Magic Tools to Install and Manage Software:

Virtual Environment.

Oleg N. Starovoytov
HPC User Services
LSU HPC / LONI
sys-help@loni.org

Louisiana State University, Baton Rouge
October 22nd, 2025

Outlines

- **Introduction to CONDA virtual environment**

1. Installing software packages on an HPC cluster
2. Introduction to virtual environment (VE)
3. Introduction to CONDA environment and package manager
4. Basics of installing the software packages

- **Software management using CONDA environment**

1. Configuring the CONDA environment on an HPC cluster
   - Configuring the CONDA environment for sharing
   - Managing CONDA environments
2. Configuring the CONDA environment in OOD
3. Installation of non-Python packages
4. Troubleshooting

Introduction

The HPC cluster usage policies

https://www.hpc.lsu.edu/users/hpcpolicy.php

All software available on HPC clusters

https://www.hpc.lsu.edu/docs/guides/

How to request a software package install

https://www.hpc.lsu.edu/docs/guides/software-request.php

Installing software packages on an HPC cluster

- **Installing software packages on an HPC cluster**
  - System Package Managers (You do not have root access on HPC),
  - Modules on HPC environment with preinstalled software packages,
  - Precompiled executables,
  - Installation from source using various compilers,
  - Singularity/Apptainer (For Containerized Applications).

- **Installing software packages on an HPC cluster**

  – System Package Managers, **YUM/DNF** (RHEL/CentOS/Rocky):
  sudo yum install package-name

    - You do not have root privileges on the HPC LONI or LSU clusters!

- **Installing software packages on an HPC cluster**

  – Modules with preinstalled software packages:
  module load preinstalled package

  – Issues:
    - Module conflicts,
    - Performance variability,
    - Missing modules,
    - Permission restrictions, etc.

- **Installing software packages on an HPC cluster**

  – Precompiled executables:
  only Linux binaries (no Windows .exe)

  – Issues:
    - Architecture mismatch,
    - Library dependencies,
    - Operating system differences,
    - MPI compatibility,
    - File system and path issues, etc.

- **Installing software packages on the HPC cluster**

  – Installation from source using various compilers:
  GNU, Intel ® oneAPI compilers, PGI ®, NVIDIA® HPC SDK, etc.

  – Issues:
    - Dependency management (Spack might help),
    - MPI and parallelism issues,
    - Optimization and performance tuning,
    - Permissions and Environment issues,
    - Conflicting libraries, etc.

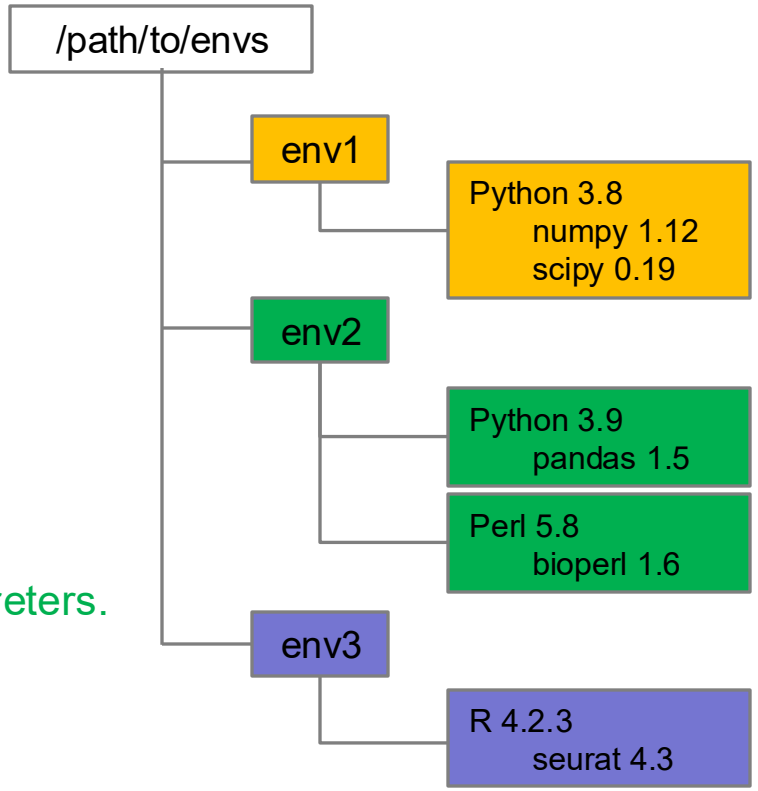
- **Installing software packages on an HPC cluster**

  - Singularity/Apptainer (For Containerized Applications): singularity pull docker://ubuntu:20.04 using ubuntu_20.04.sif
  - Issues:
    - Learning to build and run singularity images,
    - Slower Performance for I/O-intensive tasks,
    - Non-customizable.

Introduction to virtual environment (VE)

- **Introduction to virtual environment (VE)**

The virtual environment is an isolated execution environment that can be used to run software applications independently of the host.



Key features:

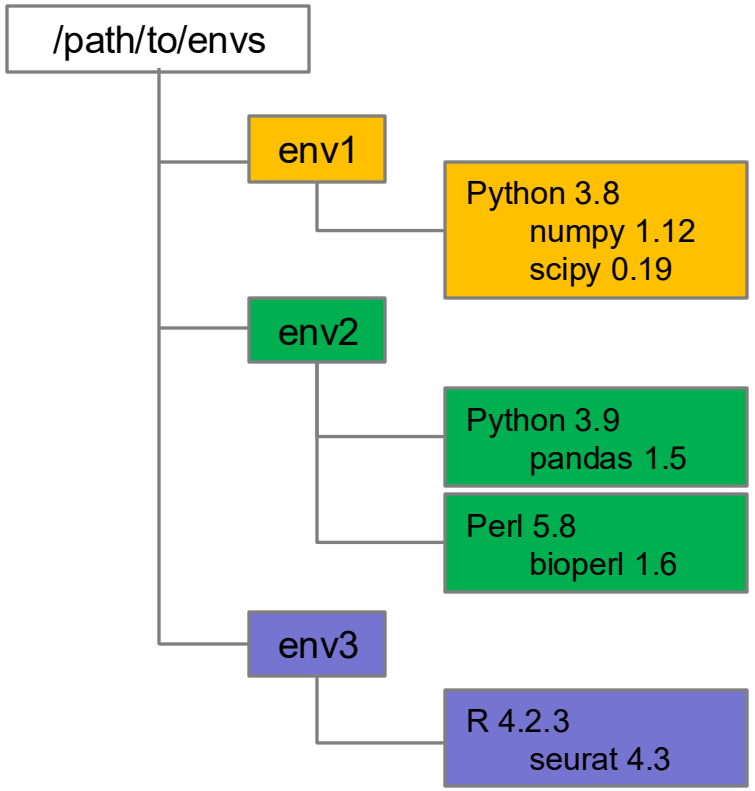
-Isolation – It runs independently avoiding conflicts with system-wide software.

