# Managing Software Packages with CONDA Virtual Environment

**Jason Li**
HPC User Services
LSU HPC / LONI
sys-help@loni.org
Louisiana State University
Baton Rouge
Mar 29, 2023

# Outlines

1. **Why Conda?**
   1) Scenarios
   2) Concepts

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) Scenarios
   2) Concepts

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

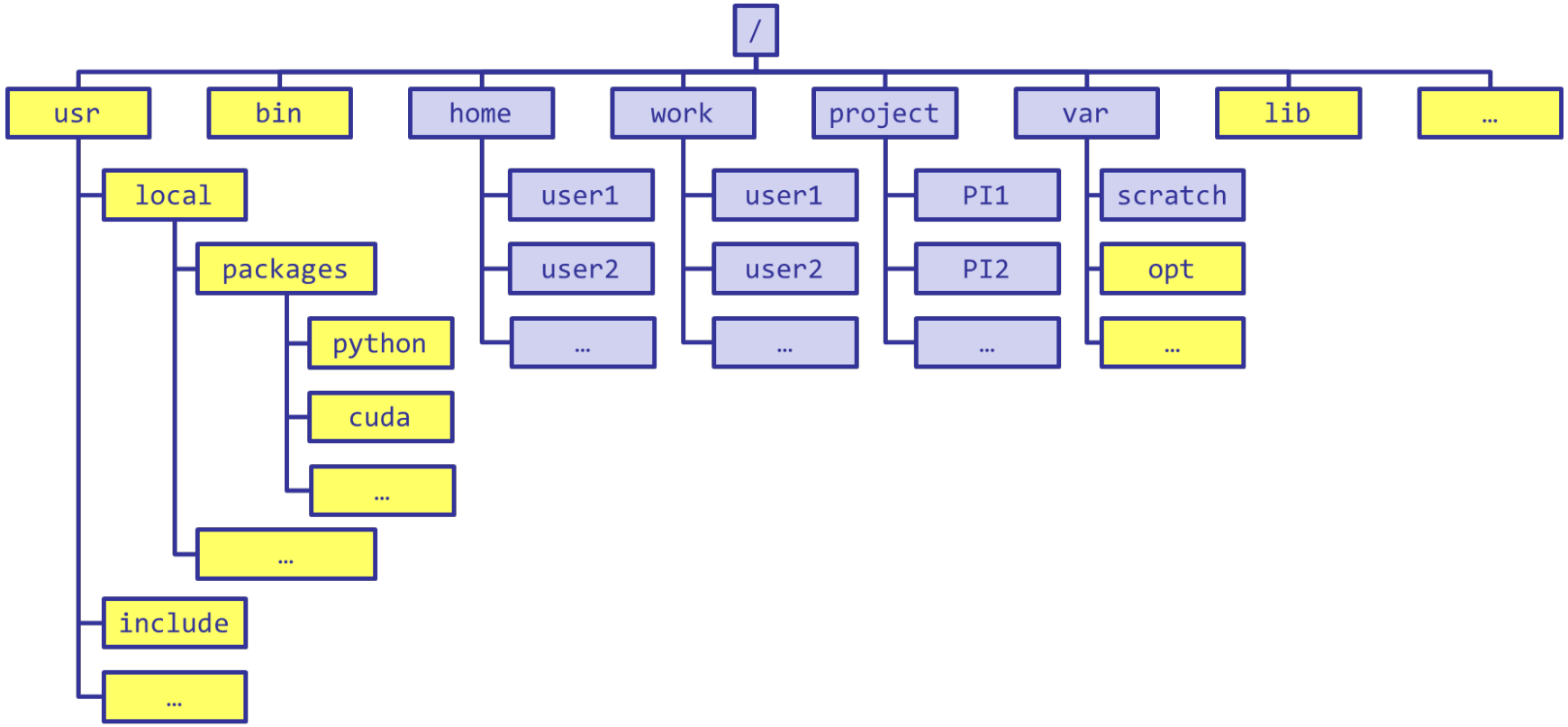
3. **Advanced Tips**

# 1) Scenarios

- **Facing difficulties installing / managing your software packages on HPC?**
- **Some typical scenarios…**

