Troubleshooting
   8) Story time…

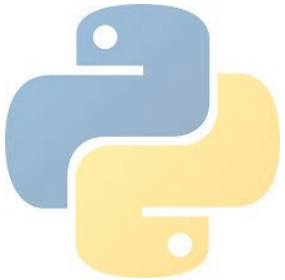
8) Story time…

- **If you are a first-time Conda user…**
  - Don’t worry about this part. Just enjoy the stories. ☺
- **If you have used Conda before…**
  - This part may be relevant to you.

8) Story time…

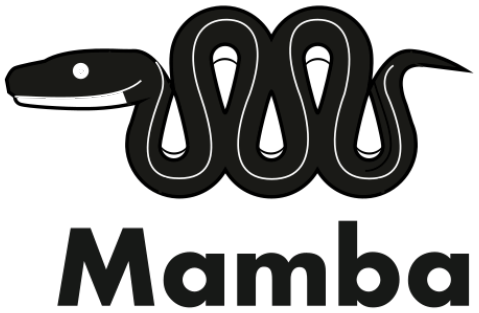
a) What is this “Mamba” I kept hearing about? Do I need it?





8) Story time…

a) What is this “Mamba” I kept hearing about? Do I need it?




```
conda install -c conda-forge mamba
mamba install -c conda-forge -c bioconda busco=5.6.1
```

8) Story time…

a) What is this "Mamba" I kept hearing about? Do I need it?

– **Mamba**:

"A better Conda"

- A **drop-in replacement** of Conda
- **Faster**, and **resolve Conda failures**

– A landmark handshake (Jul 2023)

- Mamba solver is now included in Conda
- Do not **HAVE TO** use Mamba, if already using our Conda module
- You **CAN** if you want → Available when you load Conda module

```
(base) [jasonli3@mike4 ~]$ mamba --version
mamba 1.5.6
conda 23.11.0
(base) [jasonli3@mike4 ~]$
```

8) Story time…

b) The Anaconda drama…

1) Get Conda

b) Install your own Conda (only if needed)

– Conda distributions

Popular Distribution	Desc
Anaconda	Full size Conda + Python, w/ a LO… Anaconda Inc.
Miniconda	Minimum size Conda + Python onl…
Miniforge	Minimum size Conda + Python onl…

LSU INFORMATION TECHNOLOGY SERVICES

1. Why Conda? | 2. Basic Usage

1) Where to get software?

- **Tier 1: Developer release (official release)**
  - On software's official website, look for "**Conda**".
  - E.g., PyTorch, Spyder, CudaToolKit
- **Tier 2: Trustworthy channels**

Name	Notes
conda-forge (default)	• Community supported, rule-enforced and generally trustworthy
bioconda	• Community supported for bioinformatics
nvidia / cuda	• Nvidia official channel
pytorch	• PyTorch official channel
intel	• Intel official channel
	…
main / anaconda / r	• *Default channel, officially managed by Conda* **(We do NOT recommend)**

LSU INFORMATION TECHNOLOGY SERVICES

NI

1. Why Conda? | 2. Basic Usage | 3. Advanced Tips 71

LSU INFORMATION TECHNOLOGY SERVICES

NI

8) Story time…

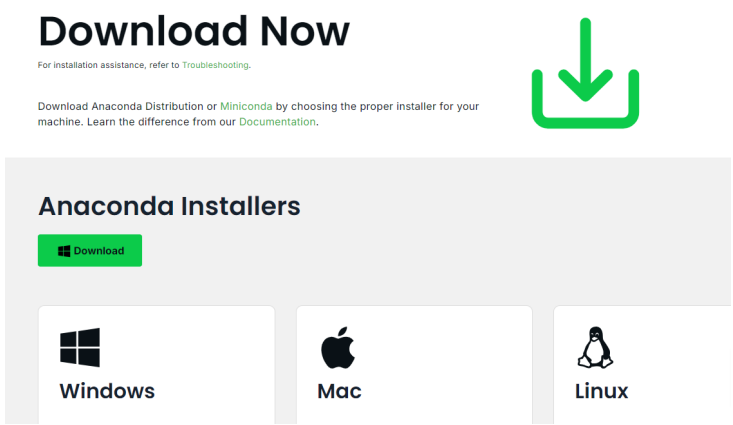
b) The Anaconda drama…

– “**Anaconda**” disambiguation…

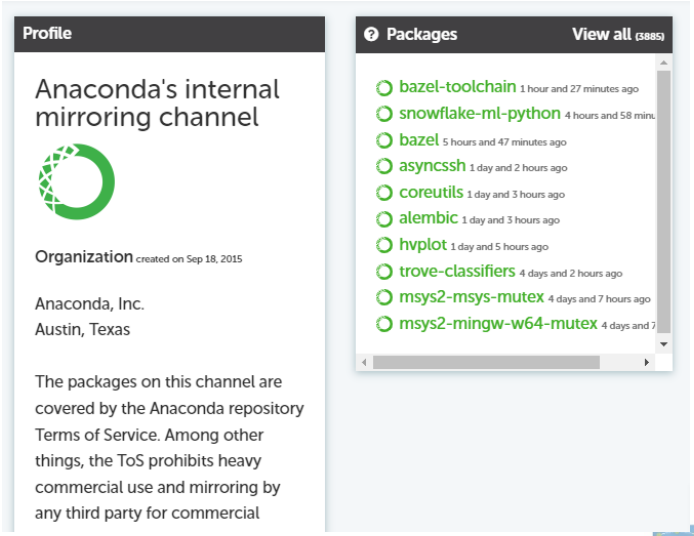
Company
(Anaconda Inc.)