-Customizable - Allows installation of specific libraries, frameworks, and interpreters.

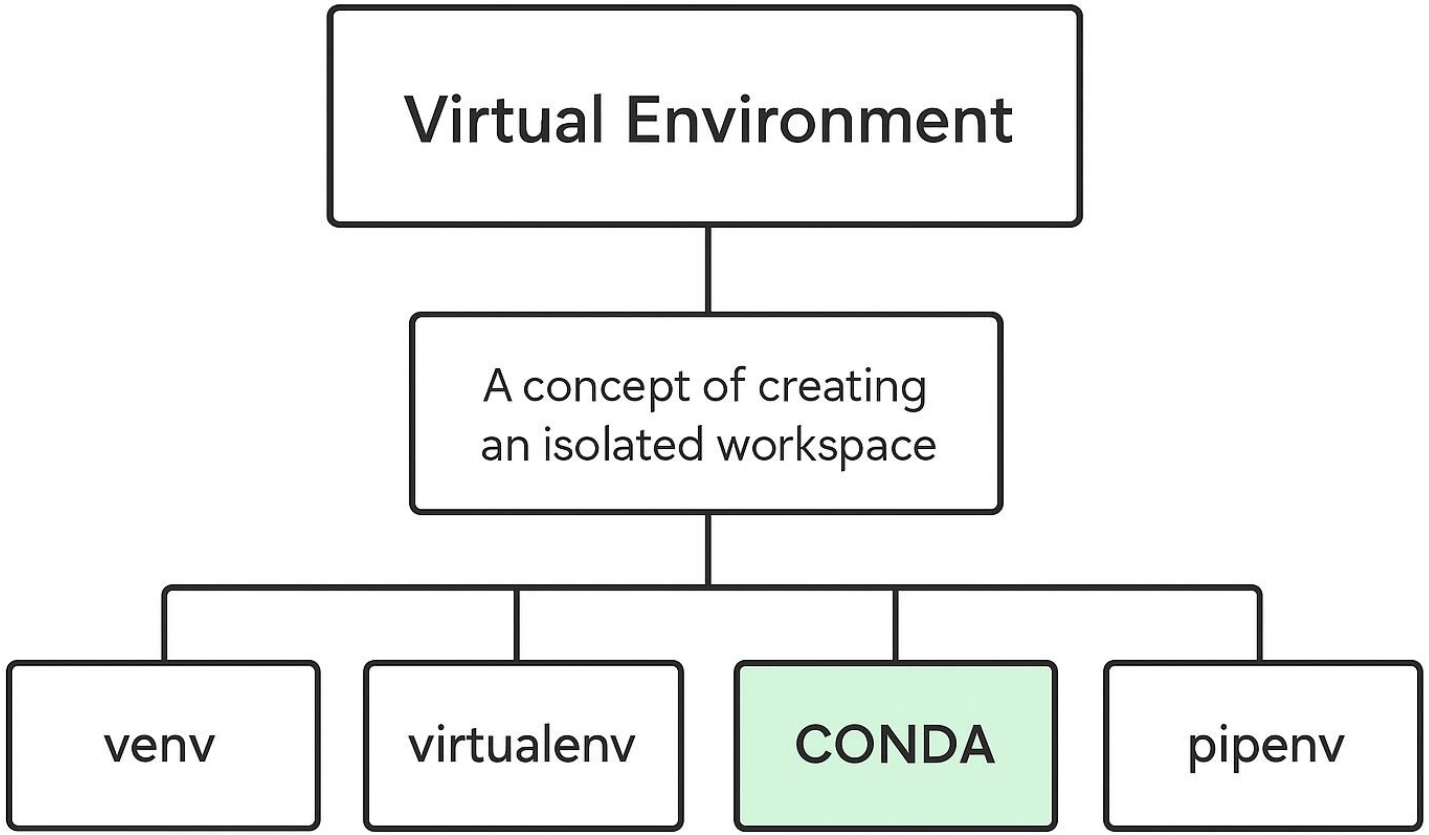
-Portability - Configurations can be shared and replicated across machines.

-Security - Reduces the risk of unintended system modifications.

- **Introduction to virtual environment (VE)**
  - Why use a virtual environment on HPC?
    - Package management
      - Isolated from the host system environment.
      - All dependencies are installed within the VE.
      - Sharing, migrating, or recreating the VE.



Introduction to CONDA® virtual environment



Introduction to CONDA environment and package manager

Introduction to CONDA® virtual environment

- **Introduction to CONDA environment and package manager**

–CONDA® (For Python & Scientific Packages):

R, Ruby, Lua, Scala, Java, JavaScript, C, C++, and Fortran packages

- **Introduction to CONDA environment and package manager**

  - How to obtain CONDA: there are two ways
    - It is preinstalled on LONI and HPC LSU clusters:
      module av conda

    - You can install your preferred CONDA distribution:
      - Anaconda: Full-size CONDA and a lot of Python packages, Anaconda Inc.
      - Miniconda: Minimum size CONDA and Python only, Anaconda Inc.
      - Miniforge: Minimum size CONDA and Python only. Community support.

Introduction to CONDA® virtual environment

LSU

- **Introduction to CONDA environment and package manager**

  - How to obtain CONDA: there are two ways
    - It is preinstalled on LONI and HPC LSU clusters:
      module av conda

    - You can install your preferred CONDA distribution:
      - Anaconda: Full-size CONDA and a lot of Python packages, Anaconda Inc.
      - Miniconda: Minimum size CONDA and Python only, Anaconda Inc.
      - Miniforge: Minimum size CONDA and Python only. Community support.

- **Introduction to CONDA environment and package manager**

  - How to obtain CONDA
    - It is preinstalled on LSU and LONI HPC clusters

Command	Description
module av conda	Checking if there is a CONDA module.
module load conda	Loading the CONDA module.
conda init	Initializing CONDA environment.