# 1) Scenarios

## a) Permission denied



# 1) Scenarios

## a) Permission denied

```
[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) Scenarios

## a) Permission denied

```
[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) Scenarios

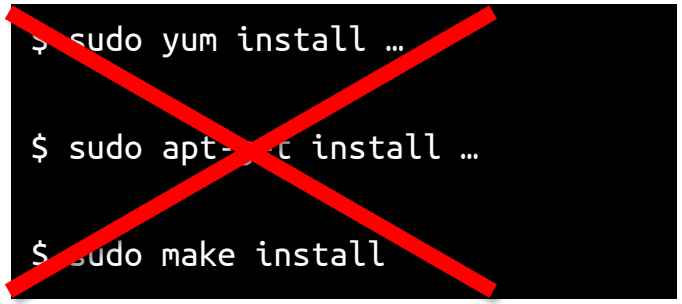
## a) Permission denied

```
[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) Scenarios

## a) Permission denied



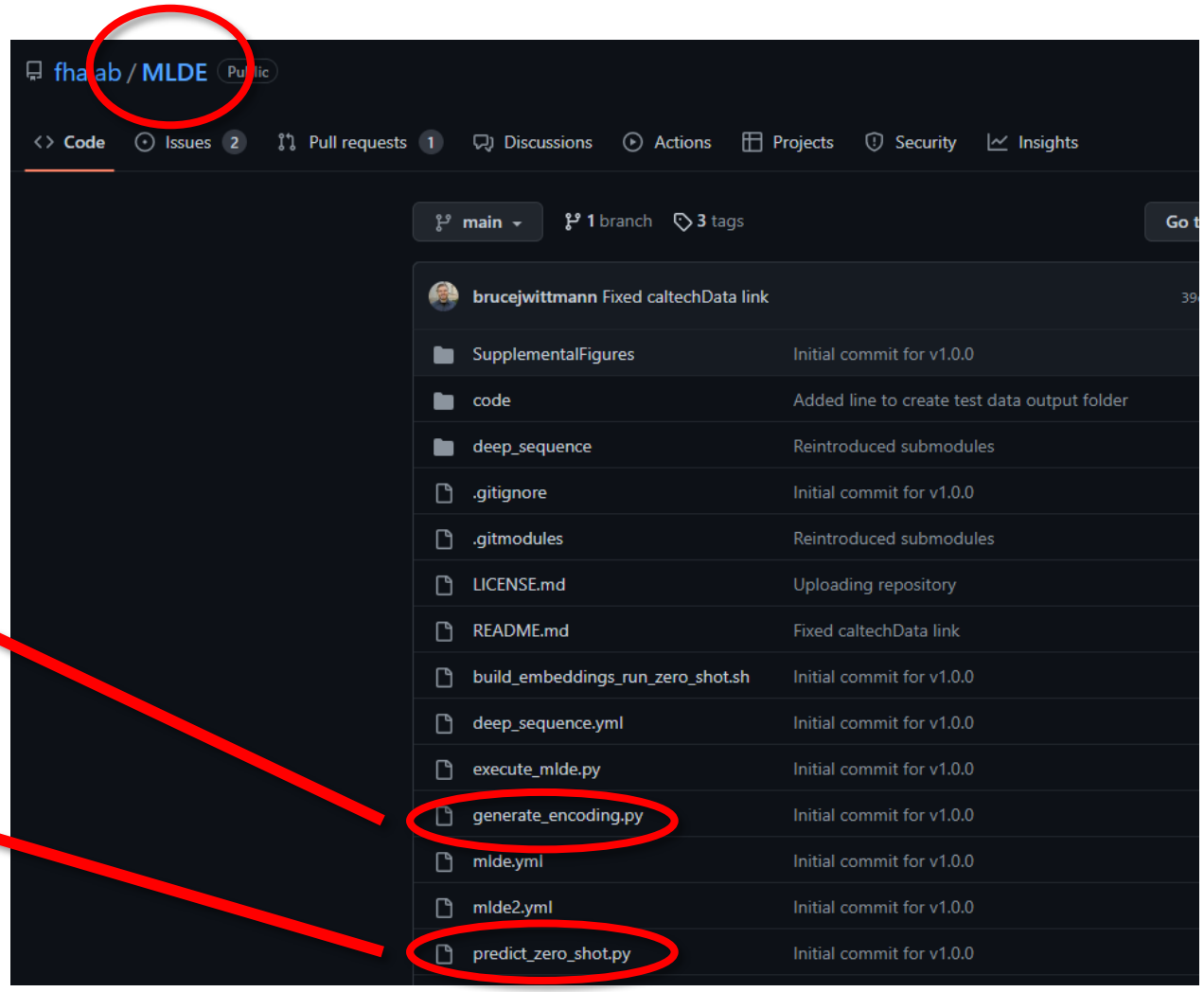
# 1) Scenarios

## b) Conflicted packages

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

Tensorflow 1.13

PyTorch > 1.5



# 1) Scenarios

## c) Need a different version?

```
[jasonli3@smic2 ~]$ module av python/ r/ perl/
------- /usr/local/packages/Modules/default/modulefiles/linux-rhel7-ivybridge --------
python/2.7.7-anaconda    python/2.7.13-anaconda-tensorflow  python/3.8.5-anaconda-ood
python/2.7.7/GCC-4.9.0   python/3.6.2-anaconda-tensorflow
python/2.7.10-mkl-mic    python/3.8.5-anaconda

------- /usr/local/packages/Modules/default/modulefiles/linux-rhel7-ivybridge --------
r/4.0.3/intel-19.0.5  r/4.1.2/gcc-9.3.0

