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
Mar 6, 2024

Magic Tools to Install & Manage Software

Part 1:  **Virtual Environment**

Part 2:  **ingularity 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**

Outlines

1) Problems

- **Core problem:**

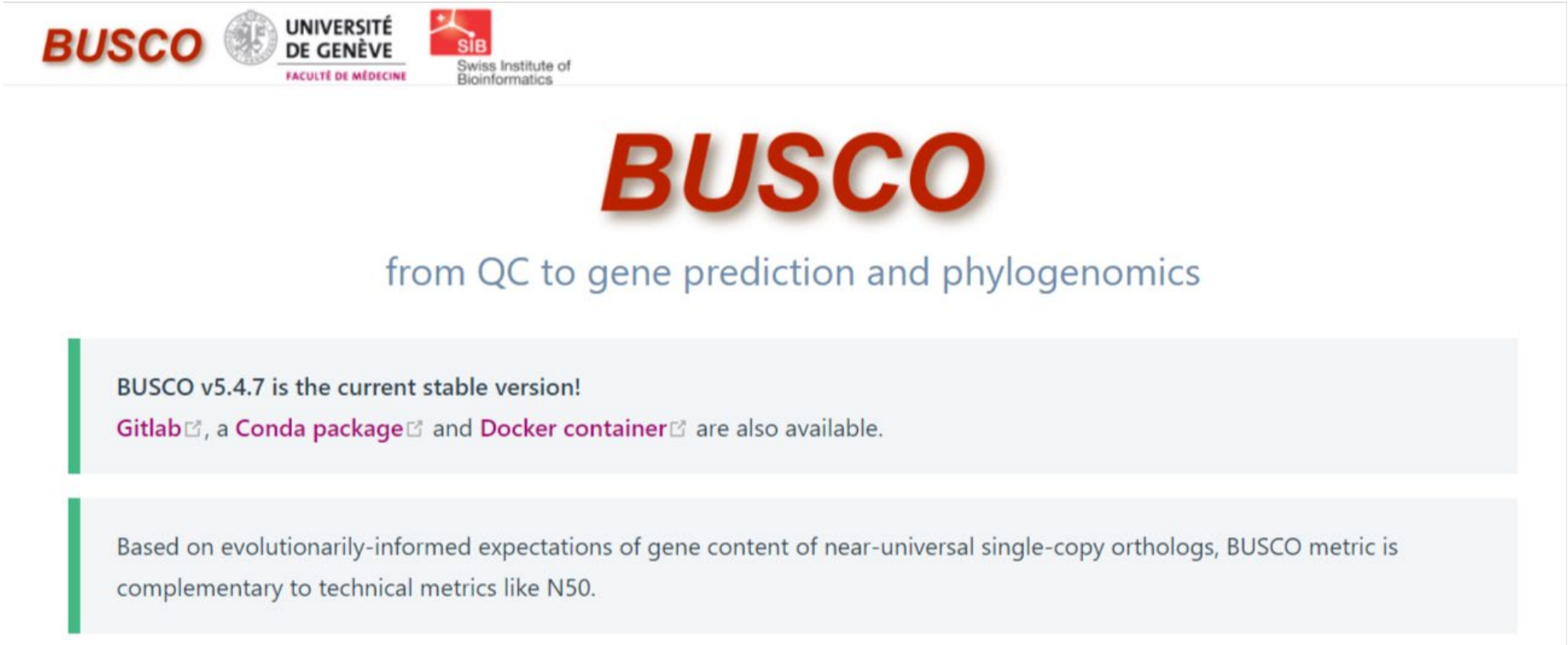
Installing software on an HPC system

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.

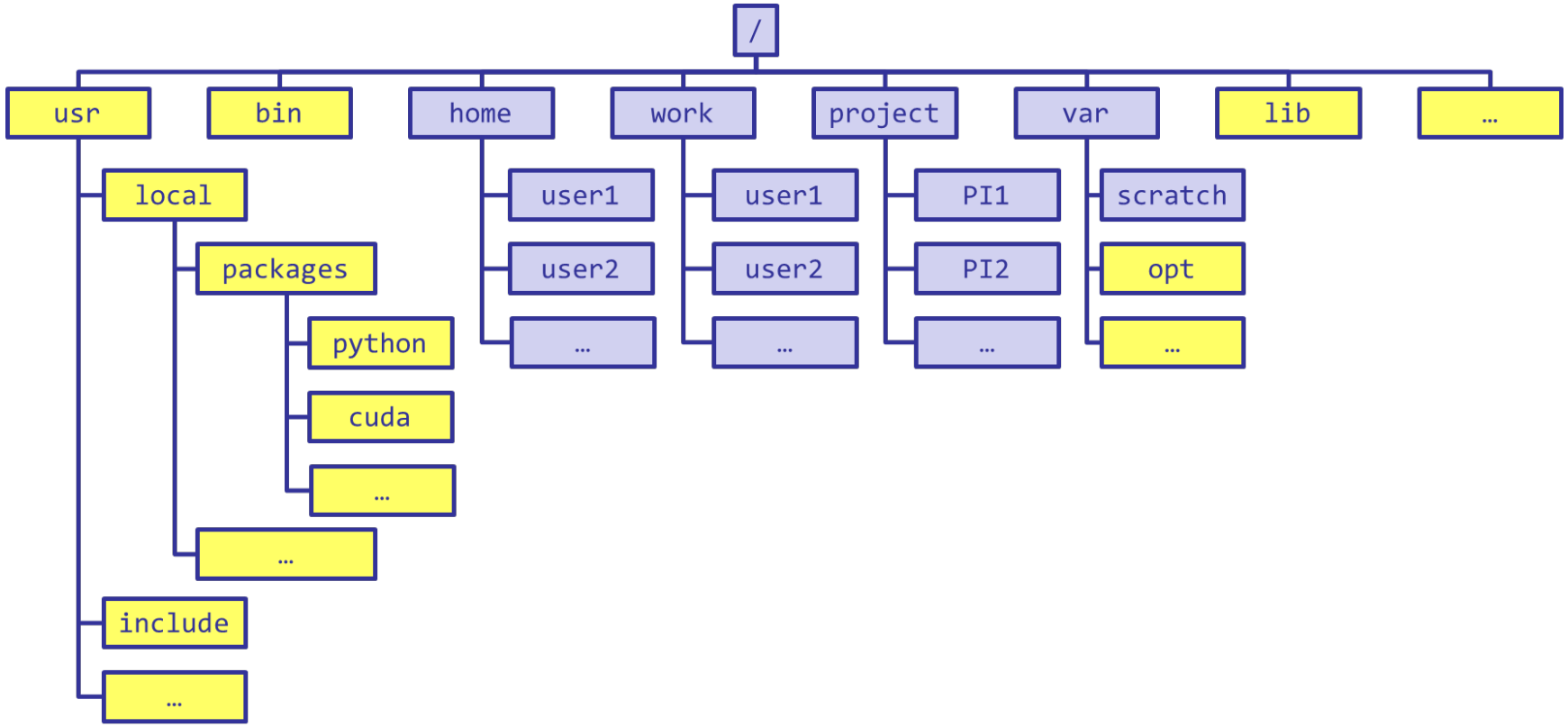
- Dependencies

The following dependencies are required for AUGUSTUS:

- for gzip compressed input: (set ZIPINPUT = false in common.mk if t… are 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!)



