



Consorti de
Serveis Universitaris
de Catalunya

Conda environment system & how to use it on CSUC machines

Víctor Pérez

16 / 12 / 2021

Index

- 1. What is conda?**
- 2. Scope of the project**
- 3. How to use conda**
- 4. Conda environments**
- 5. Package management**
- 6. Python through conda**
- 7. R through conda**

What is conda?

- Originally: **Anaconda**, a distribution of Python including common scientific packages

<https://www.anaconda.com/>

- Extended to include **R** and R packages, scientific libraries, other software, etc.
- **conda**: core package manager for the Anaconda project

What is conda?

- **conda** installs and updates binary versions of Python and R packages from its own (or third party) repositories
- It is an alternative to other repository systems, like **pip** for Python or **CRAN** for R
- It is also a way to manage dependencies for Python and R packages

But conda isn't...

- A repository of **system** software packages (such as apt or yum)
- A repository of **source code**
- A replacement for **environment modules**
- Exhaustive or infallible

Scope of the project

- Python
- R
- Jupyter, Spyder, Rstudio...
- NumPy, SciPy, Pandas, Numba, Dask...
- Scikit-learn, TensorFlow, Theano...
- Matplotlib, Bokeh, Datashader, Holoviews...
- A variety of external libraries and tools

Conda channels

- Channels are thematic collections of packages, useful to avoid version conflicts (equivalent to repositories)
- Examples:
 - **pkgs/main**: default channel
 - **conda-forge**: large collection of third party packages
 - **bioconda**: software for bioinformatics
 - **r**: tailored to R users

How to use conda

- We need to load an **environment module** to configure **conda**:

module load conda/current

How to use conda

- Then we use the command **conda (+ action)** to run it:

conda list

conda activate

conda create

conda search

conda install

conda update

conda help

Conda environments

- Inside a given installation of conda, there are a number of **environments**
- Environments are **profiles**: each will have a different list of packages and versions installed
- We are transitioning to working exclusively with **private environments** owned by users

Activating/Deactivating environments

- To see a list of environments: **conda env list**

```
vperezg@login2:/home/vperezg>conda env list
# conda environments:
#
base                *  /prod/apps/conda/current
matplotlib           /scratch/vperezg/.conda/envs/matplotlib
r-4                  /scratch/vperezg/.conda/envs/r-4
tensorflow            /scratch/vperezg/.conda/envs/tensorflow
tensorflow-gpu       /scratch/vperezg/.conda/envs/tensorflow-gpu
```

- To load an env: **conda activate <env_name>**

```
vperezg@login2:/home/vperezg>conda activate matplotlib
(matplotlib) vperezg@login2:/home/vperezg>
```

- To unload: **conda deactivate**

```
(matplotlib) vperezg@login2:/home/vperezg>conda deactivate
vperezg@login2:/home/vperezg>
```

Activating/Deactivating environments

- To see the contents of an env: **conda list [-n env_name]**

```
(matplotlib) vperezg@login2:/home/vperezg>conda list
# packages in environment at /scratch/vperezg/.conda/envs/matplotlib:
#
# Name                                Version                                Build                                Channel
_libgcc_mutex                          0.1                                    conda_forge                          conda-forge
_openmp_mutex                           4.5                                    1_gnu                                 conda-forge
ca-certificates                         2020.12.5                              ha878542_0                            conda-forge
certifi                                  2020.12.5                              py39hf3d152e_1                        conda-forge
cyclur                                   0.10.0                                  py_2                                   conda-forge
dbus                                     1.13.6                                  hfdff14a_1                            conda-forge
expat                                    2.2.10                                  h9c3ff4c_0                            conda-forge
fontconfig                               2.13.1                                  hb2837de_1004                         conda-forge
```

(by default, currently activated environment)

- Note: **source activate** and **source deactivate** are obsolete

Environments

- Users can create their own private envs (stored at `$HOME/.conda/envs`) visible only to them
- To create a new empty environment: **conda create -n <env_name>**
- To create a new environment with packages preinstalled in it: **conda create -n <env_name> *[list of packages]***

Environments

- To install one or more packages in a private env: **conda install [-n *env_name*] <packages>**
- If no env is specified, they are installed in the currently active environment
- Version and channel can also be specified: **conda install [-n *env_name*] [-C *channel*] <package=version>**

Environments

- To update packages in an environment: **conda update [-n *env_name*] <specific packages>** or **conda update [-n *env_name*] --all**
- To uninstall packages: **conda remove [-n *env_name*] <packages>**
- To completely delete an environment: **conda remove -n <env_name> --all**

Environments

- Private environments can be shared with other users in your group
- To be able to use environments created by your group colleague, create a `.condarc` file in your home with the following lines:
- Or just contact us to do it for you!

Using Python through conda

```
#!/bin/bash
```

```
#SBATCH -p std
```

```
#SBATCH -N 1
```

```
#SBATCH -n 1
```

```
module load conda/current
```

```
conda activate <your_environment>
```

```
python example.py
```

Using R through conda

```
#!/bin/bash
```

```
#SBATCH -p std
```

```
#SBATCH -N 1
```

```
#SBATCH -n 1
```

```
module load conda/current
```

```
conda activate <your_environment>
```

```
Rscript example.R
```

Using pip or CRAN through conda

- It is possible to install Python packages using pip, or R packages using BioConductor or CRAN, but it requires configuring a proxy
- Let us know if you need to do this so we can set up proxy details for you

Best practices

- It is more convenient to designate one person in the group as environment manager and install/configure everything on their account
- Avoid clutter; it's better to create multiple single-purpose environments than one large environment with too many packages
- Be mindful of version collision when updating environments; if you don't need to update, don't
- When in doubt, contact us – we can do it for you



Consorti de
Serveis Universitaris
de Catalunya

<https://confluence.csuc.cat/display/HPCKB/Training>