------- /usr/local/packages/Modules/default/modulefiles/linux-rhel7-ivybridge --------
perl/5.32.0/intel-19.0.5
```

# 1) Scenarios

## d) Over complicated dependencies

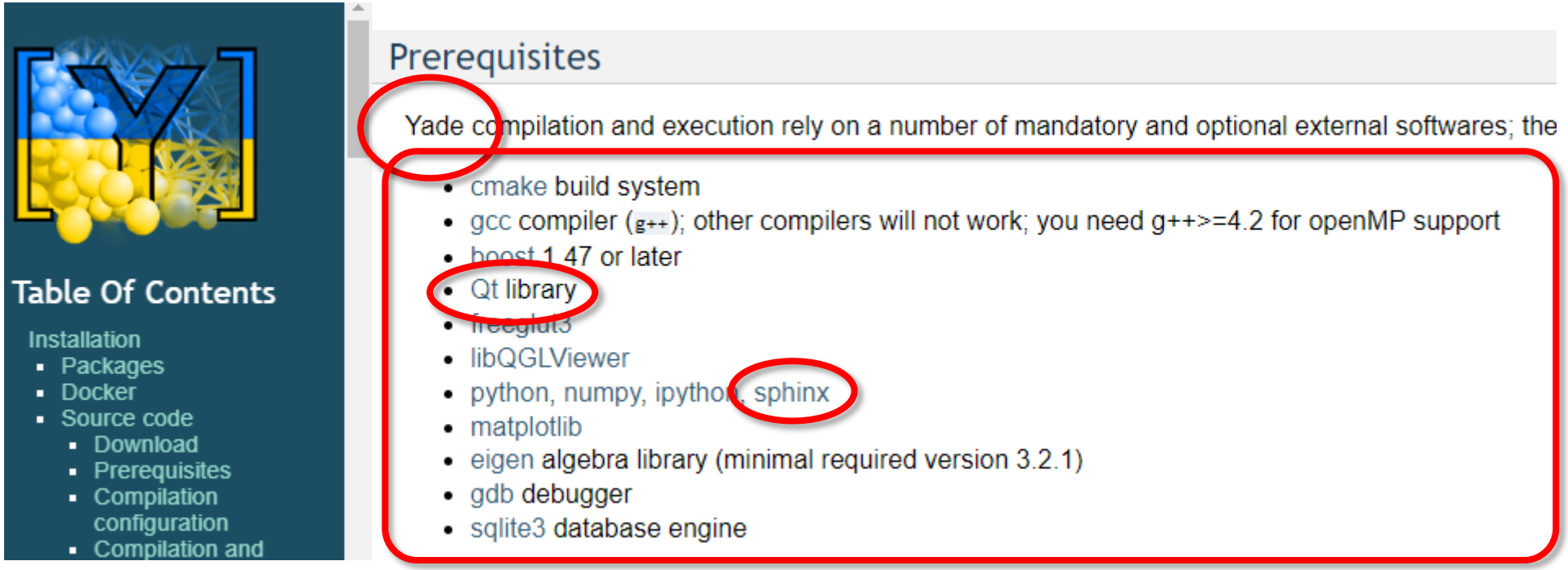


Table Of Contents

Installation
- Packages
- Docker
- Source code
  - Download
  - Prerequisites
  - Compilation configuration
  - Compilation and

Prerequisites

Yade compilation and execution rely on a number of mandatory and optional external softwares; the

- cmake build system
- gcc compiler (g++); other compilers will not work; you need g++>=4.2 for openMP support
- boost 1.47 or later
- Qt library
- freeglut3
- libQGLViewer
- python, numpy, ipython, sphinx
- matplotlib
- eigen algebra library (minimal required version 3.2.1)
- gdb debugger
- sqlite3 database engine

## e) 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) Scenarios

**Any of those apply to you?**

# Outlines

1. **Why Conda?**
   1) Scenarios
   2) Concepts

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

3. **Advanced Tips**

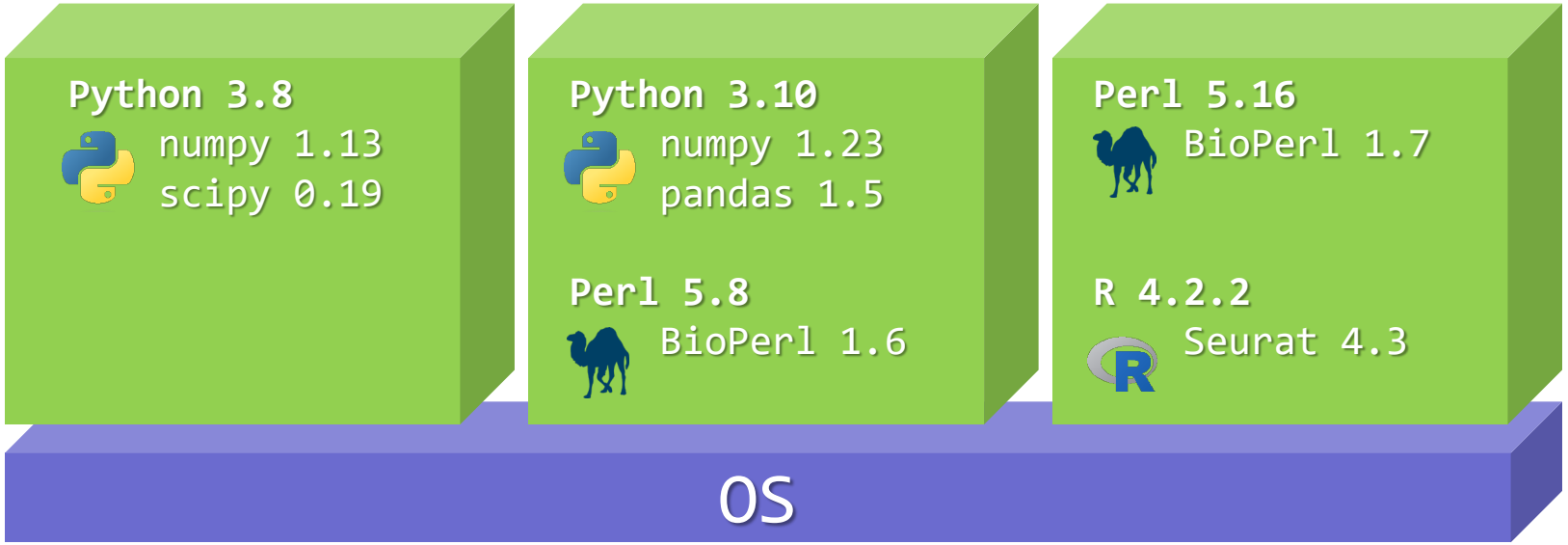

# Virtual Environment

## a) Conda


- A **software**
- Installs / updates / manages packages & dependencies
- Creates / loads / switches between virtual environments
- Initially for Python → General purposes
- Advantage: Does **NOT** need sudo permission!