1) Problems

b) Permission denied (Welcome to HPC!)

```
[jasonli3@smic2 ~]$ module load python/3.6.2-anaconda-tensorflow
[jasonli3@smic2 ~]$ module li
Currently Loaded Modulefiles:
 1) python/3.6.2-anaconda-tensorflow
```

1) Problems

b) Permission denied (Welcome to HPC!)

```
[jasonli3@smic2 ~]$ module load python/3.6.2-anaconda-tensorflow
[jasonli3@smic2 ~]$ module li
Currently Loaded Modulefiles:
 1) python/3.6.2-anaconda-tensorflow
[jasonli3@smic2 ~]$ pip install geos
Collecting geos
  Downloading https://files.pythonhosted.org/packages/49/5b/b8acf74c01187a36aa41b6523deb9baa59c
    100% |                                | 409kB 3.0MB/s
```

1) Problems

b) Permission denied (Welcome to HPC!)

```
[jasonli3@smic2 ~]$ module load python/3.6.2-anaconda-tensorflow
[jasonli3@smic2 ~]$ module li
Currently Loaded Modulefiles:
 1) python/3.6.2-anaconda-tensorflow
[jasonli3@smic2 ~]$ pip install geos
Collecting geos
  Downloading https://files.pythonhosted.org/packages/49/5b/b8acf74c01187a36aa41b6523deb9baa59c
    100% |                                | 409kB 3.0MB/s
```

```
    os.makedirs(path)
  File "/usr/local/packages/python/3.6.2-anaconda/lib/python3.6/os.py", line 220, in makedirs
    mkdir(name, mode)
PermissionError: [Errno 13] Permission denied: '/usr/local/packages/python/3.6.2-anaconda/lib/p
You are using pip version 9.0.1, however version 23.0.1 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.
[jasonli3@smic2 ~]$
```

1) Problems

b) Permission denied (Welcome to HPC!)

- If you ask Google / ChatGPT…

```
$ sudo yum install …

$ sudo apt-get install …

$ sudo make install
```

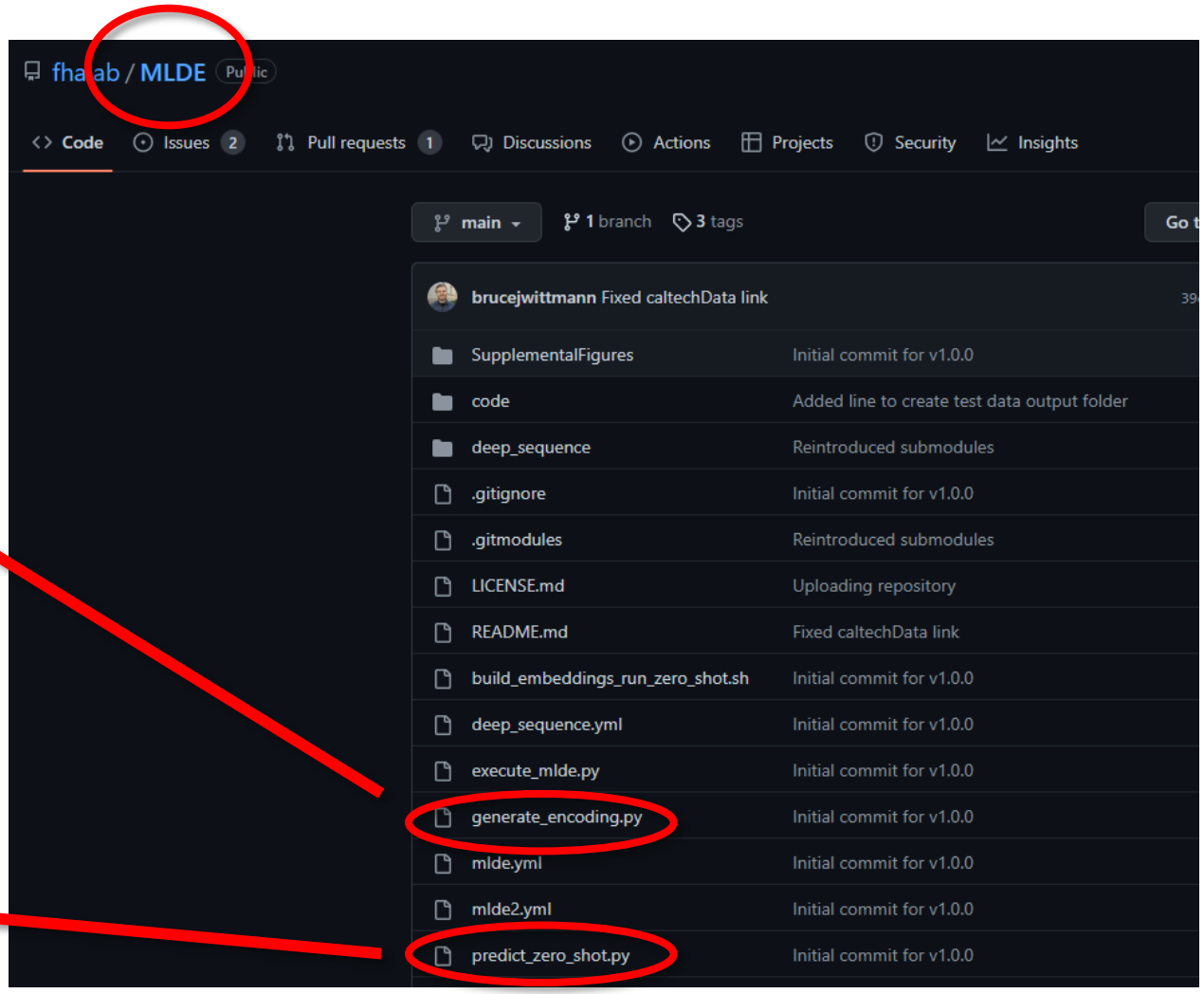
1) Problems

c) Conflicted packages

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

Tensorflow 1.13

PyTorch > 1.5



d) Sharing / Migrating your environment

– 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

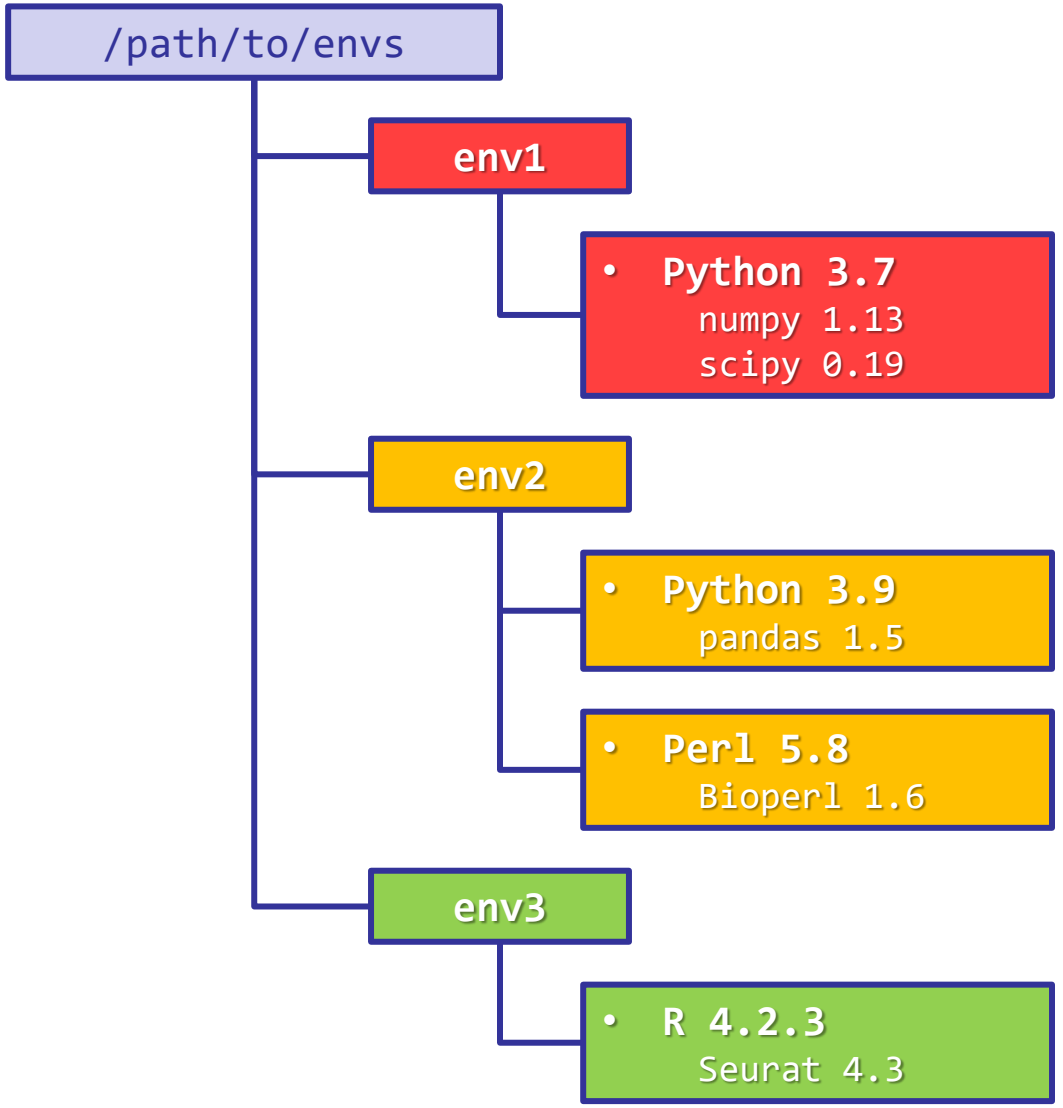
1. **Why Conda?**
   1) Problems
   2) Virtual environment & Conda
2. **Basic Usage**
   1) Get Conda
   2) Typical workflow
   3) Creating a virtual environment
   4) Installing software packages
3. **Advanced Tips**