- **Introduction to CONDA environment and package manager**

  - How to obtain CONDA
    - You can install your own CONDA distribution

Command	Description
wget https://github.com/conda-forge/miniforge/.../Miniforge3-25.1.1-2-Linux-x86_64.sh	Download the CONDA installation script.
chmod u+x Miniforge3-25.1.1-2-Linux-x86_64.sh	Change permission for the CONDA script.
./Miniforge3-25.1.1-2-Linux-x86_64.sh	Run the CONDA installation.

- **Introduction to CONDA environment and package manager**

  – Getting help

Command	Description
conda --version	Display the CONDA version.
conda info	Display the CONDA system info.
conda --help	To learn about available built-in commands.
conda [command] --help	To check the available options for a particular command.

- **Introduction to CONDA environment and package manager**

  – To create a virtual environment

Command	Description
conda create –n VIRT_ENV	Create a virtual environment with the CONDA.
source activate VIRT_ENV	Activate the CONDA virtual environment.
conda install PACKAGE-NAME	Install some packages with the CONDA installation.

- **Introduction to CONDA environment and package manager**

  – To use a virtual environment

Command	Description
conda env list / conda info --envs	List all available environments.
source activate VIRT_ENV	Activate the CONDA virtual environment.
python script_name.py	Run software or/and use installed packages.
conda deactivate	Deactivates the CONDA virtual environment.

Basics of installing the software packages

- **Basics of installing the software packages**

  – To install packages in a virtual environment

Command	Description
conda install PACKAGE-NAME	Install a software package.
conda install PACKAGE-NAME=version	Install a software package with a particular version.
conda install PACKAGE-NAME=version –c CHANNEL	Install a software package with a particular version and from a specific channel.
conda install PACKAGE-NAME1 PACKAGE-NAME2 …	Install multiple packages.

- **Basics of installing the software packages**

  – Other useful commands

Command	Description
conda search PACKAGE-NAME	Searching a software package.
conda search PACKAGE-NAME=version - - info	Search a software package with a particular version and from a specific channel.
conda update/upgrade PACKAGE-NAME	Update a package to the latest version.
conda uninstall/remove PACKAGE-NAME	Uninstall or remove a package.

Always use a virtual environment with CONDA®

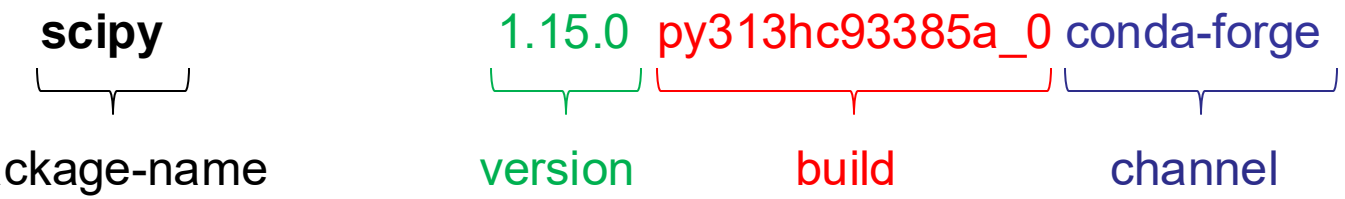
Introduction to CONDA® virtual environment

Exercise1:

1. Create a virtual environment, **myenv1**: conda create –n **myenv1**
2. Activate conda virtual environment: conda activate **myenv1**
3. Search for the **SciPy** version: conda search scipy
4. Install the second latest version: conda install scipy=**version**
5. Check the version of **SciPy** and **Python**: conda list

Exercise2:

1. Create a virtual environment, **myenv2.**
2. Search for the **SciPy** version.
3. Install the second latest version with the build: conda install scipy=**version=build**
4. Check the version of **SciPy** and **Python.**



Configuring the CONDA environment on an HPC cluster

- **Configuring the CONDA environment on an HPC cluster**

  – Obtaining packages from various channels

Channel	Description
conda-forge	Community supported for general purposes.
bioconda	Community supported for bioinformatics.
nvidia / cuda	NVIDIA official support.
pytorch	Pytorch official support.

- **Configuring the CONDA environment on an HPC cluster**

  – Configuring CONDA channels, check the .condarc file

Command	User CONDA path
conda config - -show channels	List available channels
conda config - -prepend channels NAME	Adding a channel with high priority
conda config - -append channels NAME	Adding a channel with low priority

- **Configuring the CONDA environment on an HPC cluster**

  – Configuring CONDA path, check the .condarc file

Path	Default CONDA path	User CONDA path
Environment	/home/$USER/.conda/envs	/path/to/your/directory/envs
Package	/home/$USER/.conda/pkgs	/path/to/your/directory/pkgs

- **Configuring the CONDA environment on an HPC cluster**

  – Configuring CONDA path, check the .condarc file

Command	Description
conda config - -add envs_dirs /path/to/envs	Adding a desired path to the environment directory.
conda config - -add pkgs_dirs /path/to/pkgs	Adding a desired path to the package directory.
conda config - -remove envs_dirs /path/to/envs	Removing a desired path to the environment directory.
vi ~/.condarc	Manually add paths to CONDA config file.

- **Configuring the CONDA environment on an HPC cluster**

  - Configuring CONDA path, check the .condarc file

```
olegsupp@mike1 /work/olegsupp$
olegsupp@mike1 /work/olegsupp$ vi ~/.condarc

pkgs_dirs:
  - /work/olegsupp/Programs/CONDA/pkgs
envs_dirs:
  - /work/olegsupp/Programs/CONDA/envs
channels:
  - conda-forge
  - defaults
  - bioconda
```

- **Configuring the CONDA environment on an HPC cluster**

  – Configuring CONDA environment at specific directories

Directory	Description	Space quota
/home	Available for all users. Backed up periodically.	10 GB
/work	Available for all users. Not backed up. All files are subject for purge after 60 days of inactivity.	Unlimited
/project	Available only for PI. (Valid for one year, subject for renewal) Not backed up. Shared among research team members.	from 100 GB

- **Configuring the CONDA environment for sharing**

  - Configuring CONDA for sharing includes three steps

Step	Description
1	PI should apply for a storage allocation. Email sys-help@loni.org and make a request to add users to the project allocation.
2	Set up envs_dirs to redirect a virtual environment to the /project directory. Install software packages in the virtual environment.
3	Set up envs_dirs to access a virtual environment in the /project directory.

