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: **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
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
1) Problems

a) Dependencies (Welcome to Linux!)

BUSCO from QC to gene prediction and phylogenomics

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.
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/
- http://bioinf.uni-greifswald.de/augustus/
- https://github.com/soedinglab/metaeuk
- https://github.com/hyattpd/Prodigal
- http://hmmer.org/
- https://github.com/smiraarab/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/
- 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.
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, tBLASTx 2.2+, Augustus 3.2+, Prodigal, Metaev, HMMER 3.1+, SEPP, and R + ggplot2 for the plotting companion script. Some of these tools are necessary only for analysing certain types 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!)

```
[jasonli3@smic2 ~]$ module load python/3.6.2-anaconda-tensorflow
[jasonli3@smic2 ~]$ module list
Currently Loaded Modulefiles:
  1) python/3.6.2-anaconda-tensorflow
```
b) Permission denied (Welcome to HPC!)

```
[jasonli3@smic2 ~]$ module load python/3.6.2-anaconda-tensorflow
[jasonli3@smic2 ~]$ module list
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
```
b) **Permission denied** (Welcome to HPC!)

```bash
[jasonli@smic2 ~]$ module load python/3.6.2-anaconda-tensorflow
[jasonli@smic2 ~]$ module list
Currently Loaded Module files:
  1) python/3.6.2-anaconda-tensorflow
[jasonli@smic2 ~]$ pip install geos
Collecting geos
  Downloading https://files.pythonhosted.org/packages/49/5b/b8acf74c01187a36aa41b6523de9bbaa59c0a1d085f9e5f3745e247b/geos-2.13.1.tar.gz (409kB)
    100% [======================================] 409kB 3.0MB/s

File "/usr/local/packages/python/3.6.2-anaconda/lib/python3.6/site-packages/pip/attrs/_util/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 20.0.1 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.
[jasonli@smic2 ~]$
```
b) **Permission denied** (Welcome to HPC!)

- If you ask Google / ChatGPT…

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

---

**1) Problems**

**1. Why Conda?**

**2. Basic Usage**

**3. Advanced Tips**
1) Problems

c) Conflicted packages

- What if I need two packages with conflicted dependencies?

Tensorflow 1.13

PyTorch > 1.5
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?
1) Problems

Any of those apply to you?
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
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)
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…
b) How does it solve my problems?

• Dependency issue
  - Pack all dependencies in the VE. No need to rely on the system.
  - Tools like Conda can help install dependencies.

• Permission issue
  - Create a VE where you have write permission.
  - Does not need sudo permission

• Conflicted packages
  - Install in different VEs.

• Share / Migrate
  - Create VE in /project and share w/ group
  - Export recipe and build on a different system
c) **What is Conda?**

- **/path/to/envs**
  - **env1**
    - Python 3.7
    - numpy 1.13
    - scipy 0.19
  - **env2**
    - Python 3.9
    - pandas 1.5
  - **env3**
    - R 4.2.3
    - Seurat 4.3

Technology →
2) Virtual environment & Conda

c) What is Conda?

↑ **Software** system that implements the technology
2) Virtual environment & Conda

c) What is Conda?
2) Virtual environment & Conda

c) What is Conda?
**Summary**

Technology that helps with software installation →

↓ Software system that implements the technology

conda

- Python 3.7
  - numpy 1.13
  - scipy 0.19

- Python 3.9
  - pandas 1.5

- Perl 5.8
  - Bioperl 1.6

- R 4.2.3
  - Seurat 4.3
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

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)
```
2. Basic Usage

3. Advanced Tips

1. Why 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
```
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 ...

Create virtual environment

Activate virtual environment

Install software packages

To use ...

Activate virtual environment

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>Created 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>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>conda create -n ENVIRONMENT</td>
<td>Create a virtual environment</td>
</tr>
<tr>
<td>source activate ENVIRONMENT</td>
<td>Activate a virtual environment</td>
</tr>
<tr>
<td>conda deactivate</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>

```bash
(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>conda create -n ENVIRONMENT</td>
<td>Create a virtual environment</td>
</tr>
<tr>
<td>source 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>

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

<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>
<tr>
<td>conda list</td>
<td>List all installed software package</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>
</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>

### 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>
</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>

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/](https://pytorch.org/get-started/locally/)
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/
c) Bonus: Hot packages!

ii. Tensorflow (2.15.1, w/ GPU support)

```bash
$ 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
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

<table>
<thead>
<tr>
<th>Name</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>default / main / 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 Conda modules</th>
<th>Customized Conda</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Environments</strong></td>
<td><code>/home/$USER/.conda/envs/</code></td>
<td><code>/path/to/conda/envs/</code></td>
</tr>
<tr>
<td><strong>Cache</strong></td>
<td><code>/work/$USER/.conda/pkgs/</code></td>
<td><code>/path/to/conda/pkgs/</code></td>
</tr>
</tbody>
</table>

```bash
[jasonli3@smic1 ~]$ balance
User filesystem quotas for jasonli3 (uid 15827):
   Filesystem   MB used   MB quota
   /home        950       10000
   /work/project 329639   0      614737
Storage allocation   MB used   MB quota
```
2. Basic Usage

a) Method 1: Command lines

```
conda config [options]
```

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

b) Method 2: Configuration file

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

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

```ini
envs_dirs:
  - /work/jasonli3/.conda/envs/
  - /project/jasonli3/.conda/envs
pkgs_dirs:
  - /work/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**! |
c) Places to store your virtual environments:

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

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

4) Migrate / clone virtual environment

- **Solution**

To migrate or clone a virtual environment, you can export the environment recipe to a file:

```
conda env export > myenv.yml
```

To use this recipe in another environment, you can run:

```
conda env create -f myenv.yml
```

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 in Open OnDemand

[1] https://youtu.be/JLXN0AZgacq
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

<table>
<thead>
<tr>
<th>LSU HPC</th>
<th>LONI</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMIC, Super Mike 3</td>
<td>QB2, QB3</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](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)

```r
> 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!**

<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. Why Conda?
2. Basic Usage
3. Advanced Tips

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

• It gets even better…

  – 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("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 list
Currently Loaded Module files:
  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/lib/python3.8/site-packages/numpy/__init__.py: before Intel(R) MKL initialization ensuring correct out-of-the-box operation under c
essel is not assured. Please install mkl-service! See http://github.com/IntelPy
>> fail!
```

```
Fail!
```
7) Troubleshooting

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

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

- **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

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

<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>Self-contained</strong></td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Isolated</strong></td>
<td>Yes (but still accessible from outside)</td>
<td>Perfect (completely isolated from outside)</td>
</tr>
<tr>
<td><strong>Editability</strong></td>
<td>Yes</td>
<td>No (Must create a new image)</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 recipe)</td>
<td>Great (just 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 a little more understanding</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></td>
<td>• Less hassle to create and install software from scratch</td>
<td>• Less hassle if the developer releases a working container</td>
</tr>
<tr>
<td></td>
<td>• If you need to frequently make modifications</td>
<td>• If you don’t need to make changes after it is created</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Portability</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Reduce disk usage</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Your system admins yelled at you about security risk</td>
</tr>
</tbody>
</table>
Contact us

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