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., environmental variables)



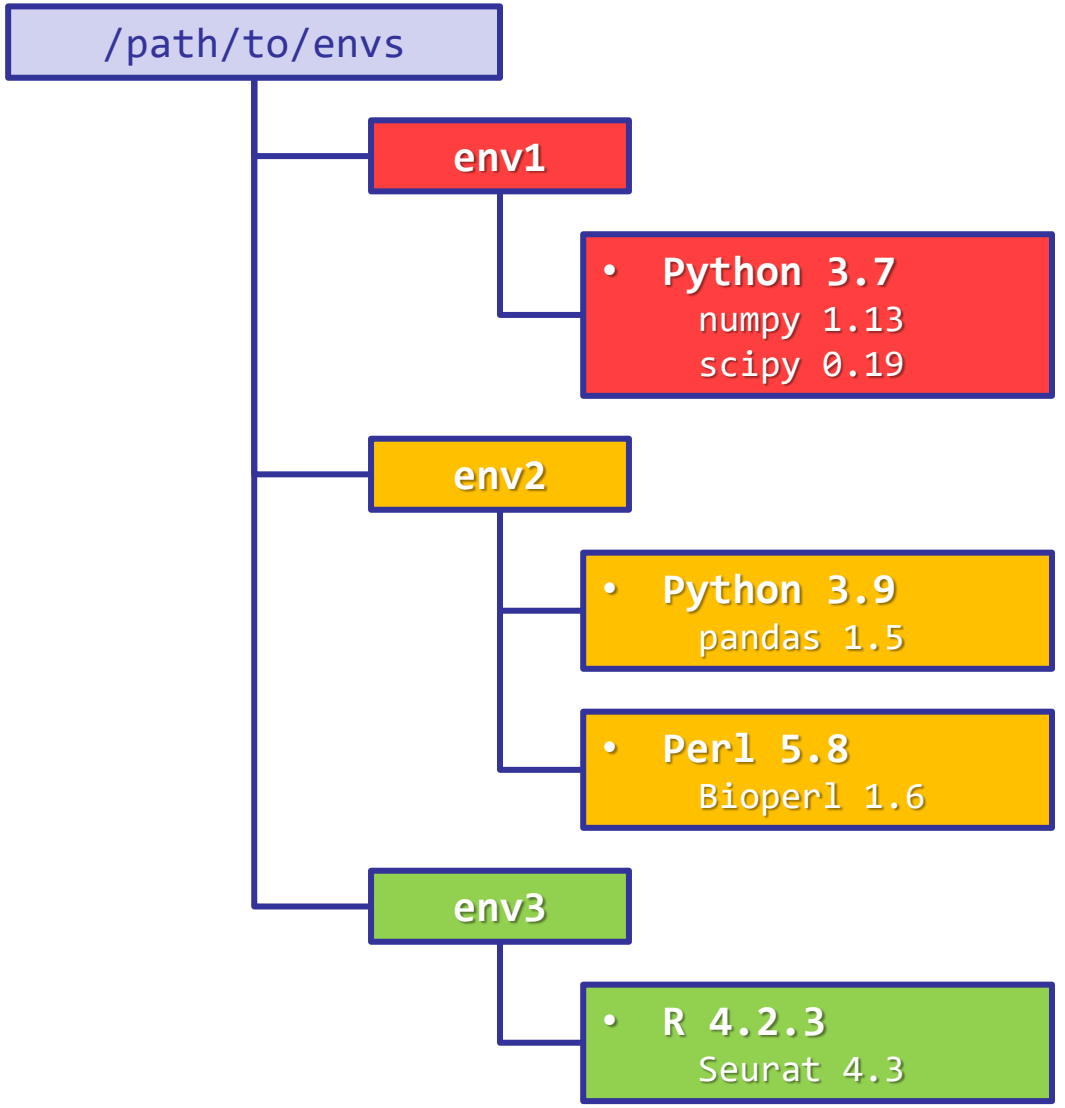
2) Virtual environment & Conda

a) What is a virtual environment?

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

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

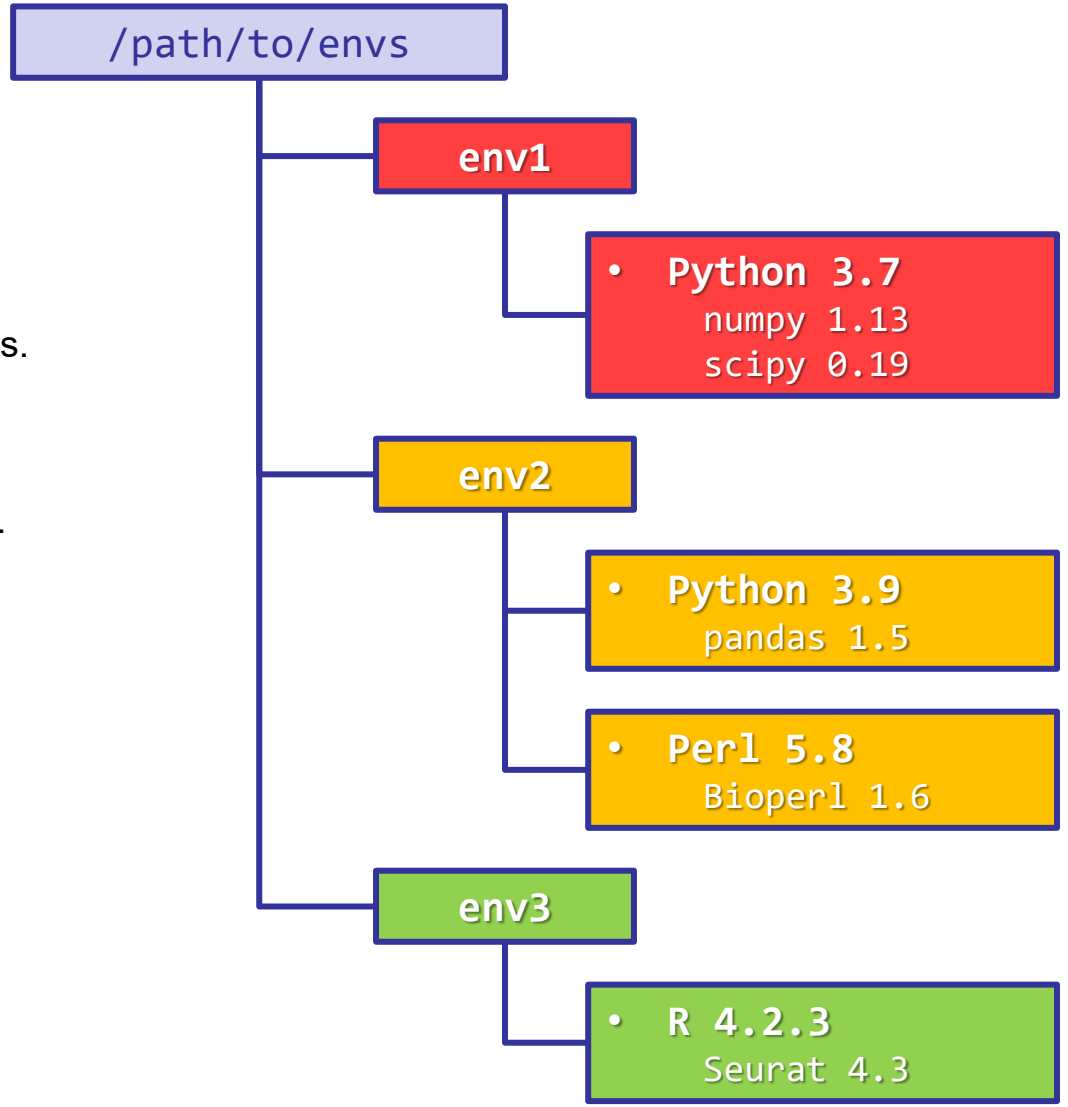
    Whatever happens in a VE stays in that VE…



2) Virtual environment & Conda

b) How does it solve my problems?