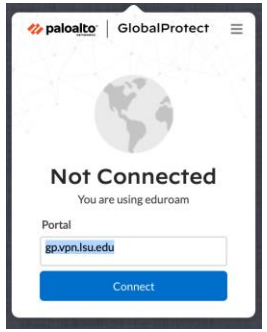
- **Managing CONDA environments**

Command	Description
conda env export –n VIRT_ENV > environment.yml	Export every package including dependencies.
conda env export - -from-history > environment.yml	Export only packages explicitly asked for.
conda env create -n VIRT_ENV –f environment.yml	Create VIRT_ENV using environment.yml file.
conda list --export > requirements.txt	Export list of packages to the text file.
conda create -name VIRT_ENV –file requirements.txt	Create VIRT_ENV using requirements.txt file.

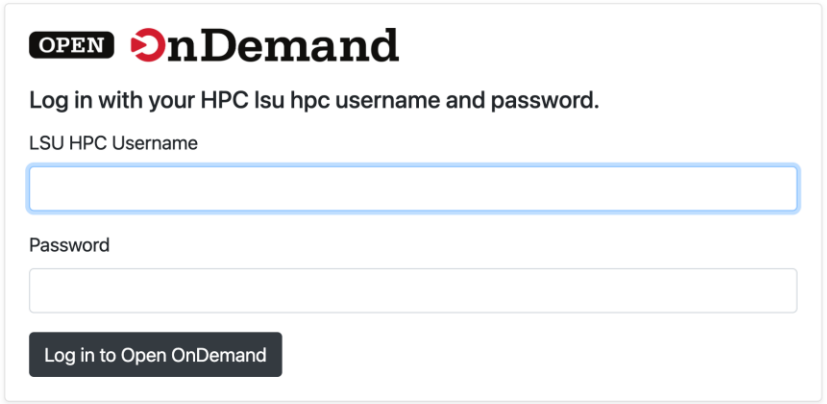
Configuring the CONDA environment with OOD

Software management using CONDA® environment

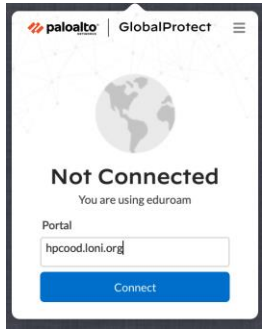
- **Configuring the CONDA environment with OOD**



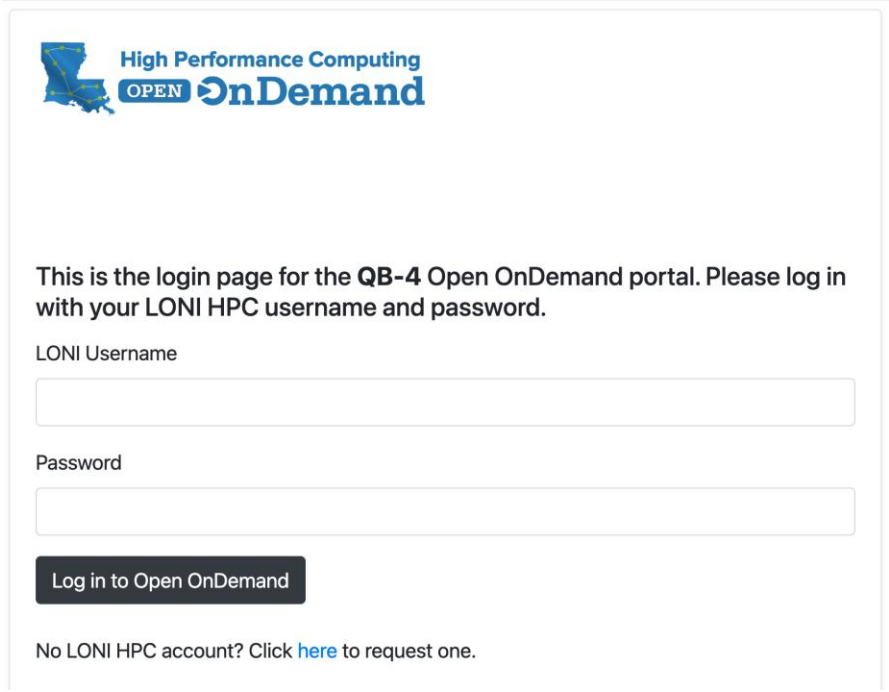
You need to use a VPN if you are outside of campus.



https://ondemand.mike.hpc.lsu.edu



You need to use a VPN if you are outside of campus.