## b) Virtual Environment (VE)

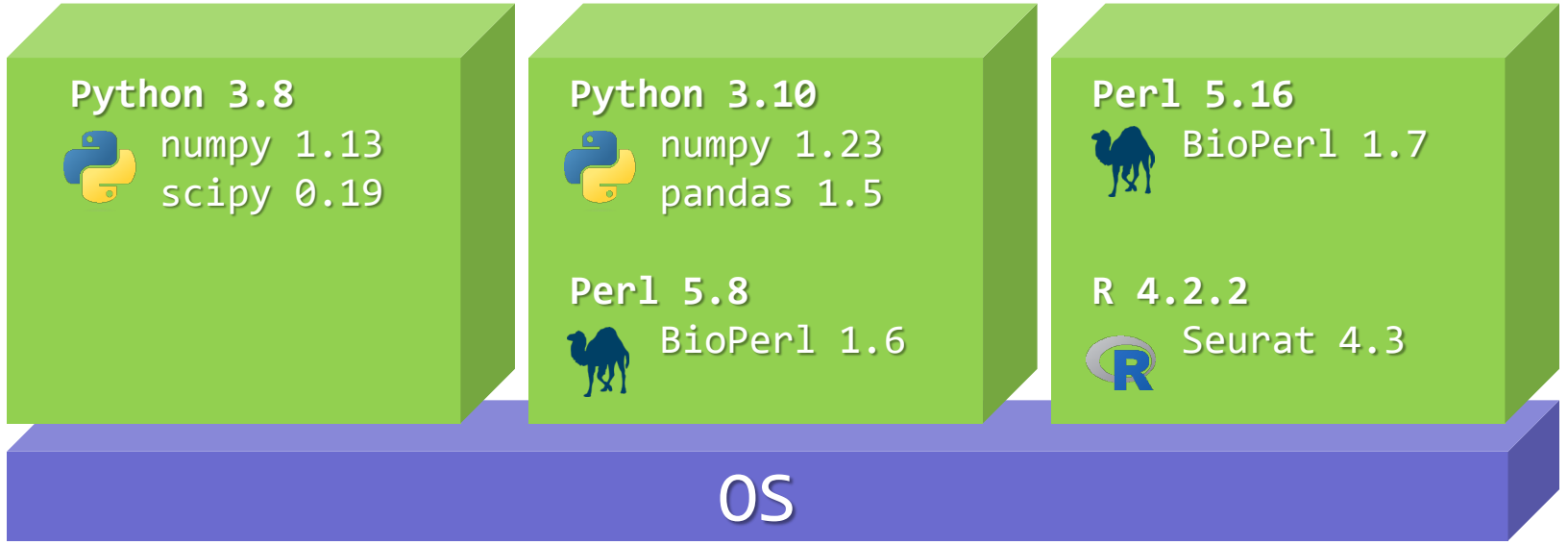
- An **environment**
- **Isolated** and **self-contained** to install and manage packages & dependencies



## b) Virtual Environment (VE)

- An **environment**
- **Isolated** and **self-contained** to install and manage packages & dependencies

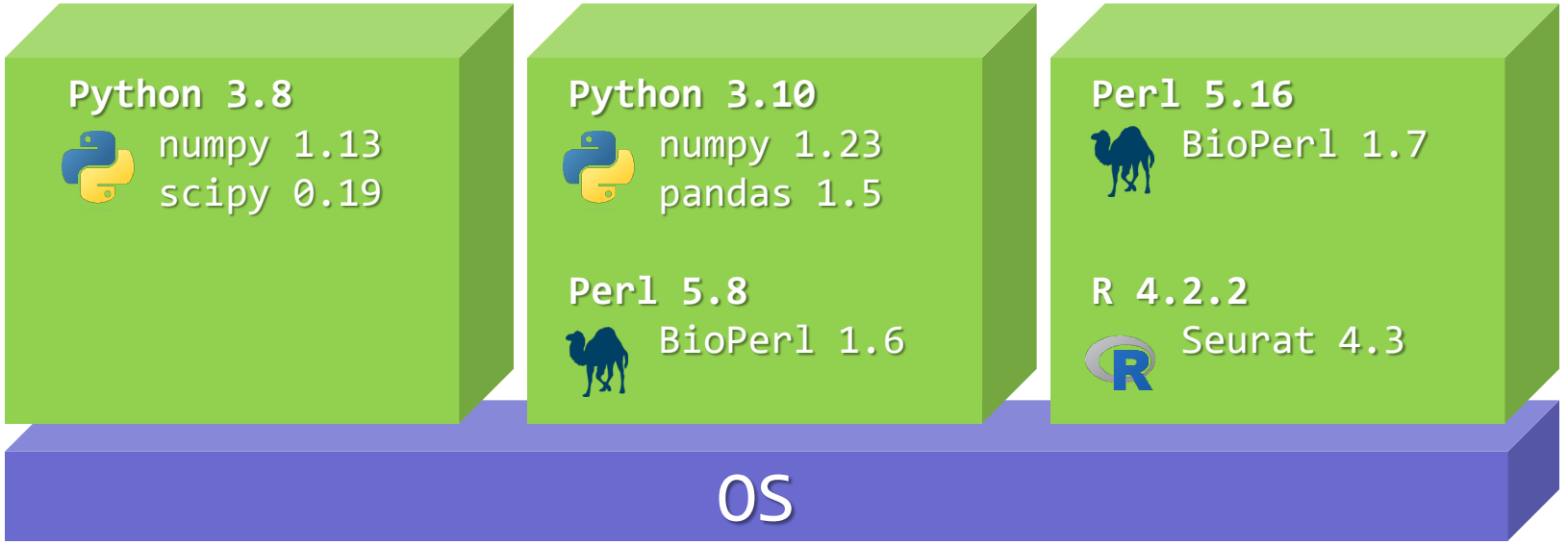
Whatever happens in a VE stays in that VE…



## b) Virtual Environment (VE)

- An **environment**
- **Isolated** and **self-contained** to install and manage packages & dependencies

All dependencies are installed within the VE



- **Relation**


- is **a tool** to create / manage
- is **not the only** tool to create / manage
- **usually** works with

**Virtual Environment**

# Before we continue…

- **Rule of thumb:**

**If a software package you need is managed by Conda,**
**you (most likely) can install / manage it by yourself without a problem**

# Outlines

1. Why Conda?
   1) Scenarios
   2) Concepts

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) Scenarios
   2) Concepts

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

3. Advanced Tips

# 1) Get Conda

- First install a conda in your user environment … ?

# 1) Get Conda

## a) Use conda that comes with system-wide python module

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