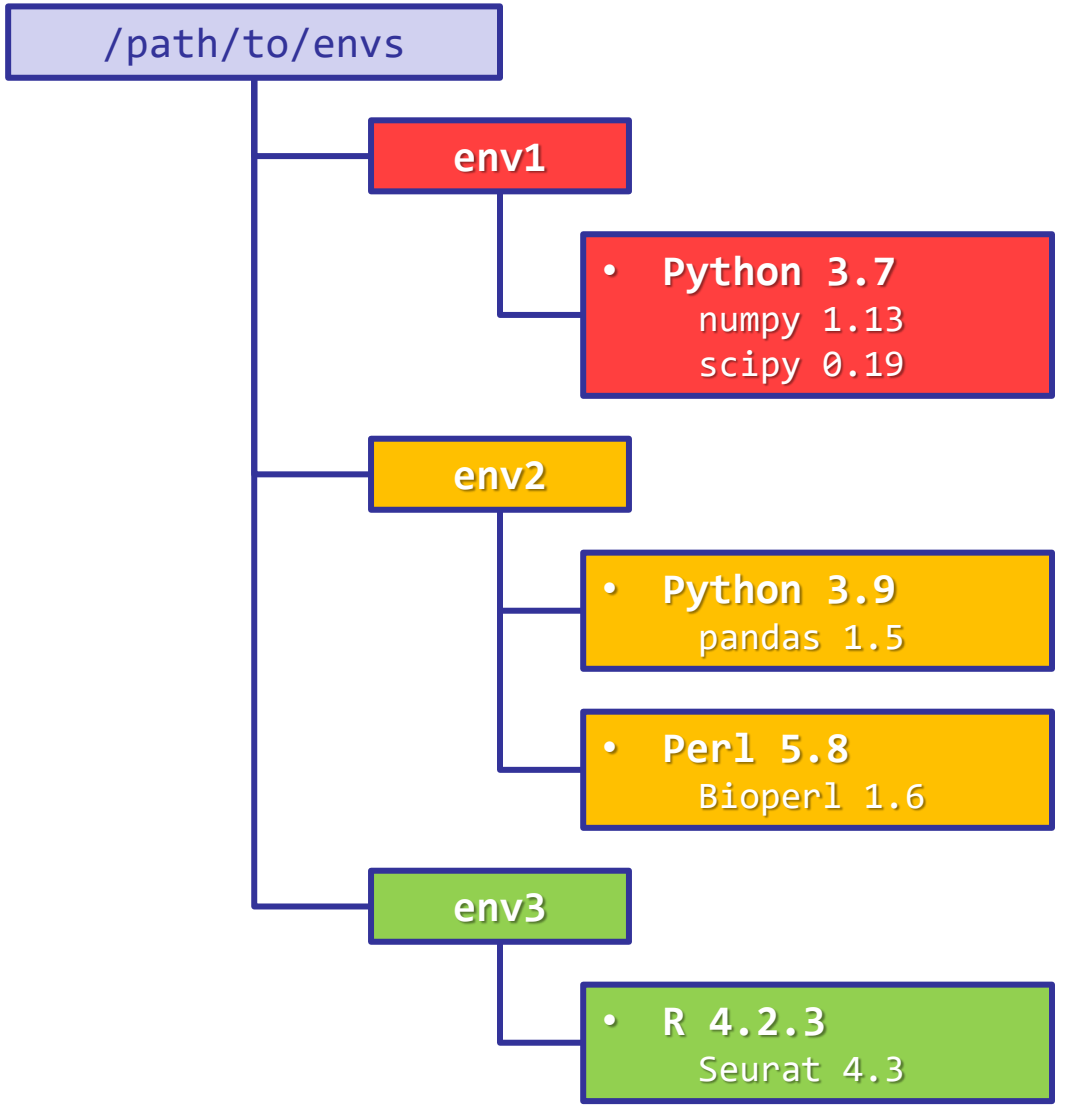
- **Dependency issue**
  - Pack all dependencies in the VE. No need to rely on the system.
  - Tools like **Conda** can help install dependencies.
- **Permission issue**
  - Create a VE where you have write permission.
  - Does not need **sudo** permission
- **Conflicted packages**
  - Install in different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



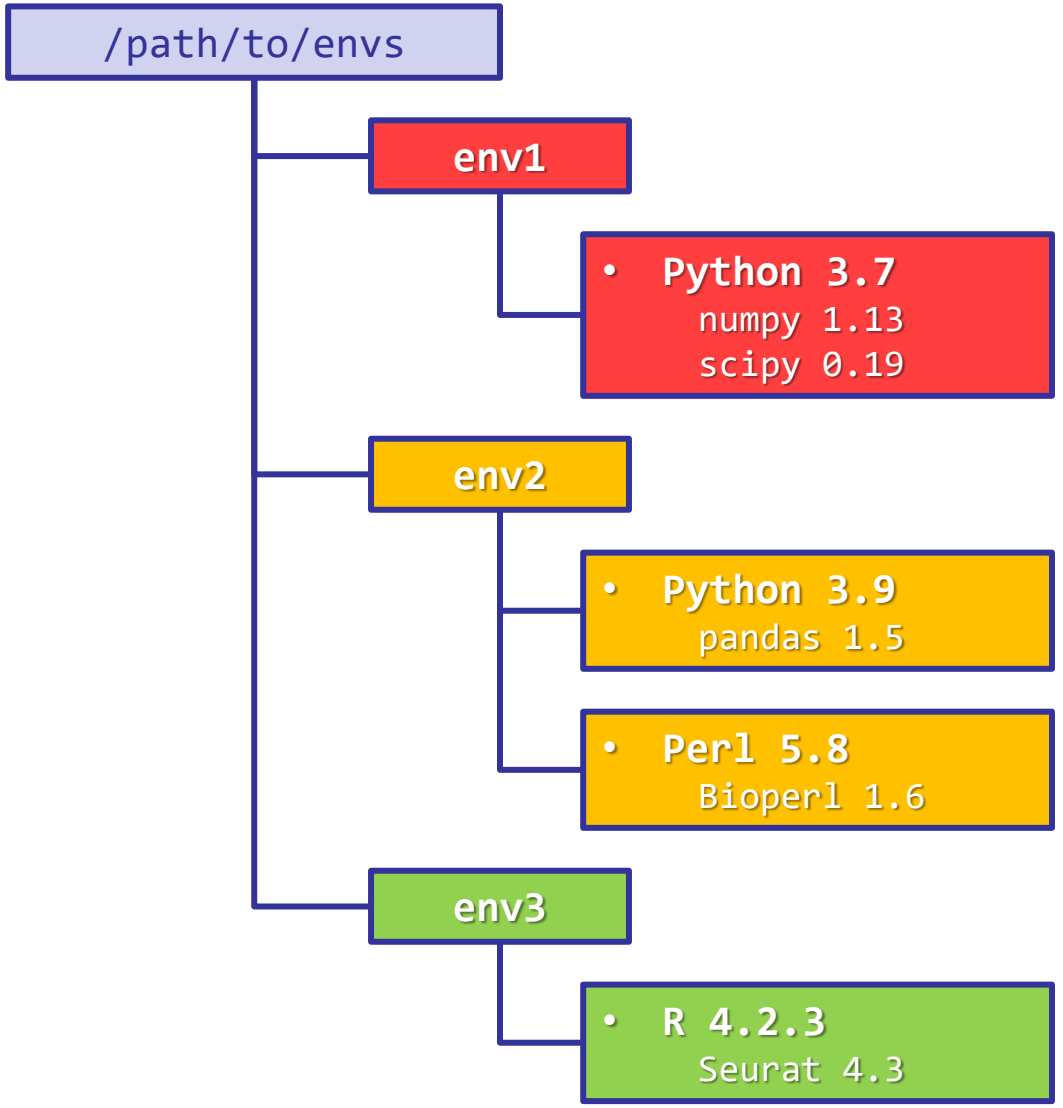
2) Virtual environment & Conda

c) What is Conda?