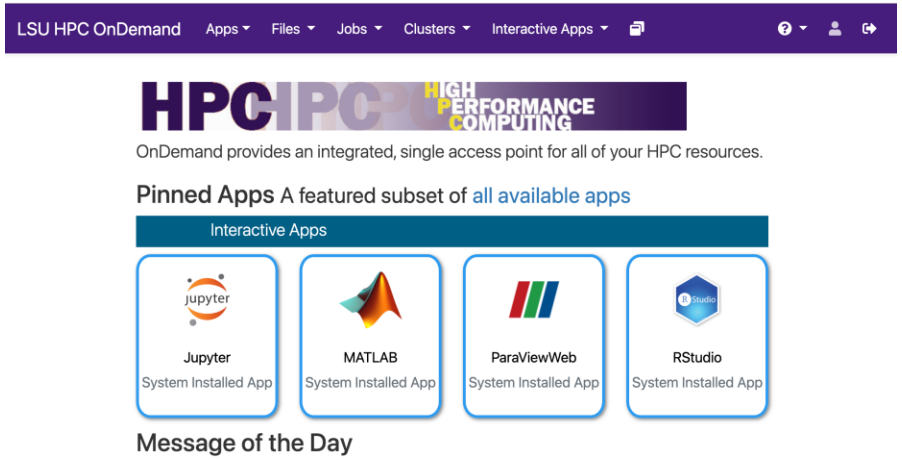


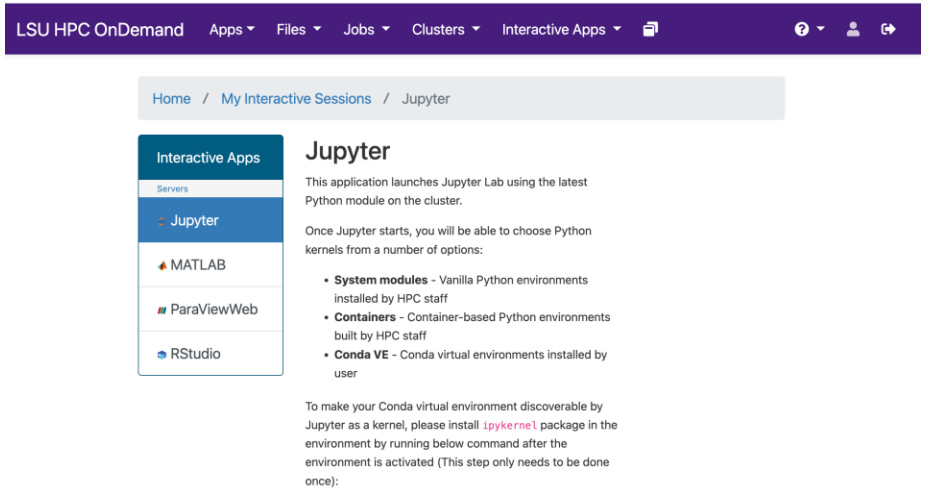
https://ondemand.qbc.loni.org

https://ondemand.qbd.loni.org

Software management using CONDA® environment

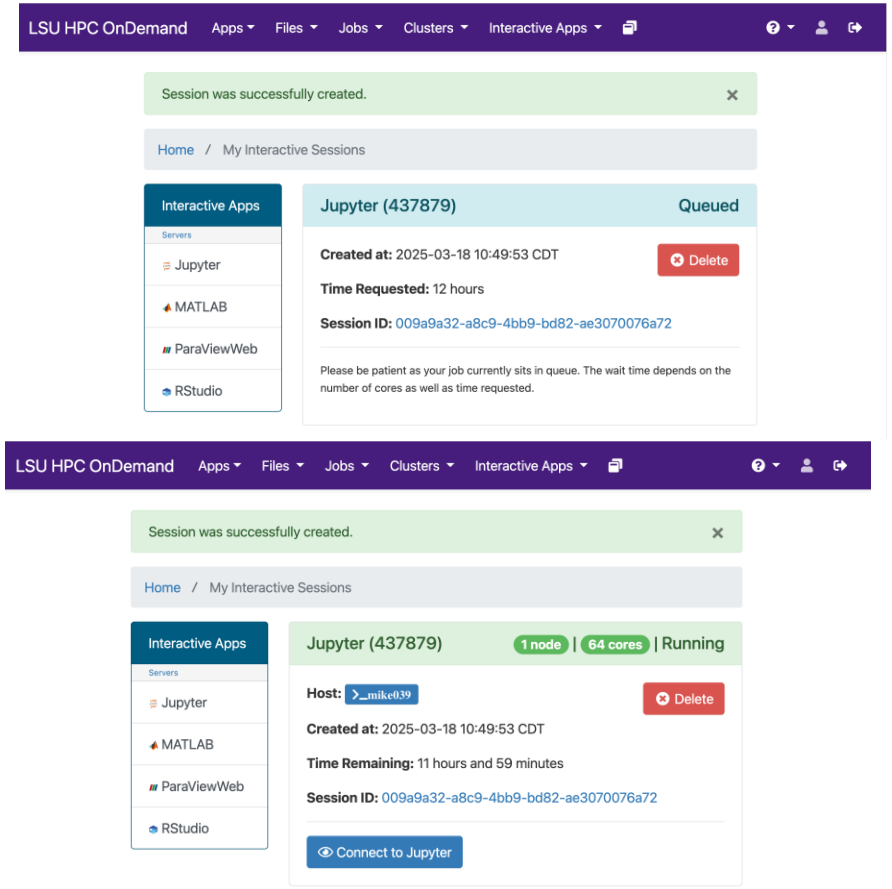
- **Configuring the CONDA environment with OOD**