```
$ module load python
```

**Step 1: Can use Conda**

```
$ conda init
```

**Step 2: Can use Conda without loading python 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: Allow execution
$ ./Miniconda3-latest-Linux-x86_64.sh
                                                    Step 3: Run and follow prompts
```

# Outlines

1. Why Conda?
   1) Scenarios
   2) Concepts

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

3. Advanced Tips

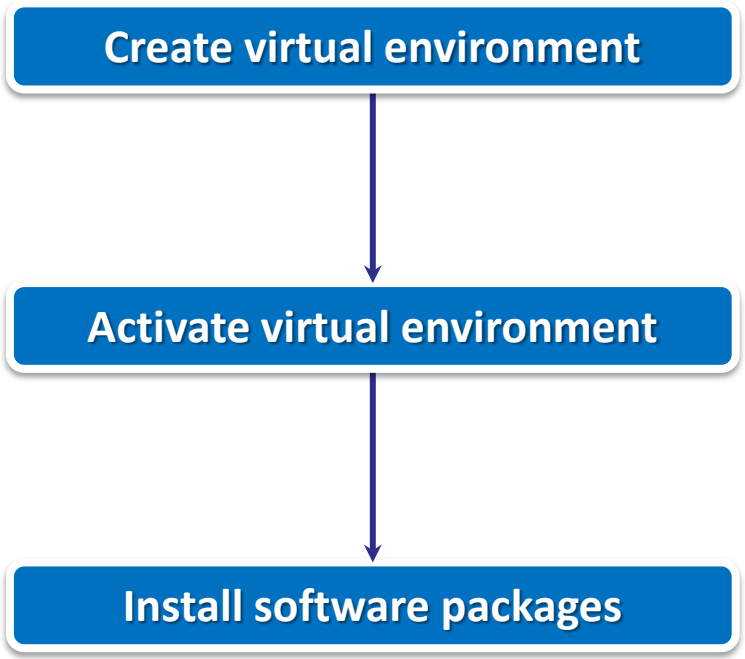
# 2) Typical workflow

- **Key:**

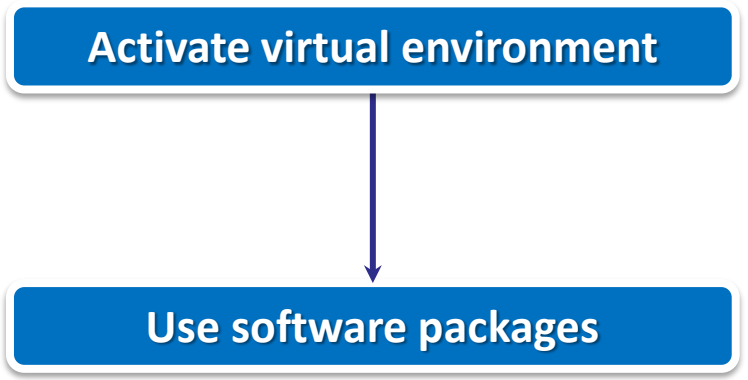
**Always use a virtual environment!**

# 2) Typical workflow

**To install …**




**To use …**



# Outlines

1. Why Conda?
   1) Scenarios
   2) Concepts

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` | Cre |

```
(base) [jasonli3@smic2 ~]$ 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: 23.1.0

Please update conda by running

    $ conda update -n base -c defaults conda



## Package Plan ##

  environment location: /home/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
```

[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 |

```
(myenv) [jasonli3@smic2 ~]$ conda deactivate
(base) [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 |
| `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) Scenarios
   2) Concepts

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

3. Advanced Tips

# 4) Installing software packages

- **Before installation…**

**Make sure a virtual environment is activated!**

# 4) Installing software packages

- **Most frequently used commands**

| Command | Description |
| --- | --- |
| `conda install NAME` | Install a software package |

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

# 4) Installing software packages

- **Most frequently used commands**

| Command | Description |
|---|---|
| `conda install NAME` | Install a software package |
| `conda install NAME=VERSION` | Install a specific version |

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

# 4) Installing software packages

- **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, …) |

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

# 4) Installing software packages

- **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) |

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

# 4) Installing software packages

- **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

- **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

- **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

- **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

- **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

- **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

- **Bonus: Hot packages!**

a) PyTorch (w/ GPU support)

```
$ # Start an interactive job (using qsub or srun)
```

**Step 1: Start an interactive job (otherwise will timeout)**

```
$ conda create -n torch
$ source activate torch
$ conda install -c pytorch -c nvidia pytorch==1.13.1 torchvision==0.14.1 torchaudio==0.13.1 pytorch-cuda=11.6
```

**Step 2: Create a VE and install PyTorch & dependencies**

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

# 4) Installing software packages

- **Bonus: Hot packages!**

b) Tensorflow (w/ GPU support)

```
$ # Start an interactive job on *GPU* nodes (using qsub or srun)
```

**Step 1: Start an interactive job on *GPU* nodes (otherwise will fail)**

```
$ conda create -n tf
$ source activate tf
$ conda install -c conda-forge -c nvidia tensorflow-gpu=2.11 cudatoolkit=11.6 cuda-nvcc=11.6
```

**Step 2: Create a VE and install tensorflow & dependencies**

```
$ mkdir -p $CONDA_PREFIX/etc/conda/activate.d
$ echo 'export XLA_FLAGS=--xla_gpu_cuda_data_dir=$CONDA_PREFIX' >> $CONDA_PREFIX/etc/conda/activate.d/env_vars.sh
```

**Step 3: Run these commands to set up 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) Scenarios
   2) Concepts
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) Change Conda path

- **Default Conda path:**
  - Environments: `~/.conda/envs/`
  - Downloaded packages: `~/.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
```

# 1) Change Conda path

- **Solution:**

**Step 1**: Ask your PI to apply for a storage allocation (/project) and add you to it

**Step 2**: Set up Conda to create / find virtual environments in /project

**Step 3**: Create your virtual environment and install software package

# 1) Change Conda path

- **Solution:**

~~**Step 1**: Ask your PI to apply for a storage allocation (/project) and add you to it~~