Technology →



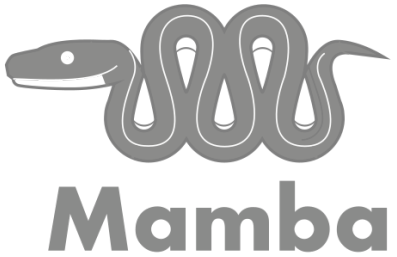
c) What is Conda?

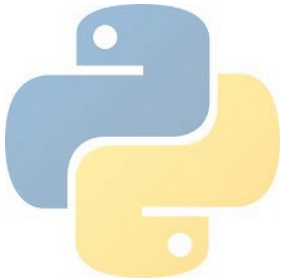


↑ **Software** system that implements the technology

2) Virtual environment & Conda

c) What is Conda?





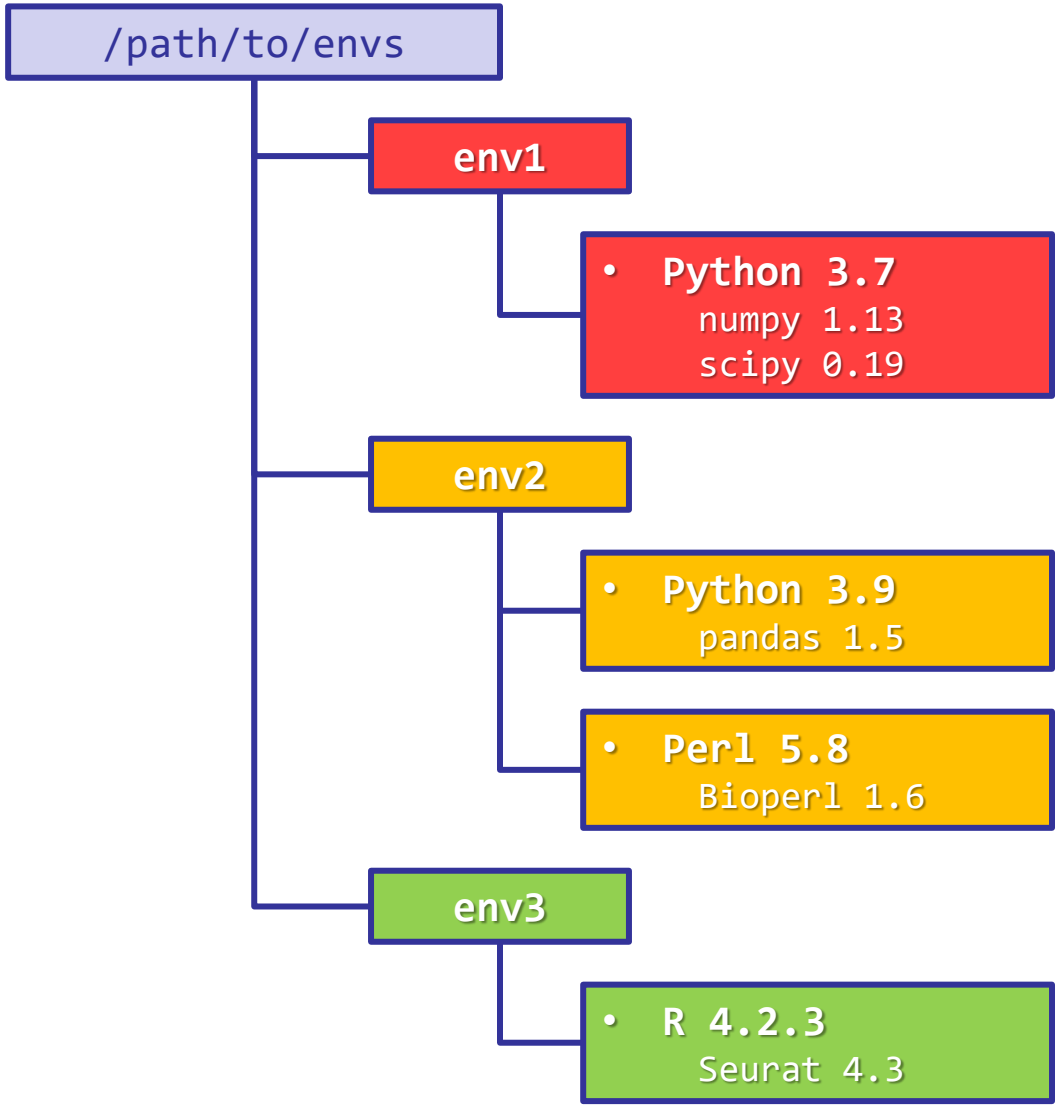
c) What is Conda?

Summary

Technology that helps
with software installation →

↓ **Software** system that
implements the technology

CONDA



Before we continue…

- **In general:**

If a software package you need is managed by Conda,
you (most likely) can install / manage it without needing our help

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

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) 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 (recommended)

1) Get Conda

b) Install miniconda

– Latest version: https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh

```
$ wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh
```
Step 1: Download miniconda

```
$ chmod u+x Miniconda3-latest-Linux-x86_64.sh
```
Step 2: Enable execution

```
$ ./Miniconda3-latest-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

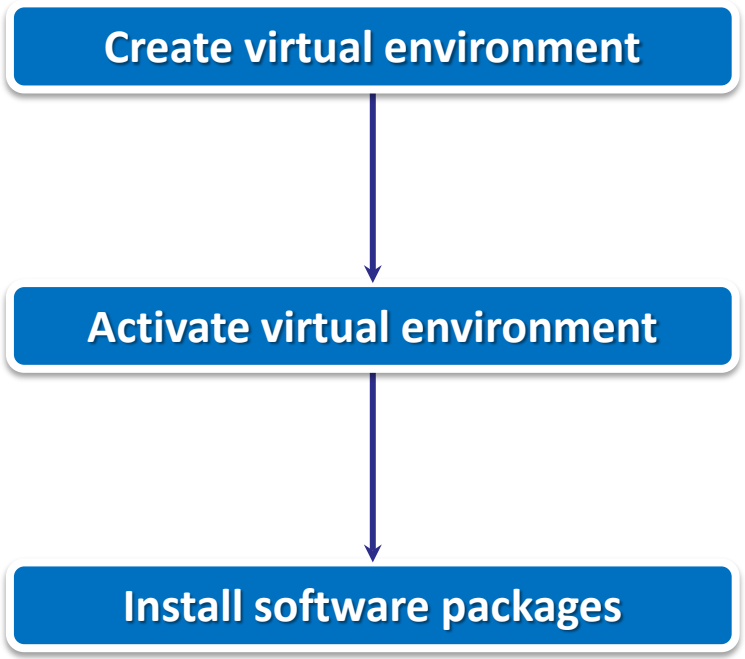
2) Typical workflow

- **Key:**

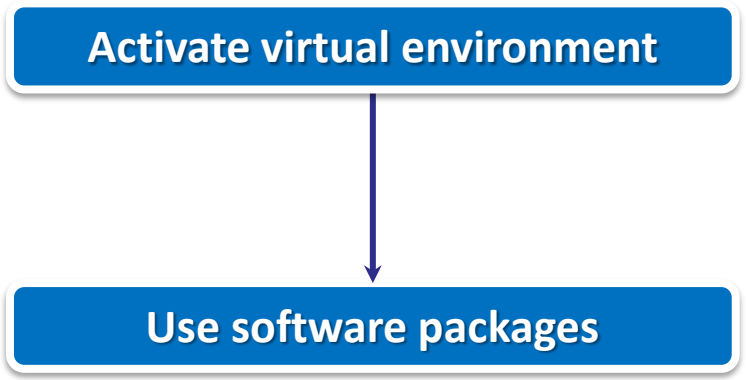
Always use a virtual environment!

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

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

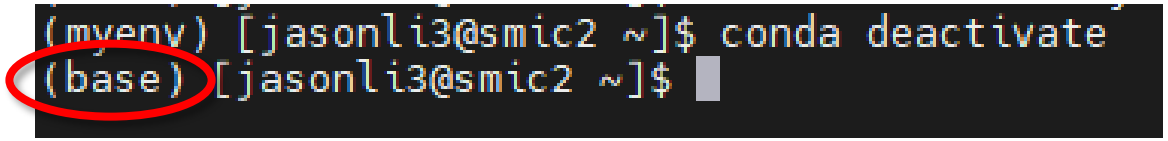
```
(base) [jasonli3@smic2 ~]$ source activate myenv
(myenv) [jasonli3@smic2 ~]$
```

[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

4) Installing software packages

- **Before installation…**

Make sure a virtual environment is activated!

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

4) Installing software packages

b) Other useful commands

Command	Description
`conda search NAME`	Search available package versions

[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

[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

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

[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 (2.2.0, w/ GPU support)

