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
Oct 25, 2023
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. 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) Problems

- Core problem:

Installing software on an HPC system
1) Problems

- Traditional Linux solution:
  - Compiling from source code
a) Dependencies (Welcome to Linux!)

BUSCO v5.4.7 is the current stable version!
Gitlab, a Conda package and Docker container are also available.

Based on evolutionarily-informed expectations of gene content of near-universal single-copy orthologs, BUSCO metric is complementary to technical metrics like N50.
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, IBBMap, tBLASTx 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/
- http://bioinf.uni-greifswald.de/augustus/
- https://github.com/soedinglab/metaeuk
- https://github.com/hyaatpd/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.
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, IBBMap, tblASTn 2.2+, Augustus 3.2+, Prodigal, Metaeku, 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://biopython.org/)
- [https://pandas.pydata.org/](https://pandas.pydata.org/)
- [http://bioinf.uni-greifswald.de/augustus/](http://bioinf.uni-greifswald.de/augustus/)
- [https://github.com/soedinglab/metaektu](https://github.com/soedinglab/metaektu)
- [https://github.com/hyattpd/Prodigal](https://github.com/hyattpd/Prodigal)
- [http://hmmer.org/](http://hmmer.org/)
- [https://github.com/smirarab/sepp](https://github.com/smirarab/sepp)
- [https://www.r-project.org/](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+, Augostus 3.2+, Prodigal, MetaEuK, HMMER3.1+, SEPP, and R + ggplot 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/
- http://bioinf.uni-greifswald.de/augustus/

Please make sure that each software package listed above works INDEPENDENTLY of BUSCO before attempting to run any BUSCO assessments.
b) **Permission denied** *(Welcome to HPC!)*
b) **Permission denied** (Welcome to HPC!)

```
$ module load python/3.6.2-anaconda-tensorflow
$ module list
Currently Loaded Modulefiles:
  1) python/3.6.2-anaconda-tensorflow
```
b) **Permission denied** (Welcome to HPC!)

```
[jasonl3@smic2 ~]$ module load python/3.6.2-anaconda-tensorflow
[jasonl3@smic2 ~]$ module list
Currently Loaded Modulefiles:
  1) python/3.6.2-anaconda-tensorflow
[jasonl3@smic2 ~]$ pip install geos
Collecting geos
  Downloading https://files.pythonhosted.org/packages/49/5b/b8acfb74c01187a36aa41b6523de69baa9e59c
  100% |████████████████████████████████| 409KB 3.0MB/s
```
b) **Permission denied** (Welcome to HPC!)

```bash
[jasonl3@smic2 ~]$ module load python/3.6.2-anaconda-tensorflow
[jasonl3@smic2 ~]$ module list
Currently Loaded Modules:
  1) python/3.6.2-anaconda-tensorflow
[jasonl3@smic2 ~]$ pip install geos
Collecting geos
  Downloading https://files.pythonhosted.org/packages/49/5b/b8aef74c01187a36aa41b6523deb9bada59c50f7395e53b1246cb2680/409KB 3.0MB/s

File "'/usr/local/packages/python/3.6.2-anaconda/lib/python3.6/os.py", line 220, in makedirs
  os.makedirs(path)
File "'/usr/local/packages/python/3.6.2-anaconda/lib/python3.6/os.py", line 220, in makedirs
  mkdir(name, mode)
You are using pip version 9.0.1, however version 20.0.1 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.
[jasonl3@smic2 ~]$ 
```
b) **Permission denied** (Welcome to HPC!)

- If you ask Google / ChatGPT…

```
$ sudo yum install ...
$ sudo apt-get install ...
$ sudo make install
```


c) Conflicted packages

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

Tensorflow 1.13

PyTorch > 1.5
1) Problems

   1. Problems

   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?
Any of those apply to you?
1) Problems

Magic Tools to Install / Manage Software

Part 1: **CONDA** Virtual Environment

Part 2: **ingularity** Container
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) Virtual environment & Conda

Virtual Environment
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)
a) What is a virtual environment?

