A brief introduction to the use of Conda on the Trantor cluster

Conda is both an environment management system (similarly to “virtualenv”) and a package management system (similarly to “pip”). While it focuses on Python, it can also be used to install software packages related to other programming languages and software binaries.

Once created, Conda environments can be used both on head and compute nodes. It is important to note, however, that environment creation and manipulation is only possible from the head nodes.

Initializing the shell for conda

Before using conda, it is necessary to properly configure your shell with the command:

```
/cluster/shared/software/miniconda3/bin/conda init
```

For changes to take effect, you have to logout and login again. After logging in, you’ll see the prefix “(base)” at the left of the shell prompt (e.g. (base) [user@hostname ~]$). The name (or the path) inside the parentheses shows the currently active environment.

Currently available preconfigured environments

At the time of writing, the following preconfigured environments are available on the cluster:

- “**python3**” : base python 3 environment.
- “**callisto**” : a Python environment for scientific computing. It includes the following packages:
    - Python 3.8.5
    - IPython 7.16.1
    - Numpy 1.19.1
    - Pandas 1.0.5
    - Scipy 1.5.0
    - Scikit-learn 0.23.1
    - Seaborn 0.10.1
    - Matplotlib 3.2.2

You can use the ‘`conda list`’ command to get a detailed list of all the packages installed in the *currently active environment*.

Apart from “base”, all the other preconfigured Conda environments are stored under `/cluster/shared/software/conda_envs/` .

Activating a different environment

You can activate a different environment by means of the ‘conda activate’ command:

```
conda activate <environment name or path>
```

The ‘conda activate’ command requires the short name or the full path of the environment as argument. Please note that the short name can be used only for “base” or for your custom environments (discussed in the following). In the case of preconfigured environments (such as “callisto”), you have to provide their full path.

Examples:

```
conda activate base
```

```
conda activate my_custom_env
```

```
conda activate /cluster/shared/software/conda_envs/callisto
```

Conda environments can be deactivated by means of the ‘`conda deactivate`’ command.

Activating a Conda environment in PBS Jobs

Note
You can find an introduction to PBS at the following web page:
https://hpccenter.sns.it/page/wiki/pages/Submitting_Inspecting_and_Cancelling_PBS_Jobs.html
If you are not familiar with PBS, be sure to read it first!

To run a Python or an R script bound to a specific Conda environment as a Job, you have to remember to activate the related conda environment in the job script. Eg:

```
#!/bin/bash
#PBS -l select=1:ncpus=2
#PBS -q q07daneel
#PBS -N my_job
...
conda activate /cluster/shared/software/conda_envs/callisto
python my_python_script.py
```

There is however a complication: when the Job is submitted in non-interactive mode (the common case), the *.bashrc* file is not being automatically “*sourced*”[1] by the bash environment in which the job script is executed. Since “*conda init*” writes in *.bashrc* the

[1] https://linuxize.com/post/bash-source-command/

commands required to configure the shell for Conda usage, **the "*conda activate*" command will fail**.

To solve, it is sufficient to "*source*" the *.bashrc* file at the beginning of the job script, as in the following example:

```
#!/bin/bash
#PBS -l select=1:ncpus=2
#PBS -q q07daneel
#PBS -N my_job
source ~/.bashrc
conda activate /cluster/shared/software/conda_envs/callisto
python my_python_script.py
```

User-defined environments

Although the preconfigured environments can't be modified by regular users, you can create custom virtual environments according to your specific needs. The files related to these environments will be stored in your home directory. It follows that your custom environments will be available both on head and compute nodes.

The following subsections describe the steps required respectively for creating, modifying and removing a virtual environment with Conda.

[Optional] Setting conda-forge as primary channel

It is recommended to set "**conda-forge**" as the primary channel for searching and downloading packages, since it provides considerably more updated software with respect to the default channels.

> **Note**
> Setting **conda-forge** as the primary channel is **mandatory** if you want to create an **R** environment that can be used within JupyterLab!

To do so, follow these steps:

1. Change the working directory to your home:

   ```
   cd
   ```

2. Exit from any active conda environment by executing (multiple time if needed) the following command:

   ```
   conda deactivate
   ```

3. Adds conda-forge to the top of the channel list:

   ```
   conda config --add channels conda-forge
   ```

```
conda config --set channel_priority strict
```

Note
Although is it possible to exploit conda-forge only to install specific packages (i.e. by using the `--channel` option of the the `conda install` command), this may lead sometime to incompatibilities between the packages downloaded from conda-forge and the ones retrieved from the default channels. Refer to the following Web page for a detailed explanation of the problem and possible solutions:
https://conda-forge.org/docs/user/tipsandtricks.html#using-multiple-channels

Creating a custom Python-based virtual environment

1. Change the working directory to your home:

   ```
   cd
   ```

2. Create a new Conda environment with the following command:

   ```
   conda create -n my_env_name python=x.y
   ```

   where "*my_env_name*" must be replaced with the name of the new environment, and the "*x.y*" must be replaced with the desired Python version.

3. Activate the new environment:

   ```
   conda activate my_env_name
   ```

4. Install the desired packages (e.g. numpy, matplotlib etc.) :

   ```
   conda install numpy matplotlib …
   ```

Creating a custom R-based virtual environment

1. Change the working directory to your home:

   ```
   cd
   ```

2. Create a new Conda environment with the following command:

   ```
   conda create -n my_env_name
   ```

   where "*my_env_name*" must be replaced with the name of the new environment.

3. Activate the new environment:

   ```
   conda activate my_env_name
   ```

4. Install the packages **r-base** and **r-essentials** :

   ```
   conda install r-base r-essentials
   ```

5. Install any other desired package :

   ```
   conda install package_name_1 package_name_2 …
   ```

Modifying a custom virtual environment

1. Change the working directory to your home:

   ```
   cd
   ```

2. Activate the environment you wish to modify (if needed):

   ```
   conda activate my_env_name
   ```

3. Install or remove packages according to your needs :

   ```
   conda install package_name_1 package_name_2 …
   ```

   or

   ```
   conda remove package_name_1 package_name_2 …
   ```

> **Tip**
> In the case of **R** libraries, package names usually start with the “r-” prefix.

> **Tip**
> At the following Web page you can search among the packages that can be installed with the conda tool, provided by both the ufficial and third-party channels:
> https://anaconda.org/anaconda/repo

Removing a custom virtual environment

1. Change the working directory to your home:

   ```
   cd
   ```

2. Disable any previously activated environment (if needed):

   ```
   conda deactivate
   ```

3. Delete the unwanted environment:

   ```
   conda remove --name my_env_name --all
   ```

4. Print the list of the local conda environments to confirm the removal:

   ```
   conda env list
   ```

Further readings

- The official Conda documentation can be found at the following Web page: https://docs.conda.io/projects/conda/en/latest/

  In particular, the following sections discuss conda channels and their management:

  https://docs.conda.io/projects/conda/en/latest/user-guide/concepts/channels.html

  https://docs.conda.io/projects/conda/en/latest/user-guide/tasks/manage-channels.html

- A convenient cheat-sheet summarizing the principal Conda commands can be downloaded here: https://docs.conda.io/projects/conda/en/latest/user-guide/cheatsheet.html

- The official conda-forge documentation can be found at the following Web page:

  https://conda-forge.org/docs/index.html