```
$ conda create -n torch
$ source activate torch
$ conda install -c pytorch -c nvidia pytorch=2.2.0 torchvision torchaudio pytorch-cuda=12.1
```

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

4) Installing software packages

c) Bonus: Hot packages!

i. PyTorch (2.2.0, w/ GPU support)

```
$ conda create -n torch
$ source activate torch
$ conda install -c pytorch -c nvidia pytorch=2.2.0 torchvision torchaudio pytorch-cuda=12.1
```

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

4) Installing software packages

c) Bonus: Hot packages!

ii. Tensorflow (2.15.1, w/ GPU support)

```
$ conda create -n tf
$ source activate tf
$ conda install python=3.11
$ pip install tensorflow[and-cuda]==2.15.1   # Official recommendation is to use pip, so we use a pip that comes
                                             # with Python inside a virtual environment.
```

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

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**

Advanced Tips

A little more than the basics…

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
default / main / anaconda	• Default channel, officially managed by Conda
conda-forge	• Community supported, but rule-enforced and generally trustworthy
bioconda	• Community supported for bioinformatics
nvidia / cuda	• Nvidia official channel
pytorch	• PyTorch official channel
intel	• Intel official channel
…	

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

```
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**!

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**

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

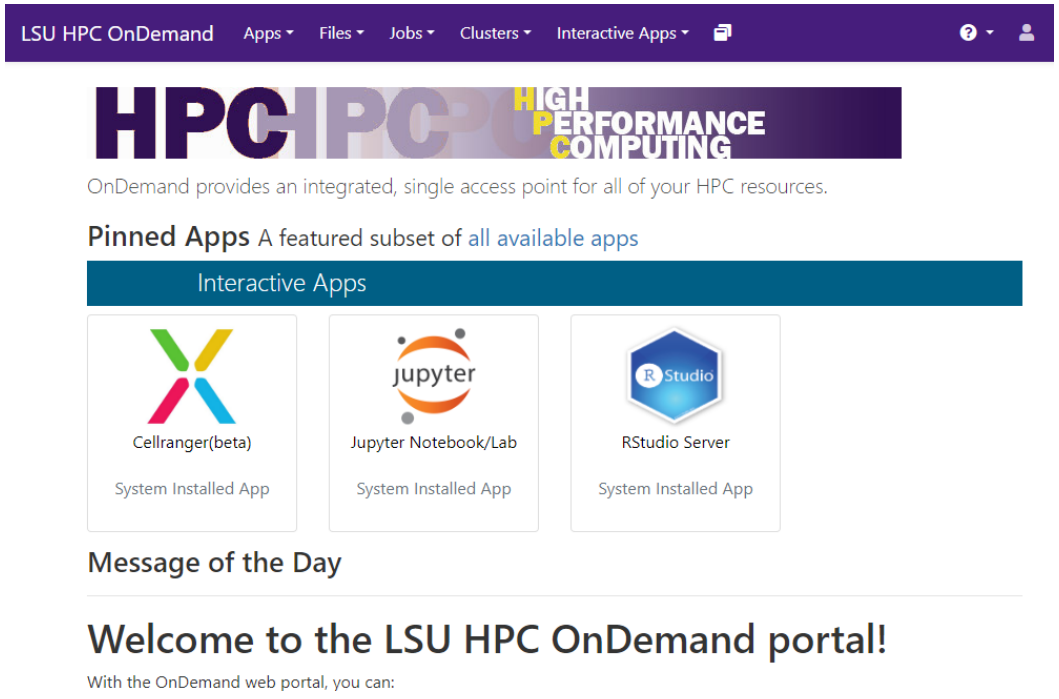
- **Solution**

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`

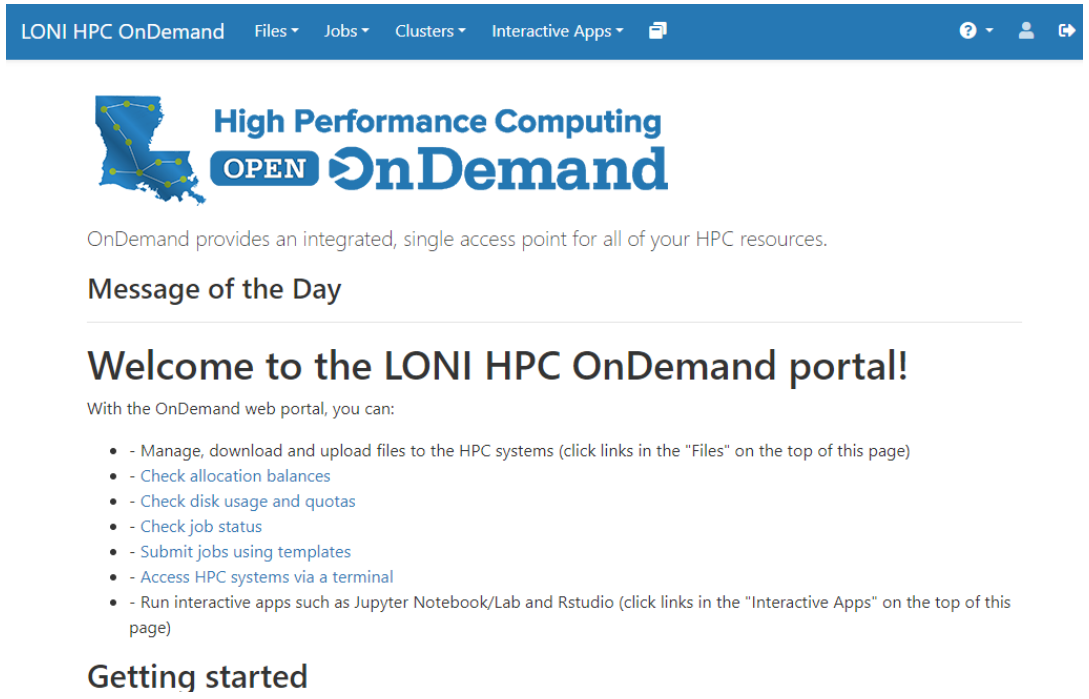
[1] https://conda.io/projects/conda/en/latest/user-guide/tasks/manage-environments.html

5) Use virtual environment in Open OnDemand

LSU HPC (SMIC / SuperMike 3)