ANACONDA®

Software
(Conda distribution)



Channel
(`conda install -c anaconda …`)



8) Story time…

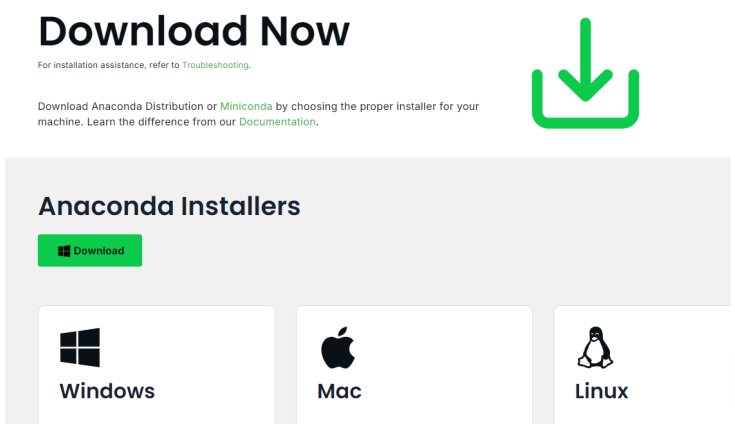
b) The Anaconda drama…

– “**Anaconda**” disambiguation…

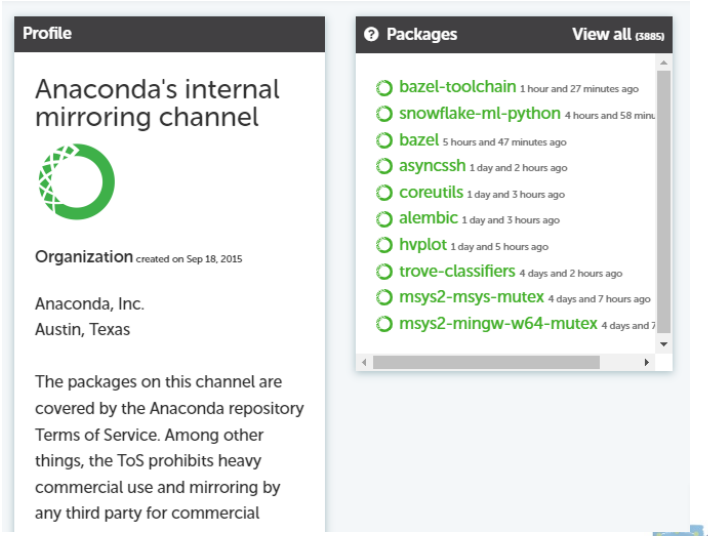
Company
(Anaconda Inc.)

ANACONDA®

Software
(Conda distribution)