• Properties

  - Contained
    
    All dependencies are installed within the VE

  - Isolated
    
    Whatever happens in a VE stays in that VE…

```
> /path/to/envs

env1
  • Python 3.7
  • numpy 1.13
  • scipy 0.19

env2
  • Python 3.9
  • pandas 1.5

env3
  • Perl 5.8
  • Bioperl 1.6
  • R 4.2.3
  • Seurat 4.3
```
2) Virtual environment & Conda

b) What is Conda

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

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

Virtual Environment
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**
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. 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

- 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.

Step 1: Can use Conda

```
$ module load python

Step 2: Can use Conda without loading python module (recommended)
```

$ conda init
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
$ chmod u+x Miniconda3-latest-Linux-x86_64.sh
$ ./Miniconda3-latest-Linux-x86_64.sh
```

- **Step 1:** Download miniconda
- **Step 2:** Allow execution
- **Step 3:** Run and follow prompts
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 ...

1. Create virtual environment
2. Activate virtual environment
3. Install software packages

To use ...

1. Activate virtual environment
2. Use software packages
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

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>conda create -n ENVIRONMENT</td>
<td>Create a virtual environment</td>
</tr>
</tbody>
</table>

```
(base) [jasonli@smile ~]$ 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/jasonli/.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
```

3) Creating a virtual environment

• Most frequently used commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>conda create -n ENVIRONMENT</td>
<td>Create a virtual environment</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>source activate ENVIRONMENT</td>
<td>Activate a virtual environment</td>
</tr>
</tbody>
</table>

3) Creating a virtual environment

- Most frequently used commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>conda create -n ENVIRONMENT</code></td>
<td>Create a virtual environment</td>
</tr>
<tr>
<td><code>source activate ENVIRONMENT</code></td>
<td>Activate a virtual environment</td>
</tr>
<tr>
<td><code>conda deactivate</code></td>
<td>Deactivate a virtual environment</td>
</tr>
</tbody>
</table>

3) Creating a virtual environment

- Most frequently used commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>conda create -n ENVIRONMENT</td>
<td>Create a virtual environment</td>
</tr>
<tr>
<td><code>source</code> activate ENVIRONMENT</td>
<td>Activate a virtual environment</td>
</tr>
<tr>
<td>conda deactivate</td>
<td>Deactivate a virtual environment</td>
</tr>
<tr>
<td>conda env list</td>
<td>List all virtual environments</td>
</tr>
</tbody>
</table>

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

3) Creating a virtual environment

- Most frequently used commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>conda create -n ENVIRONMENT</td>
<td>Create a virtual environment</td>
</tr>
<tr>
<td><strong>source</strong> activate ENVIRONMENT</td>
<td>Activate a virtual environment</td>
</tr>
<tr>
<td>conda deactivate</td>
<td>Deactivate a virtual environment</td>
</tr>
<tr>
<td>conda env list</td>
<td>List all virtual environments</td>
</tr>
<tr>
<td>conda env remove -n ENVIRONMENT</td>
<td>Remove a virtual environment and all installed packages</td>
</tr>
</tbody>
</table>

**CAUTION! NO CONFIRMATION! IRREVOCABLE!**

### 3) Creating a virtual environment

- **Most frequently used commands**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>conda create -n ENVIRONMENT</code></td>
<td>Create a virtual environment</td>
</tr>
<tr>
<td><code>source activate ENVIRONMENT</code></td>
<td>Activate a virtual environment</td>
</tr>
<tr>
<td><code>conda deactivate</code></td>
<td>Deactivate a virtual environment</td>
</tr>
<tr>
<td><code>conda env list</code></td>
<td>List all virtual environments</td>
</tr>
<tr>
<td><code>conda env remove -n ENVIRONMENT</code></td>
<td>Remove a virtual environment and all installed packages</td>
</tr>
</tbody>
</table>

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