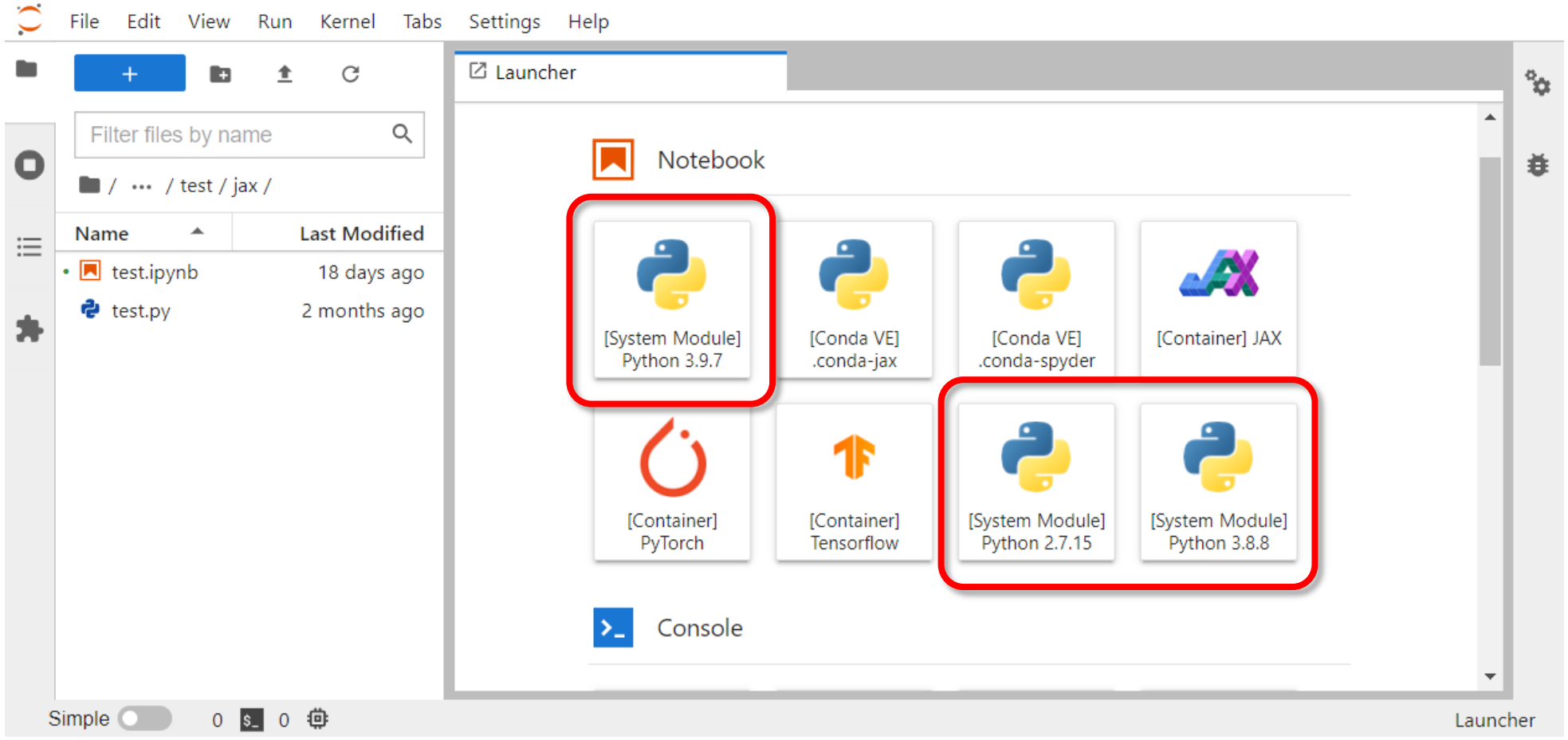


LONI (QB3)