**Step 2**: Set up Conda to create / find virtual environments in /project

~~**Step 3**: Create your virtual environment and install software package~~

# 1) Change Conda path

## a) Method 1: Command lines

```
$ conda config --add envs_dirs /path/to/envs
```

Add path to environments

```
$ conda config --add pkgs_dirs /path/to/pkgs
```

Add path to downloaded packages

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

# 1) Change Conda path

## b) Method 2: Configuration file

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

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

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

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

# 2) 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?

# 2) Share virtual environment

- **Solution:**

**Step 1**: Ask your PI to apply for a storage allocation (/project) and add **you** and **your colleagues** to it

**Step 2**: Set up Conda to create / find virtual environments in /project

**Step 3**: Create your virtual environment and install software packages

**Step 4: Ask your colleague to repeat [Step 2]**

# 3) 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 software. How can I do that?

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

# 3) Migrate / clone virtual environment

- **Solution**

| To … | Run command |
| --- | --- |

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

# 3) 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

# 3) Migrate / clone virtual environment

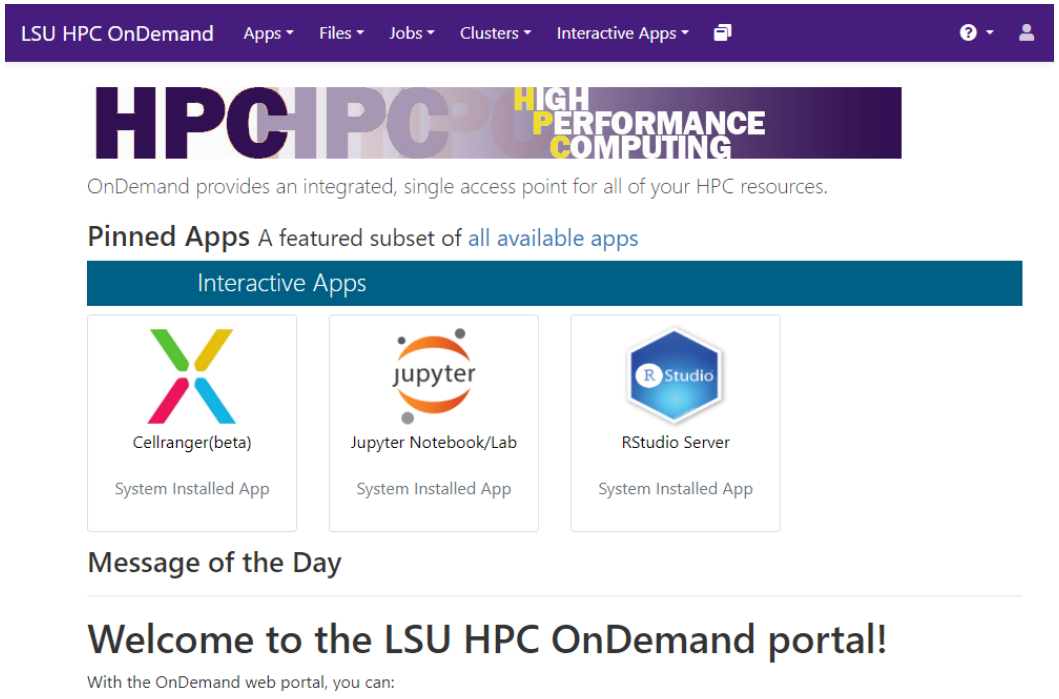
- **Solution**

| 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` |

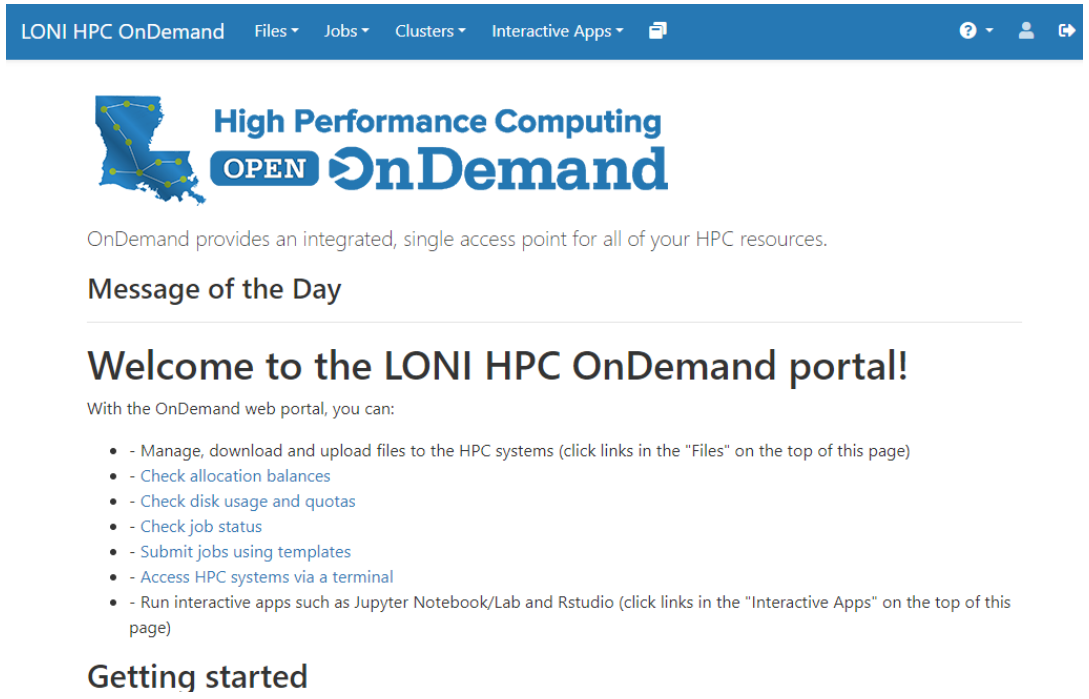
[1] https://conda.io/projects/conda/en/latest/user-guide/tasks/manage-environments.html