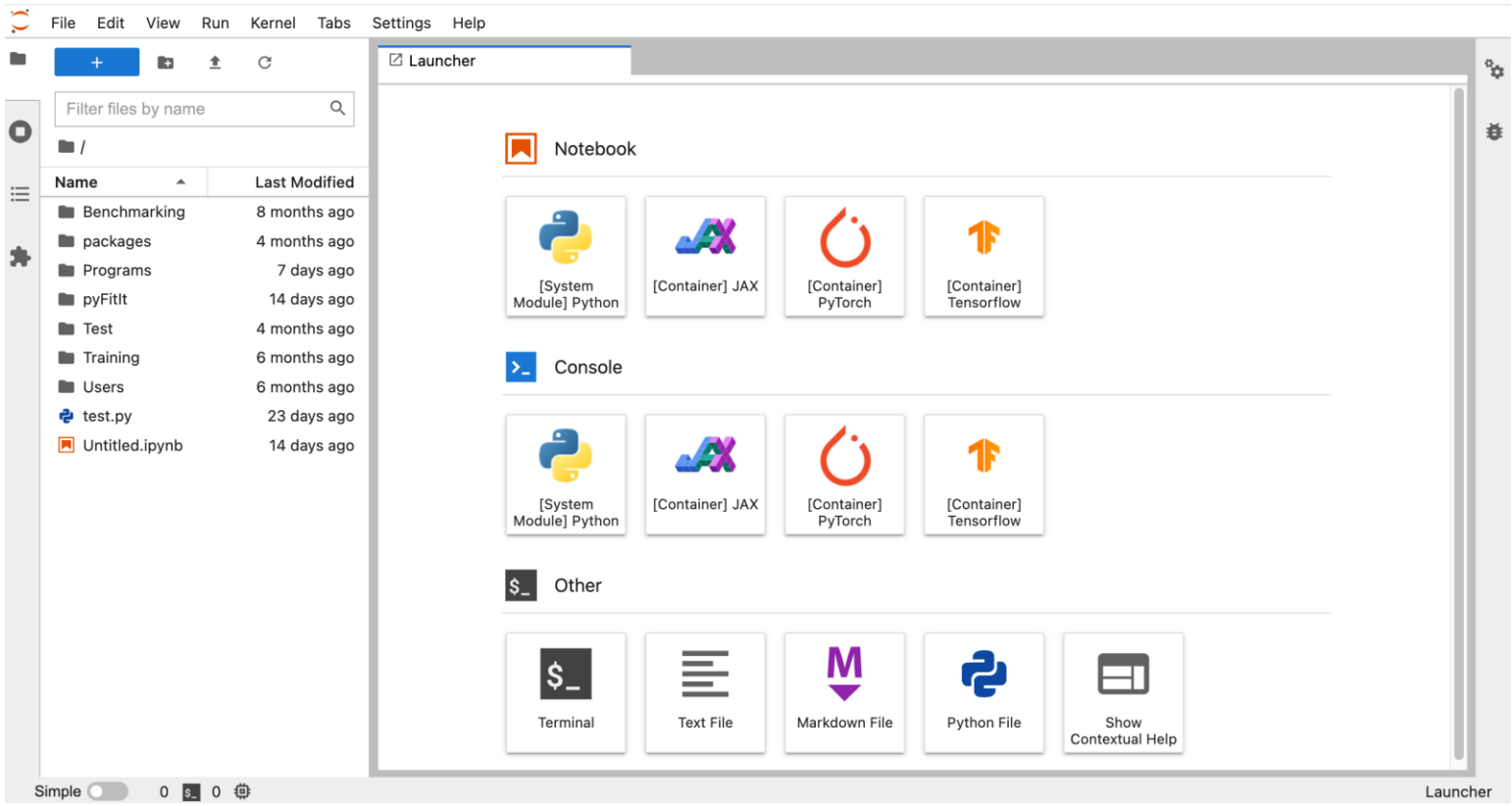




Software management using CONDA® environment

- **Configuring the CONDA environment with OOD**





Software management using CONDA® environment

- **Configuring the CONDA environment with OOD**

Command	Description
conda create –n VIRT_ENV	Create a CONDA environment first.
source activate VIRT_ENV	Activate CONDA environment next.
conda install ipykernel	Install ipykernel in CONDA environment.

Software management using CONDA® environment

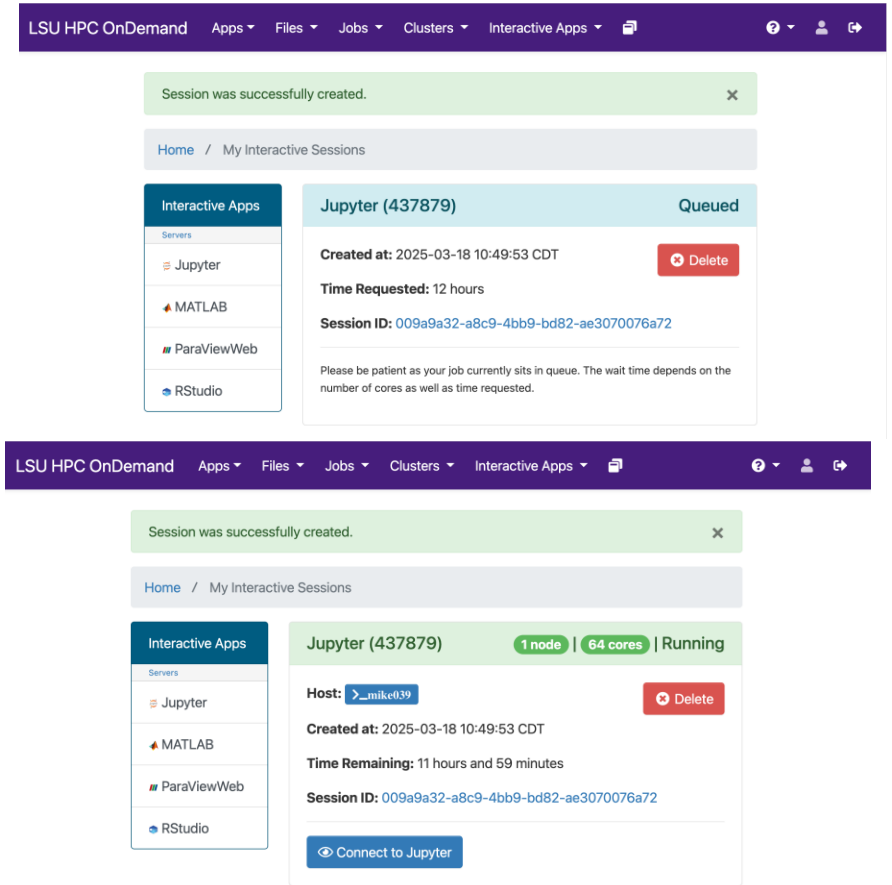
- **Configuring the CONDA environment with OOD**