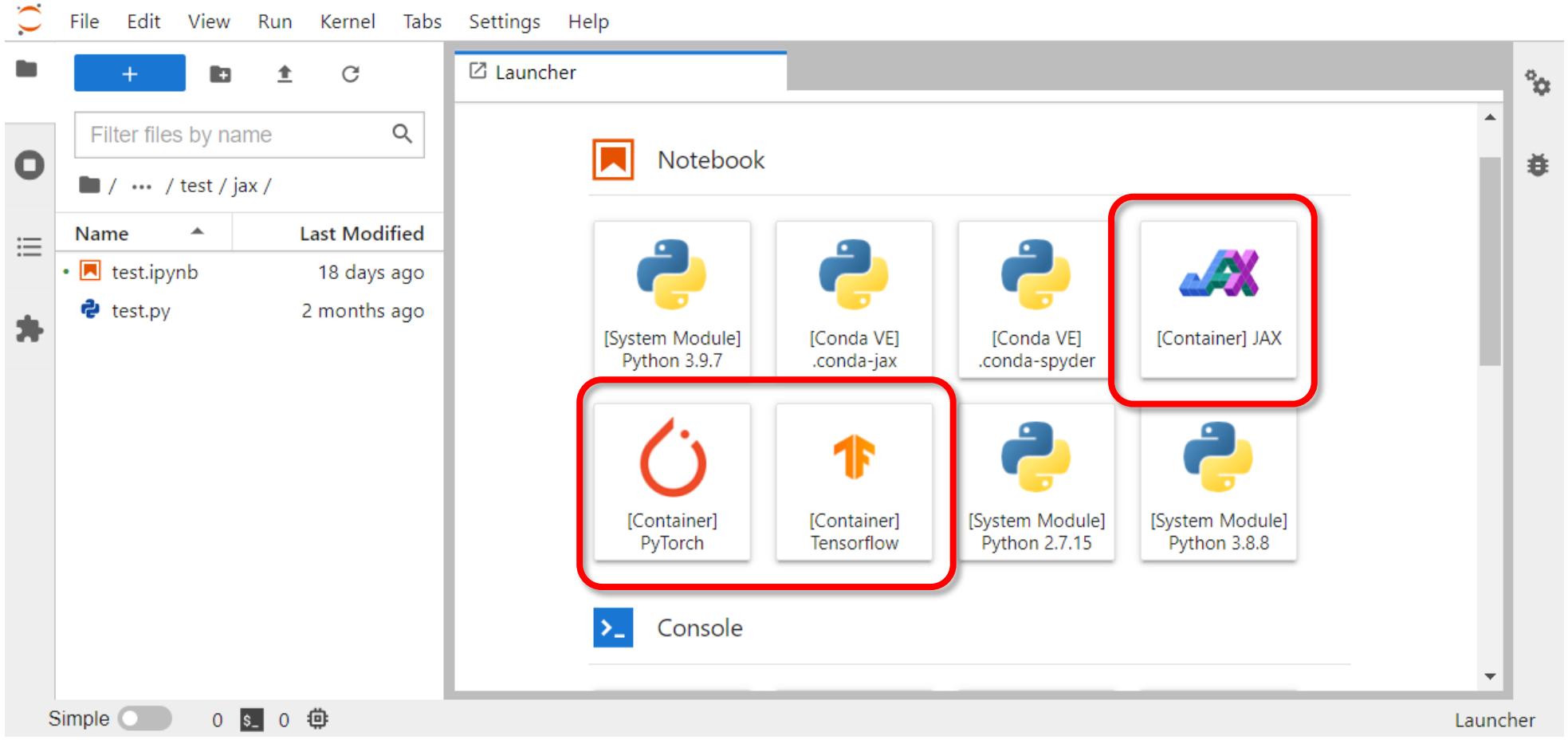
[1] https://youtu.be/JLXN0AZgaqc

5) Use virtual environment in Open OnDemand



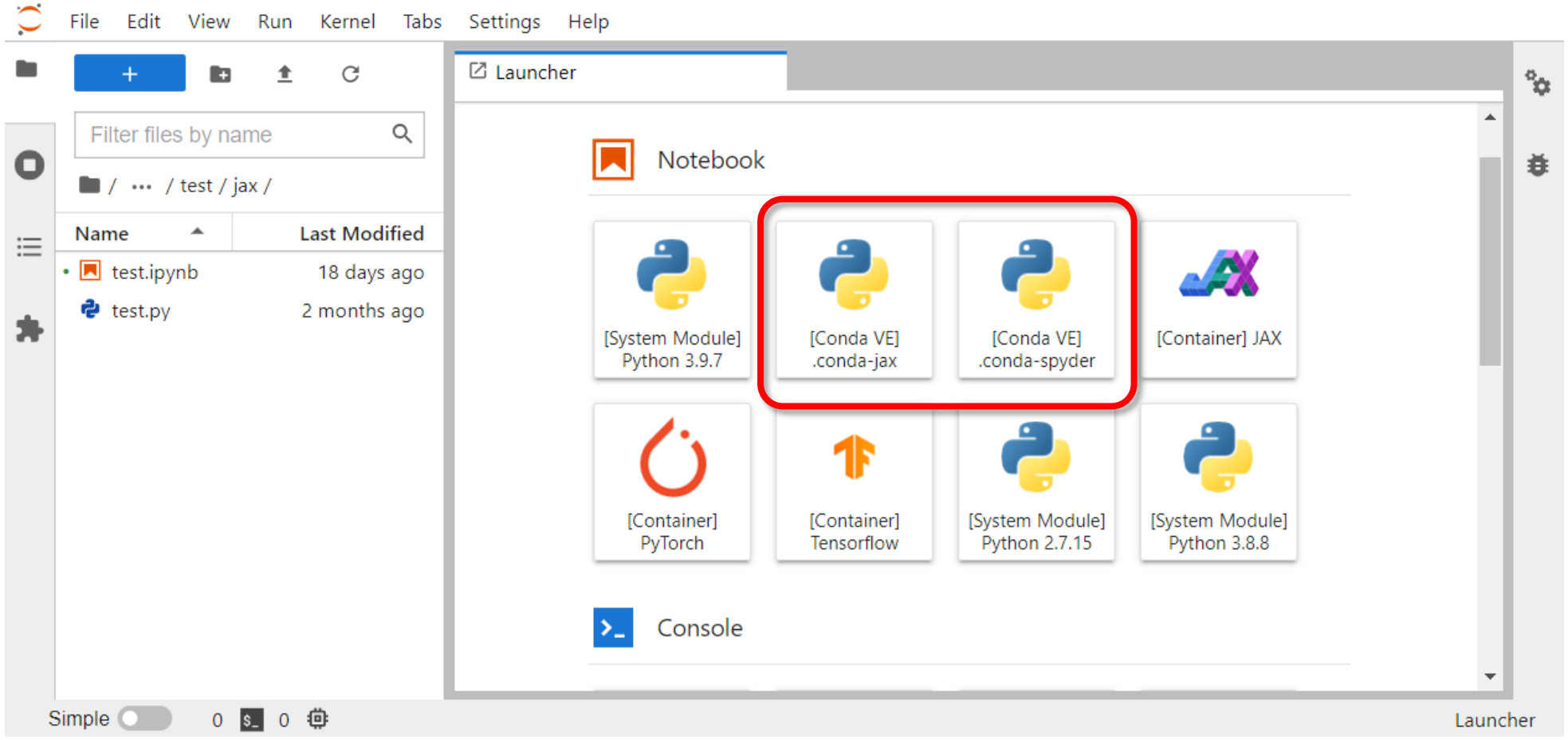
[1] https://youtu.be/JLXN0AZgaqc

5) Use virtual environment in Open OnDemand



[1] https://youtu.be/JLXN0AZgaqc

5) Use virtual environment in Open OnDemand



[1] https://youtu.be/JLXN0AZgaqc

5) Use virtual environment in Open OnDemand

- **How to:**

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

LSU HPC	LONI
SMIC, Super Mike 3	QB2, QB3

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

```
$ source activate ENVIRONMENT
```

Step 3: Install ipykernel

```
$ conda install ipykernel          # Either use this
$ pip install ipkykernel           # Or use this
```

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

[1] https://youtu.be/JLXN0AZgaqc

6) More than Python

- **Scenario**
  - I need software packages other than Python (R / Perl / Lua / …)
  - I need a different version than the system modules
  - I am using the system-wide R module, but failed to installing some packages (e.g., terra)

```
> install.packages("terra")
Warning in install.packages("terra") :
  'lib = "/home/packages/r/4.2.2/gcc-11.2.0/lib64/R/library"' is not
```

```
no
configure: error: gdal-config not found or not executable.
ERROR: configuration failed for package 'terra'
* removing '/home/jasonli3/R/x86_64-pc-linux-gnu-library/4.2/terra'

The downloaded source packages are in
        '/tmp/RtmpxIeKKz/downloaded_packages'
Warning message:
In install.packages("terra") :
  installation of package 'terra' had non-zero exit status
>
```

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 -c conda-forge`
Dependencies	hdf5	`conda install hdf5`
	netcdf	`conda install libnetcdf -c conda-forge`
	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., Use Conda to solve your R issue**

  – Use system's R module:

```
$ module load r
$ R
> install.packages("terra")          → Will fail!
```

  – Use Conda:

```
$ conda create -n terra
$ source activate terra
$ conda install r-terra              → Will succeed!
```

7) Troubleshooting

a) Cannot change default conda

– 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.

➢ 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
```

7) Troubleshooting

b) Conflict with system module

```
(spyder) [jasonli3@smic2 ~]$ module li
Currently Loaded Modulefiles:
 1) python/3.8.5-anaconda
(spyder) [jasonli3@smic2 ~]$ python
Python 3.10.9 (main, Mar  8 2023, 10:47:38) [GCC 11.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import numpy as np
/usr/local/packages/python/3.8.5-anaconda/l       ython3.8/site-packages/numpy/__init__.p
fore Intel(R) MKL initialization ensuring       ct out-of-the box operation under c
ess is not assured. Please install mkl-se              see http://github.com/IntelPyt
  from . import _distributor_init
Traceback (most recent call last):
  File "/usr/local/packages/python/3.8.5-anaconda/lib/      on3.8/site-packages/numpy/co
    from . import multiarray
  File "/usr/local/packages/python/3.8.5-anaconda/lib/python3.8/site-packages/numpy/co
```

Fail!

b) Conflict with system module

- **Rule of thumb**:

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

7) Troubleshooting

c) What if I made a mess?

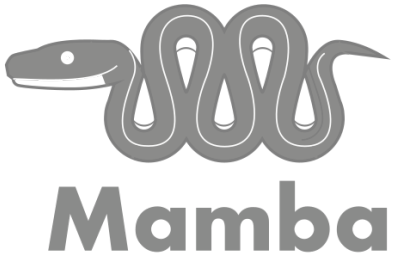
- I mixed conda / pip back and forth, and broke the environment…
- I tried to add a package in my existing environment, but Conda failed at solving the environment…
- I tried to `conda upgrade` a package in my environment, but Conda failed at solving the environment…

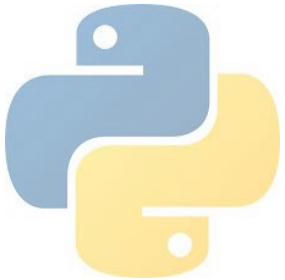
➢ It may be easier to create a new virtual environment and start fresh…

7) Troubleshooting

LSU

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

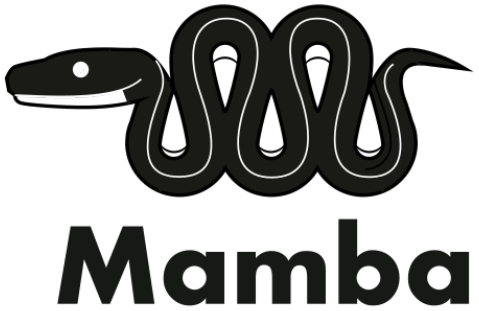




LSU INFORMATION TECHNOLOGY SERVICES

7) Troubleshooting

d) 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
```

7) Troubleshooting

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

"A better Conda"

- **Mamba**:
  - A **drop-in replacement** of Conda
  - Designed to be **faster**, and **resolve Conda failure**

- 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 ~]$
```

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

Conclusion

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

Magic Tools to Install / Manage Software

Part 2: Singularity Container

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

Louisiana State University, Baton Rouge
Mar 20, 2024

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 (with .yml recipe)	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