# 4) Use virtual environment on Open OnDemand

## LSU HPC (SMIC)

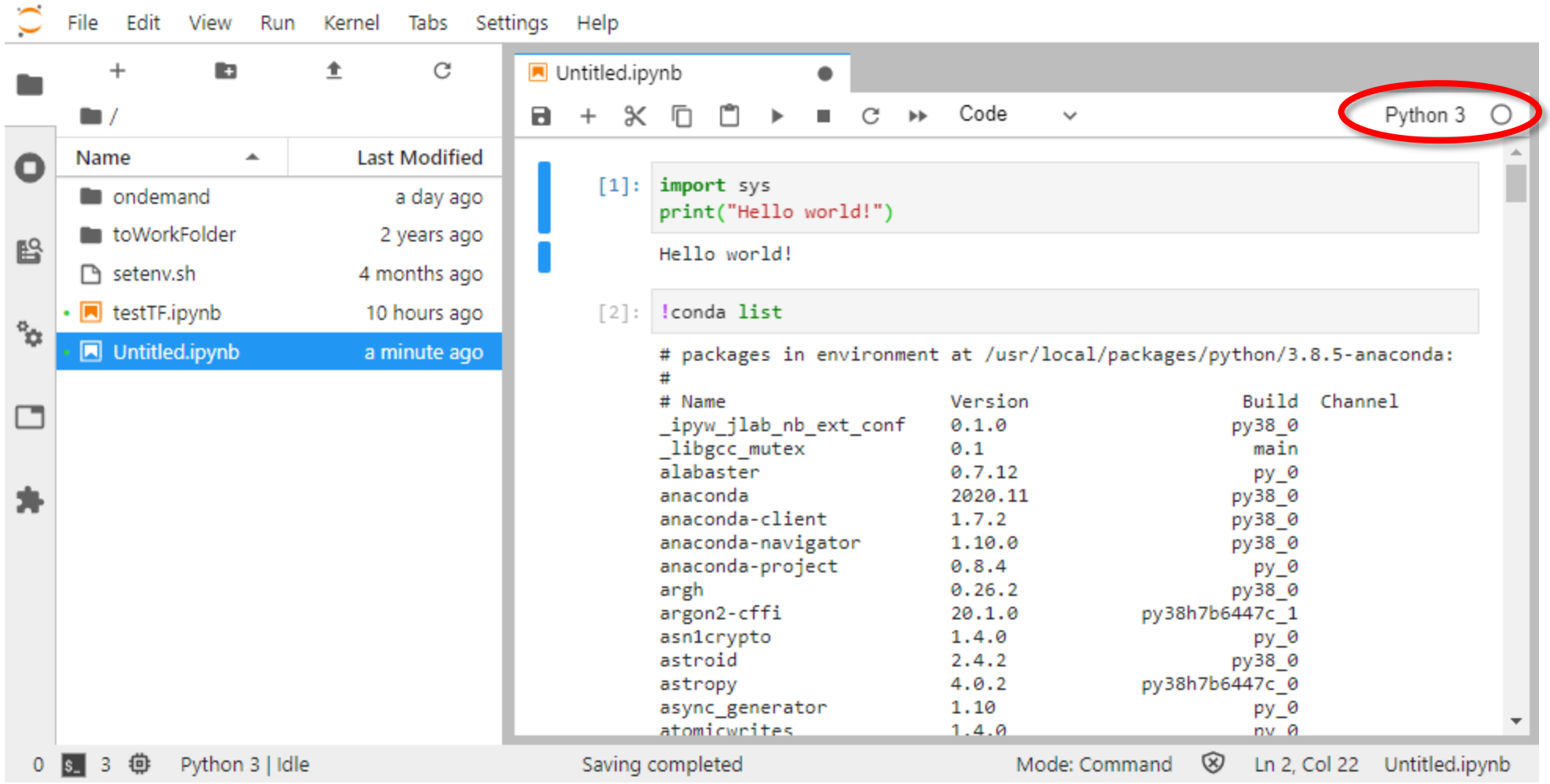



## LONI (QB2)




[1] https://youtu.be/xk5q8p6QQ_k

# 4) Use virtual environment on Open OnDemand




[1] https://youtu.be/xk5q8p6QQ_k

# 4) Use virtual environment on Open OnDemand

- **How to:**

**Step 1**: `ssh` to the cluster you want to use

| LSU HPC | LONI |
|---|---|
| SMIC | QB2 |

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

```
$ source activate ENVIRONMENT
```

**Step 3**: Install ipykernel

```
$ conda install ipykernel
```

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

[1] https://youtu.be/xk5q8p6QQ_k

# 5) 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's R module, but having trouble installing some packages (e.g., rgdal)

```
> install.packages("rgdal")
Warning in install.packages("rgdal") :
  'lib = "/home/packages/r/4.1.2/5k5jengl/rlib/R/library"' is not writable
```

```
configure: error: gdal-config not found or not executable.
ERROR: configuration failed for package 'rgdal'
* removing '/home/jasonli3/R/x86_64-pc-linux-gnu-library/4.1/rgdal'

The downloaded source packages are in
        '/tmp/Rtmpd2csho/downloaded_packages'
Warning message:
In install.packages("rgdal") :
  installation of package 'rgdal' had non-zero exit status
```

# 5) 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` |

[1] https://anaconda.org/

# 5) 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/

# 5) 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/

# 5) More than Python

- **One more cool thing…**
  - 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

# 5) More than Python

- **E.g., Use Conda to solve your R issue**

  – Use system's R module:

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

  – Use Conda:

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

# 6) Dependency resolution

- **Scenario**
  - I need to compile a code from source, but the dependencies are too convoluted