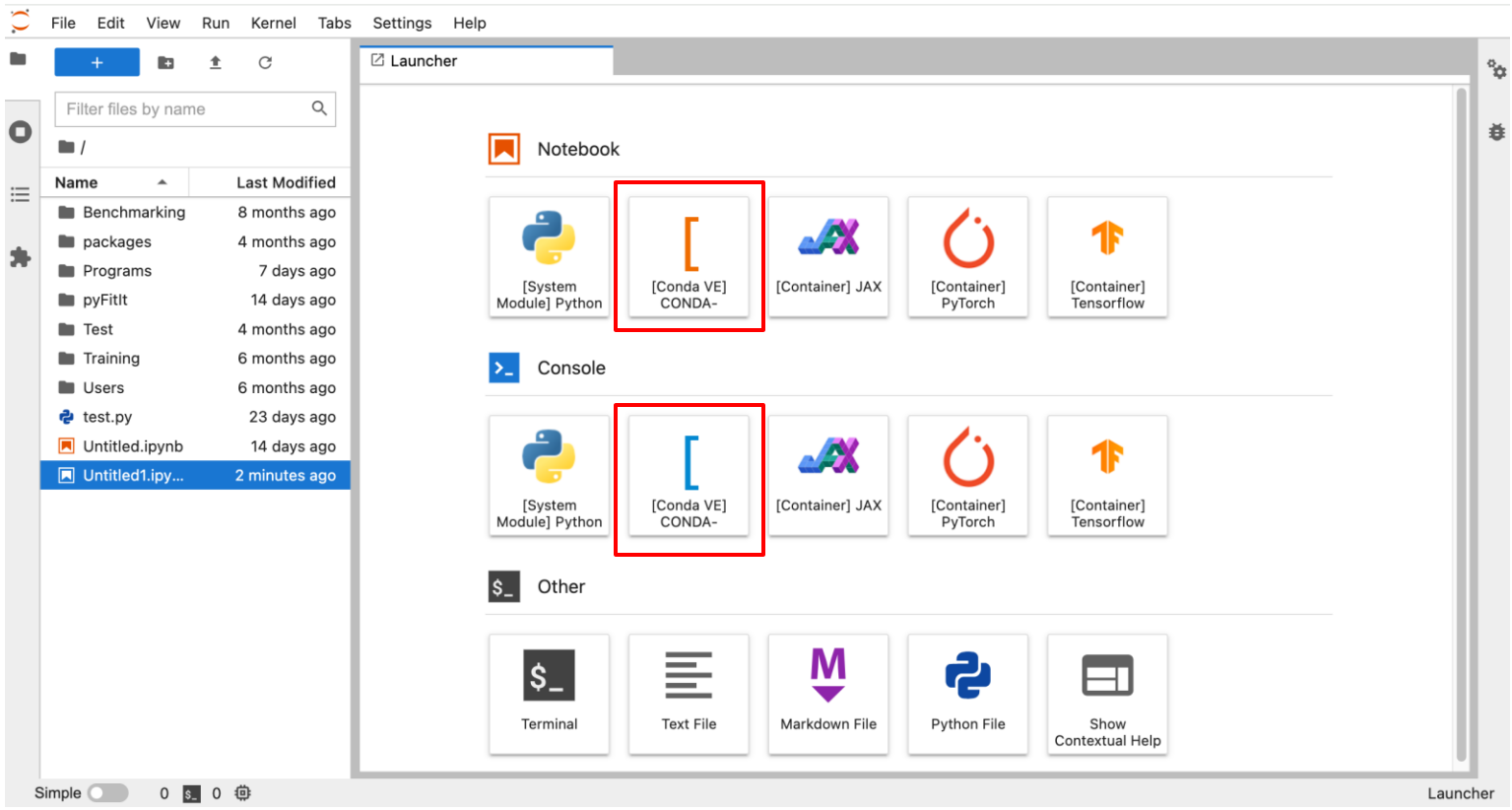
```
olegsupp@mike2 /work/olegsupp$ conda create -n VIRT_ENV python=3.12
Channels:
 - conda-forge
 - defaults
 - bioconda
Platform: linux-64
Collecting package metadata (repodata.json): done
Solving environment: done


==> WARNING: A newer version of conda exists. <==
    current version: 23.11.0
    latest version: 25.1.1

Please update conda by running

    $ conda update -n base -c conda-forge conda


## Package Plan ##

  environment location: /work/olegsupp/Programs/CONDA/envs/VIRT_ENV

  added / updated specs:
    - python=3.12


The following packages will be downloaded:

    package                    |            build
    ---------------------------|-----------------
    libsqlite-3.49.1           |       hee588c1_2         897 KB  conda-forge
    python-3.12.9              |h9e4cc4f_1_cpython        30.2 MB  conda-forge
    setuptools-75.8.2          |     pyhff2d567_0         760 KB  conda-forge
    ------------------------------------------------------------
                                           Total:        31.8 MB

The following NEW packages will be INSTALLED:

  _libgcc_mutex      conda-forge/linux-64::_libgcc_mutex-0.1-conda_forge
  _openmp_mutex      conda-forge/linux-64::_openmp_mutex-4.5-2_gnu
  bzip2              conda-forge/linux-64::bzip2-1.0.8-h4bc722e_7
  ca-certificates    conda-forge/linux-64::ca-certificates-2025.1.31-hbcca054_0
  ld_impl_linux-64   conda-forge/linux-64::ld_impl_linux-64-2.43-h712a8e2_4
  libexpat           conda-forge/linux-64::libexpat-2.6.4-h5888daf_0
  libffi             conda-forge/linux-64::libffi-3.4.6-h2dba641_0
  libgcc             conda-forge/linux-64::libgcc-14.2.0-h767d61c_2
  libgcc-ng          conda-forge/linux-64::libgcc-ng-14.2.0-h69a702a_2
  libgomp            conda-forge/linux-64::libgomp-14.2.0-h767d61c_2
  liblzma            conda-forge/linux-64::liblzma-5.6.4-hb9d3cd8_0
  libnsl             conda-forge/linux-64::libnsl-2.0.1-hd590300_0
  libsqlite          conda-forge/linux-64::libsqlite-3.49.1-hee588c1_2
  libuuid            conda-forge/linux-64::libuuid-2.38.1-h0b41bf4_0
  libxcrypt          conda-forge/linux-64::libxcrypt-4.4.36-hd590300_1
  tzdata             conda-forge/noarch::tzdata-2025a-h78e105d_0
  wheel              conda-forge/noarch::wheel-0.45.1-pyhd8ed1ab_1


Proceed ([y]/n)? y


Downloading and Extracting Packages:

Preparing transaction: done
Verifying transaction: done
Executing transaction: done
#
# To activate this environment, use
#
#     $ conda activate VIRT_ENV
#
# To deactivate an active environment, use
#
#     $ conda deactivate
```

```
Last login: Tue Mar 18 09:35:48 CDT 2025 on pts/6
(base) olegsupp@mike2 /work/olegsupp$ source activate VIRT_ENV
(VIRT_ENV) olegsupp@mike2 /work/olegsupp$ conda install ipykernel
Channels:
 - conda-forge
 - defaults
 - bioconda
Platform: linux-64
Collecting package metadata (repodata.json): done
Solving environment: done


==> WARNING: A newer version of conda exists. <==
    current version: 23.11.0
    latest version: 25.1.1

Please update conda by running

    $ conda update -n base -c conda-forge conda
```

Software management using CONDA® environment

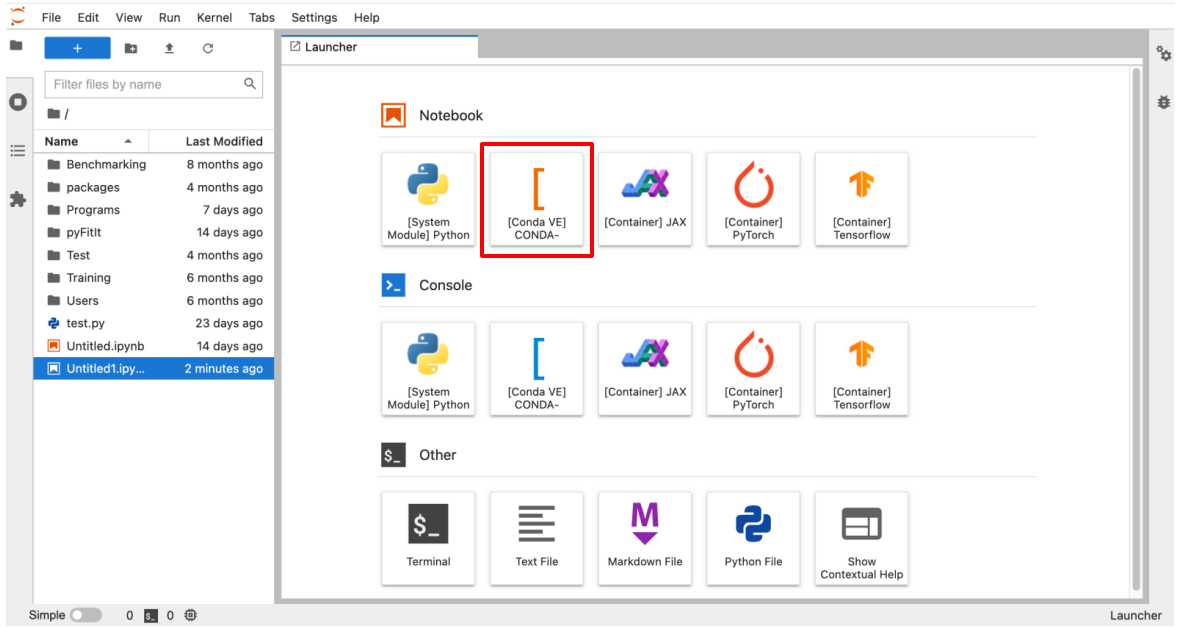
- **Configuring the CONDA environment with OOD**