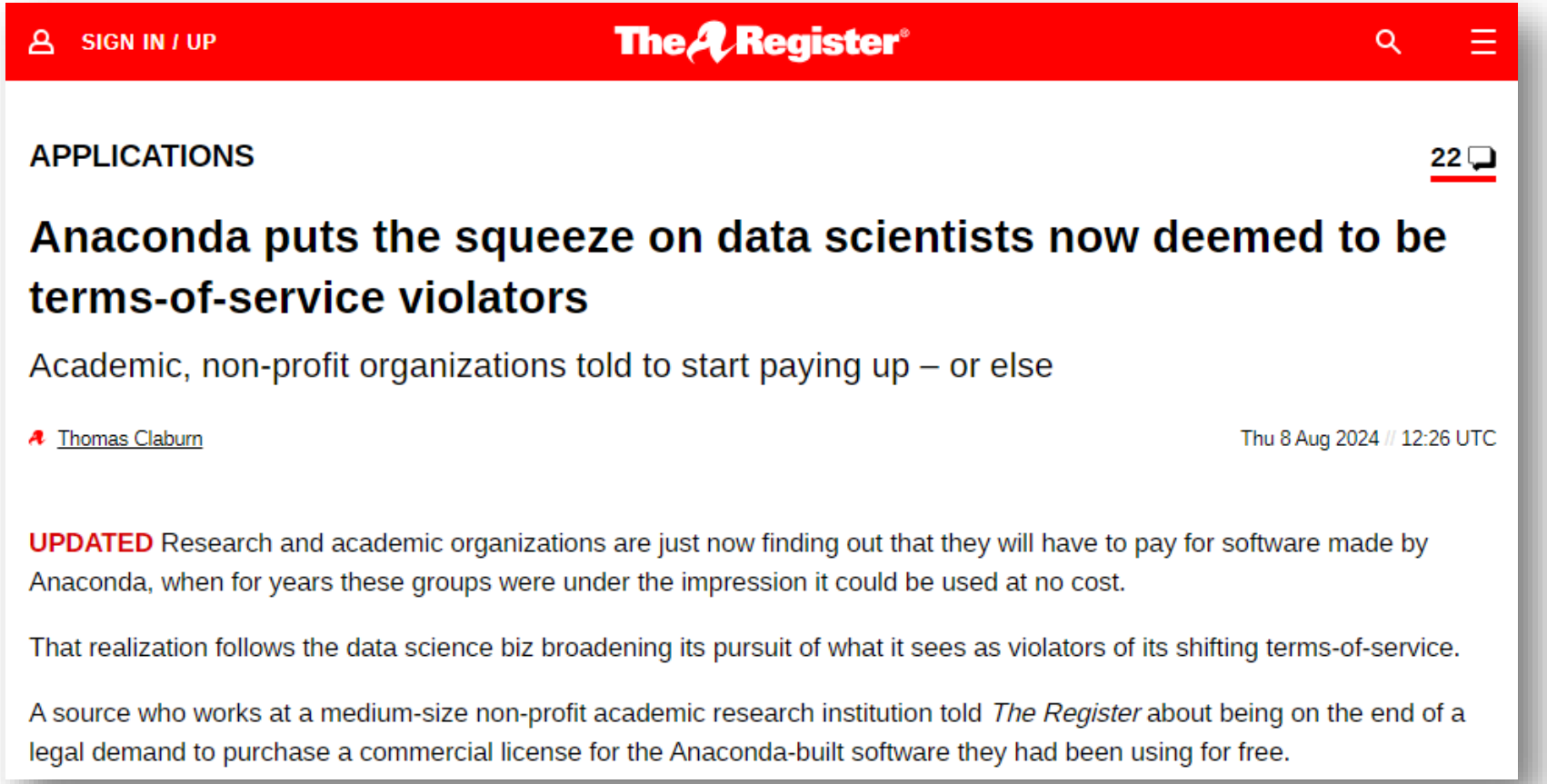
Channel
(`conda install -c anaconda …`)



LSU INFORMATION TECHNOLOGY SERVICES

8) Story time…

b) The Anaconda drama…



SIGN IN / UP

The Register

APPLICATIONS

22

Anaconda puts the squeeze on data scientists now deemed to be terms-of-service violators

Academic, non-profit organizations told to start paying up – or else

Thomas Claburn

Thu 8 Aug 2024 // 12:26 UTC

UPDATED Research and academic organizations are just now finding out that they will have to pay for software made by Anaconda, when for years these groups were under the impression it could be used at no cost.

That realization follows the data science biz broadening its pursuit of what it sees as violators of its shifting terms-of-service.

A source who works at a medium-size non-profit academic research institution told *The Register* about being on the end of a legal demand to purchase a commercial license for the Anaconda-built software they had been using for free.

[1] https://www.theregister.com/2024/08/08/anaconda_puts_the_squeeze_on

b) The Anaconda drama…

– **Solution**:

- HPC communities are moving away from anaconda channels.
- Embracing the community supported **conda-forge** as alternative.

– **What you should know:**

- For **ALL** users: Avoid using **main** / **anaconda** / **r** channels in future installations

If you …	Please…
Are using Conda modules on clusters	• Don't worry about it. (We already set all default channels to **conda-forge**)
Will install your own Conda	• Choose **Miniforge**.
Are already using your own Conda	• Set your priority channel to **conda-forge**. (Run `conda config --add channels conda-forge`)

Conclusion

Take home message

- **Rule of thumb:**

Always use a virtual environment with Conda!

Take home message

To install …

```
$ conda create …

$ source activate …

$ conda install …
```

To use …

```
$ source activate …

$ # Do whatever you need
  to do with the packages
```

Next week in our miniseries

Conda vs Singularity

	Conda / Virtual Environments	Singularity / Containers
Availability	All users	All users, but may need additional things
Self-contained	Yes	Yes
Isolated	Yes (but still accessible from outside)	Perfect (completely isolated from outside)
Editability	Yes	No (Must create a new image)
Disk usage	Large	Smaller
Portability	Possible (but .yml may not work)	Great (just copy-paste one file)
Security	Fair	Good
Ease of use	Good	May require a little more understanding

Conda vs Singularity

	Conda / Virtual Environments	Singularity / Containers
Good for	• Less hassle to create and install software from scratch • If you need to frequently make modifications	• Less hassle if the developer releases a working container • If you don't need to make changes after it is created • Portability • Reduce disk usage • Your system admins yelled at you about security risk

Contact us

- **Contact user services**
  - Email Help Ticket: sys-help@loni.org
  - Telephone Help Desk: +1 (225) 578-0900