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

a) Most frequently used commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>conda install NAME</td>
<td>Install a software package</td>
</tr>
</tbody>
</table>

## 4) Installing software packages

### a) Most frequently used commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>conda install NAME</td>
<td>Install a software package</td>
</tr>
<tr>
<td>conda install NAME=VERSION</td>
<td>Install a specific version</td>
</tr>
</tbody>
</table>

4) Installing software packages

a) Most frequently used commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>conda install NAME</td>
<td>Install a software package</td>
</tr>
<tr>
<td>conda install NAME=VERSION</td>
<td>Install a specific version</td>
</tr>
<tr>
<td>conda install NAME -c CHANNEL</td>
<td>Install from a specific channel (e.g., conda-forge, bioconda, nvidia, …)</td>
</tr>
</tbody>
</table>

## 4) Installing software packages

### a) Most frequently used commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>conda install NAME</td>
<td>Install a software package</td>
</tr>
<tr>
<td>conda install NAME=VERSION</td>
<td>Install a specific version</td>
</tr>
<tr>
<td>conda install NAME -c CHANNEL</td>
<td>Install from a specific channel (e.g., conda-forge, bioconda, nvidia, …)</td>
</tr>
<tr>
<td>conda install NAME1 NAME2 …</td>
<td>Install multiple packages at once (let conda work out dependencies)</td>
</tr>
</tbody>
</table>

### 4) Installing software packages

#### a) Most frequently used commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>conda install NAME</code></td>
<td>Install a software package</td>
</tr>
<tr>
<td><code>conda install NAME=VERSION</code></td>
<td>Install a specific version</td>
</tr>
<tr>
<td><code>conda install NAME -c CHANNEL</code></td>
<td>Install from a specific channel (e.g., conda-forge, bioconda, nvidia, …)</td>
</tr>
<tr>
<td><code>conda install NAME1 NAME2 ...</code></td>
<td>Install multiple packages at once (let conda work out dependencies)</td>
</tr>
<tr>
<td><code>conda list</code></td>
<td>List all installed software package</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>conda search NAME</td>
<td>Search available package versions</td>
</tr>
</tbody>
</table>

4) Installing software packages

b) Other useful commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>conda search NAME</td>
<td>Search available package versions</td>
</tr>
<tr>
<td>conda search NAME -c CHANNEL</td>
<td>Search available package versions in a specific channel</td>
</tr>
</tbody>
</table>

### 4) Installing software packages

#### b) Other useful commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>conda search NAME</td>
<td>Search available package versions</td>
</tr>
<tr>
<td>conda search NAME -c CHANNEL</td>
<td>Search available package versions in a specific channel</td>
</tr>
<tr>
<td>conda search NAME --info</td>
<td>Search available package versions with details</td>
</tr>
</tbody>
</table>

### b) Other useful commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>conda search NAME</td>
<td>Search available package versions</td>
</tr>
<tr>
<td>conda search NAME -c CHANNEL</td>
<td>Search available package versions in a specific channel</td>
</tr>
<tr>
<td>conda search NAME --info</td>
<td>Search available package versions with details</td>
</tr>
<tr>
<td>conda update/upgrade NAME</td>
<td>Update a package to the latest available version</td>
</tr>
</tbody>
</table>

### 4) Installing software packages

#### b) Other useful commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>conda search NAME</td>
<td>Search available package versions</td>
</tr>
<tr>
<td>conda search NAME -c CHANNEL</td>
<td>Search available package versions in a specific channel</td>
</tr>
<tr>
<td>conda search NAME --info</td>
<td>Search available package versions with details</td>
</tr>
<tr>
<td>conda update/upgrade NAME</td>
<td>Update a package to the latest available version</td>
</tr>
<tr>
<td>conda uninstall/remove NAME</td>
<td>Remove a package</td>
</tr>
</tbody>
</table>

4) Installing software packages

c) Bonus: Hot packages!

i. PyTorch (w/ GPU support)

---

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