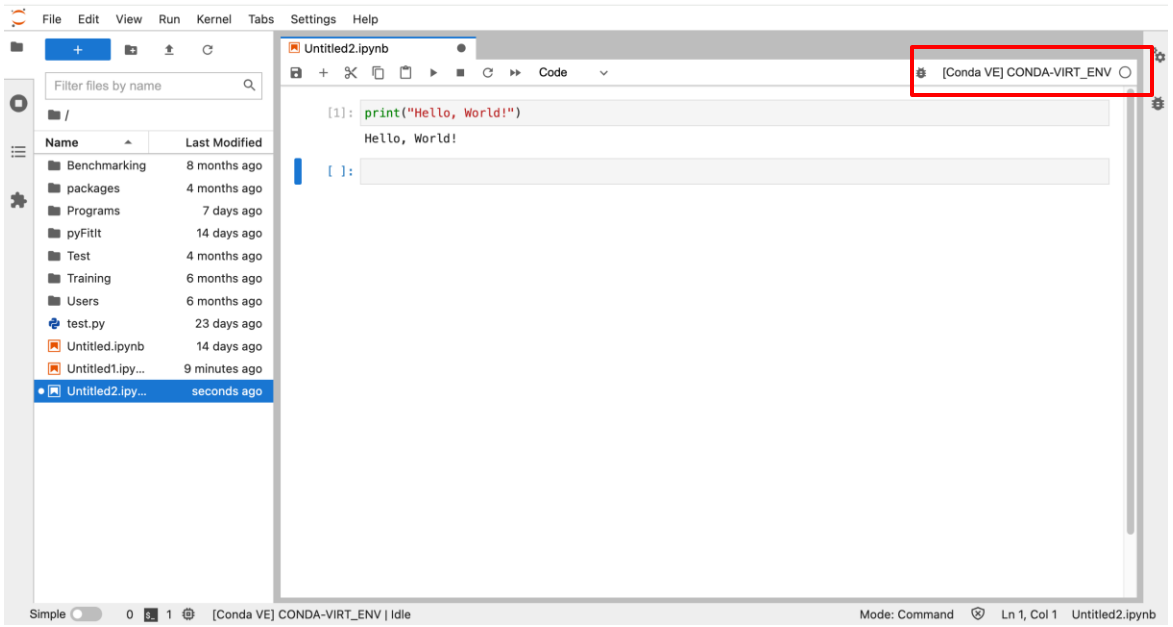




Software management using CONDA® environment

- **Configuring the CONDA environment with OOD**





Installation of non-Python packages

- **Installation of non-Python packages**

Command	Description
conda install R	Install R language.
conda install perl	Install perl language.
conda install hdf5	Install hdf5 dependency.
conda install fftw	Install fftw dependency.

- **Installation of non-Python packages**

Command	Description
module load r	Load R module
R install.packages(“Seurat”)	Install Seurat package

Command	Description
conda create –n seurat source activate seurat	Create and activate seurat virtual environment
conda install r-seurat	Install seurat package

Troubleshooting

- **Troubleshooting**

I can not switch to the new CONDA environment even after I load that.

Command	Description
unset conda	Unset the CONDA environment.
module purge	Clean up the environment unsetting all modules.
module load conda	Load the CONDA module to the user environment.
conda init	Initialize the CONDA environment.

- **Troubleshooting**

I created my virtual environment, but I do not see the pytorch package.

Command	Description
conda activate VIRT_ENV	Activate CONDA environment.
conda list torch	Check if torch is installed.
conda install pytorch	Install pytorch while CONDA environment is active.

- **Troubleshooting**

My package is corrupted. Cache files might be partially purged in /work/$USER/

Command	Description
conda clean -f	This command will clean all cache files.
conda clean --all --yes	Remove all incomplete, temporary files, tarballs, and unused package caches while keeping your environment intact.
conda remove –n VIRT_ENV --all	Remove virtual environment.

Summary

Feature	Virtual environment	Containers	Virtual machines
Scope	Isolated dependencies withing a single project	Isolated an entire application and its dependencies	Emulates a complete OS
Overhead	Low (just dependence management)	**Medium** (includes app and dependencies)	**High** (emulates full OS)
Performance	Fast (only manages Python runtime)	**Medium** (OS level virualization)	**Slow** (full OS boot required)
Storage size	Small	**Medium**	**Large**
Use case	Python development, HPC	Application packaging, deployment	Running different OS environments
Example tools	CONDA®, venv, virtualenv, pipenv	Docker, Singularity	VMware, VirtualBox

Available Tools

Implementation	Type	Containers
venv	Built-in Python module	Lightweight environment creation (built into Python3.3).
virtualenv	Third-party package	Older, more flexible tool.
conda	Cross-language environment manager	Creates environments that can include non-Python libraries.
pipenv	Modern project managers	Build on virtual environments with dependency tracking.

Available Tools

Feature	venv	CONDA
Type	Environment manager only	Package + Environment manager
Language supported	Python only	Python, R, C, C++, Fortran, etc.
Package source	PyPI (via pip)	CONDA repositories and channels
Speed and reliability	Compiles from source	Installs precompiled binaries
Easy of use for data science	Basic setup	Rich ecosystem (Anaconda, scientific packages pre-built)
System libraries	Managed separately	Can include system-level libraries (e.g. libblas, ffmpeg, etc.)

Available Tools

using **venv**

```
python3 -m venv TEST                    # create environment TEST
source TEST/bin/activate                # activate (Linux/macOS)
pip install numpy                       # install packages inside TEST
deactivate                              # exit environment
```

using CONDA

```
conda create -n TEST python=3.10 numpy pandas   # create new env with Python and packages
conda activate TEST                             # activate
conda install matplotlib                        # install new package
conda deactivate                                # deactivate
```