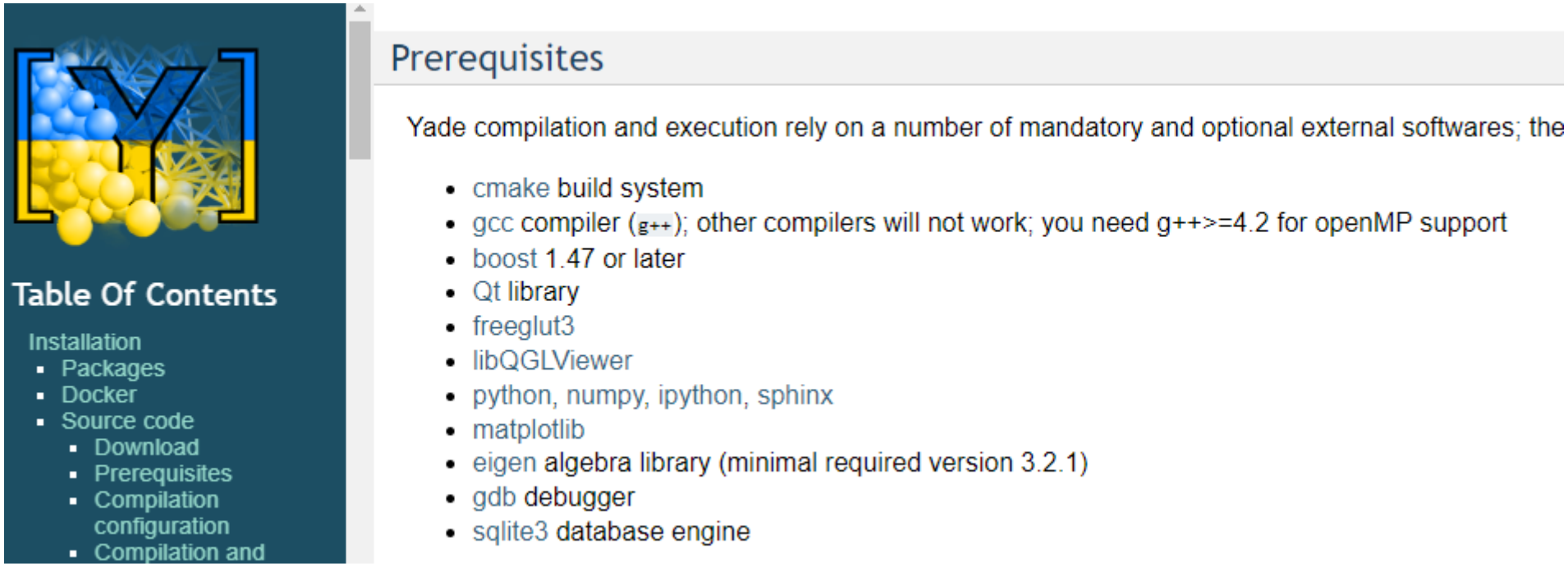


Table Of Contents

Installation
- Packages
- Docker
- Source code
  - Download
  - Prerequisites
  - Compilation configuration
  - Compilation and

## Prerequisites

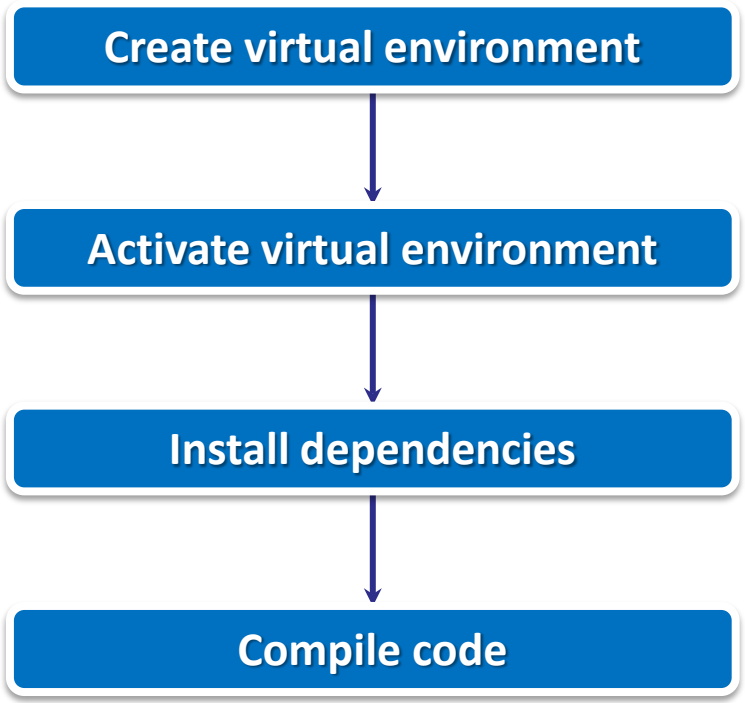
Yade compilation and execution rely on a number of mandatory and optional external softwares; the

- cmake build system
- gcc compiler (`g++`); other compilers will not work; you need g++>=4.2 for openMP support
- boost 1.47 or later
- Qt library
- freeglut3
- libQGLViewer
- python, numpy, ipython, sphinx
- matplotlib
- eigen algebra library (minimal required version 3.2.1)
- gdb debugger
- sqlite3 database engine

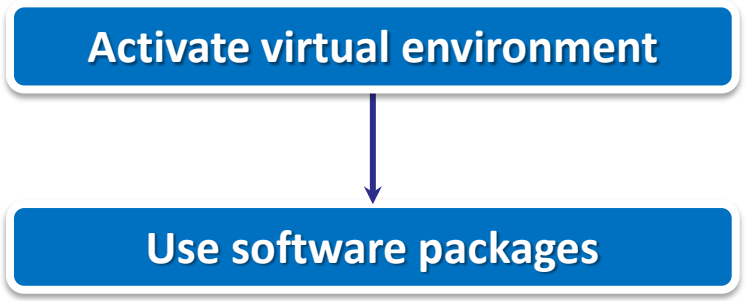
# 6) Dependency resolution

- **Solution**

**To install …**




**To use …**



# 7) Troubleshooting

## a) 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!

# 7) Troubleshooting

## a) Conflict with system module

– **Rule of thumb**:

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

# 7) Troubleshooting

## b) What if I made a mess?

– I mixed conda / pip back and forth, and broke the environment…

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

# Take home message

**To install …**

```
$ conda create …

$ source activate …

$ conda install …
```

**To use …**

```
$ source activate …

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

# Contact us

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