```bash
$ # Start an interactive job (using qsub or srun)
$ conda create -n torch
$ source activate torch
$ conda install pytorch==2.0.1 torchvision==0.15.2 torchaudio==2.0.2 pytorch-cuda=11.8 -c pytorch -c nvidia
```

Step 2: Create a VE and install PyTorch & dependencies

---

c) Bonus: Hot packages!

ii. 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=2.12 cudatoolkit=11.8 cuda-nvcc=11.8

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
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
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 images 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

<table>
<thead>
<tr>
<th>Name</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>default / anaconda</td>
<td>• Default channel, officially managed by Conda</td>
</tr>
<tr>
<td>conda-forge</td>
<td>• Community supported, but rule-enforced and generally trustworthy</td>
</tr>
<tr>
<td>bioconda</td>
<td>• Community supported for bioinformatics</td>
</tr>
<tr>
<td>nvidia / cuda</td>
<td>• Nvidia official channel</td>
</tr>
<tr>
<td>pytorch</td>
<td>• Pytorch official channel</td>
</tr>
<tr>
<td>intel</td>
<td>• Intel official channel</td>
</tr>
</tbody>
</table>

...
2) Change Conda path

- Default Conda path

<table>
<thead>
<tr>
<th></th>
<th>System-wide Python modules</th>
<th>Customized Conda</th>
</tr>
</thead>
<tbody>
<tr>
<td>Environments</td>
<td>/home/$USER/.conda/envs/</td>
<td>/path/to/conda/envs/</td>
</tr>
<tr>
<td>Cache</td>
<td>/work/$USER/.conda/pkgs/</td>
<td>/path/to/conda/pkgs/</td>
</tr>
</tbody>
</table>

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

Storage allocation   MB used    MB quota
2) Change Conda path

a) Method 1: Command lines

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

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

Add path to environments
Add path to downloaded packages

2) Change Conda path

b) Method 2: Configuration file

- Use any text editor to open: ~/.condarc

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

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

2) Change Conda path

c) Places to store your virtual environments:

<table>
<thead>
<tr>
<th>Location</th>
<th>Pros</th>
<th>Cons</th>
</tr>
</thead>
</table>
| /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:**

<table>
<thead>
<tr>
<th>Location</th>
<th>Pros</th>
<th>Cons</th>
</tr>
</thead>
</table>
| /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:
  
a) PI:
  - Apply for a storage allocation (a.k.a. /project, if hasn't)
  - Add User 1 (sharing) and User 2 (shared) to /project

b) User 1 (sharing):
  - Set up `envs_dirs` to create a virtual environment in a /project location
  - Install software in the virtual environment

c) 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 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?
### 4) Migrate / clone virtual environment

- **Solution**

<table>
<thead>
<tr>
<th>To …</th>
<th>Run command</th>
</tr>
</thead>
</table>

4) Migrate / clone virtual environment

- Solution

To …

Export virtual environment recipe to file

```yaml
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=h5eea18b_7
```

4) Migrate / clone virtual environment

- **Solution**

<table>
<thead>
<tr>
<th>To …</th>
<th>Run command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Export virtual environment recipe to file</td>
<td>conda env export &gt; myenv.yml</td>
</tr>
<tr>
<td>Create a virtual environment from file</td>
<td>conda env create -f myenv.yml</td>
</tr>
</tbody>
</table>

5) Use virtual environment on Open OnDemand

[1] https://youtu.be/JLXN0AZgqgc
5) Use virtual environment on Open OnDemand

[1] https://youtu.be/JLXN0AZgaqc
5) Use virtual environment on Open OnDemand

[1] https://youtu.be/JLXN0AZgaqc
5) Use virtual environment on Open OnDemand

[1] https://youtu.be/JLXN0AZqaqc
5) Use virtual environment on Open OnDemand

How to:

Step 1: ssh to the cluster you want to use

<table>
<thead>
<tr>
<th>LSU HPC</th>
<th>LONI</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMIC / Super Mike 3</td>
<td>QB2</td>
</tr>
</tbody>
</table>

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

- 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)

```r
> install.packages("rgdal")
warning in install.packages("rgdal"):
  lib = "/home/packages/r/4.1.2/Sk5jengl/lib/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
```
### 6) More than Python

- **Solutions**

**Many non-python packages are managed by Conda too!**

<table>
<thead>
<tr>
<th>To install ...</th>
<th>Run command ...</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Languages</strong></td>
<td></td>
</tr>
<tr>
<td>R</td>
<td>conda install R</td>
</tr>
<tr>
<td>Perl</td>
<td>conda install perl</td>
</tr>
<tr>
<td>Julia</td>
<td>conda install julia -c conda-forge</td>
</tr>
<tr>
<td><strong>Dependencies</strong></td>
<td></td>
</tr>
<tr>
<td>hdf5</td>
<td>conda install hdf5</td>
</tr>
<tr>
<td>netcdf</td>
<td>conda install libnetcdf -c conda-forge</td>
</tr>
<tr>
<td>FFTW</td>
<td>conda install fftw</td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
</tbody>
</table>

[1] [https://anaconda.org/](https://anaconda.org/)
6) More than Python

• One more cool thing…
  – You can use language specific package management tools

<table>
<thead>
<tr>
<th>Language</th>
<th>Tool</th>
</tr>
</thead>
<tbody>
<tr>
<td>Python</td>
<td>pip</td>
</tr>
<tr>
<td>R</td>
<td>install.packages</td>
</tr>
<tr>
<td>Perl</td>
<td>cpan</td>
</tr>
<tr>
<td>Julia</td>
<td>Pkg</td>
</tr>
</tbody>
</table>

  – 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("rgdal") → Will fail!
    ```

  - Use Conda:

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

a) Conflict with system module

```
[spyder] $ module list
Currently loaded module files:
  1) python/3.8.5-anaconda

[spyder] $ 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/lib/python3.8/site-packages/numpy/__init__.py
Before Intel(R) MKL initialization ensuring correct out-of-the-box operation under cblas is not assured. Please install mkl-service, see http://github.com/IntelPython
from . import _distribution init
Traceback (most recent call last):
  File "/usr/local/packages/python/3.8.5-anaconda/lib/python3.8/site-packages/numpy/core
  File "/usr/local/packages/python/3.8.5-anaconda/lib/python3.8/site-packages/numpy/core
```

Fail!
7) Troubleshooting

a) Conflict with system module

   – Rule of thumb:

   Do **NOT** load system module if you are using your own installation!
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...
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
Nov 1, 2023
## Conda vs Singularity

<table>
<thead>
<tr>
<th></th>
<th>Conda / Virtual Environments</th>
<th>Singularity / Containers</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Availability</strong></td>
<td>All users</td>
<td>All users, but may need additional things</td>
</tr>
<tr>
<td><strong>Functionality</strong></td>
<td>Good (usually)</td>
<td>Better (more likely to work)</td>
</tr>
<tr>
<td><strong>Self-contained</strong></td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Isolation</strong></td>
<td>Yes (but still accessible from outside)</td>
<td>Perfect</td>
</tr>
<tr>
<td><strong>Editability</strong></td>
<td>Yes</td>
<td>No (image file itself) / Yes (a detour with sandbox)</td>
</tr>
<tr>
<td><strong>Disk usage</strong></td>
<td>Large</td>
<td>Smaller</td>
</tr>
<tr>
<td><strong>Portability</strong></td>
<td>Possible (with .yml file)</td>
<td>Great (copy-paste one file)</td>
</tr>
<tr>
<td><strong>Security</strong></td>
<td>Fair</td>
<td>Good</td>
</tr>
<tr>
<td><strong>Ease of use</strong></td>
<td>Good</td>
<td>May require more effort</td>
</tr>
</tbody>
</table>
### Conda vs Singularity

<table>
<thead>
<tr>
<th>Good for</th>
<th>Conda / Virtual Environments</th>
<th>Singularity / Containers</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Less hassle (usually?) to create and install software from scratch</td>
<td>• Less hassle if the developer releases a working container</td>
<td></td>
</tr>
<tr>
<td>• If you need to frequently make modifications</td>
<td>• If you don’t or don’t want to make changes after it is created</td>
<td></td>
</tr>
<tr>
<td>• If you need to access files from outside of the environment (e.g., compiling a code that uses some files in the virtual environment as dependencies)</td>
<td>• Portability</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Reduce disk usage</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Your system admins yelled at you about security risk</td>
<td></td>
</tr>
</tbody>
</table>
Contact us

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