



Conda - Reproducible Python Environments

Moritz Siegel

- ▶ `conda` is the package manager core tool.
- ▶ `miniconda` is the package we installed to provide conda.
- ▶ `anaconda` is a proprietary (unfree!) distribution of packages.
- ▶ `mamba` was a faster sister of conda - the solver is now part of conda itself.
- ▶ `micromamba` sister of miniconda.

we focus on `conda` via `miniconda`.

Start Conda

First load the **miniconda3** package with **module**:

```
1 zen trainee00@l55:~$ module load miniconda3/latest
```

Execute the **conda bash hook** to fully setup the conda tool:

```
1 zen trainee00@l55:~$ eval "$(conda shell.bash hook)"
2 (base) zen trainee00@l55:~$ conda
3 usage: conda [-h] [-v] [--no-plugins] [-V] COMMAND ...
4
5 conda is a tool for managing and deploying applications,
6 environments and packages.
7
8 ...
```

Note how the prompt now shows the **base** env.

Define Your Environment

Write a `yaml` file **defining** your environment.

```
1 name: myenv
2 channels:
3   - conda-forge
4 dependencies:
5   - python=3.10
6   - tensorflow=2.15.0
```

Use e.g. `nano myenv.yaml` to write this file.

Create your conda env:

```
1 (base) zen trainee00@l155:~$ conda env create --file myenv.yaml
2 Channels:
3 - conda-forge
4 - bioconda
5 Platform: linux-64
6 Collecting package metadata (repodata.json): /
7 ...
```

Depending of the number of packages this might take some time.

Activate your conda env, note how the **prompt** changes:

```
1 (base) zen trainee00@l55:~$ conda activate myenv
2 ...
3 (myenv) zen trainee00@l55:~$ python --version
4 Python 3.10.11
5 (myenv) zen trainee00@l55:~$ which pip
6 /home/fs70824/trainee00/.conda/envs/myenv/bin/pip
```

The python version shown is the installation from conda.

If the package you need is not available via conda, add the pip package to your environment file.

```
1 name: myenv
2 channels:
3   - conda-forge
4 dependencies:
5   - python=3.10
6   - tensor
7   - pip
8   - pip:
9     - memory-profiler
```

Note: This is not recommended since conda can not track pip's dependencies this way.

Install a package with pip into the activated **myenv**:

```
1 (myenv) zen trainee00@155:~$ pip install mypackage
2 ...
```


Submit Your Conda Env To Slurm

Create a slurm batch script starting your conda env and running python:

```
1 #!/bin/bash
2 #SBATCH --job-name=slurm_conda_example
3 #SBATCH --time=00-00:05:00
4 #SBATCH --ntasks=2
5 #SBATCH --mem=2GB
6
7 module load miniconda3/latest
8 eval "$({conda shell.bash hook})"
9 conda activate myenv
10 echo "using python: $( python --version ) from $( which python )"
11 python my_program.py
```

Use e.g. `nano myenv.sh` to write this file.

List All Environments

```
1 (base) zen trainee00@l55:~$ conda env list
2 # conda environments:
3 #
4 myenv      /home/fs70824/trainee00/.conda/envs/myenv
5 base      *  /opt/sw/conda
6 py310     /opt/sw/conda/envs/py310
7 py311     /opt/sw/conda/envs/py311
8 py312     /opt/sw/conda/envs/py312
9 py38      /opt/sw/conda/envs/py38
10 py39      /opt/sw/conda/envs/py39
```

The star * shows the currently active environment.

Deactivate Your Environment

Deactivate your conda environment to get back to (base):

```
1 (myenv) zen trainee00@155:~$ conda deactivate
2 (base) zen trainee00@155:~$ conda deactivate
3 zen trainee00@155:~$
```

Type `conda deactivate` repeatedly to **get out** of all conda envs.

Remove Your Environment

If you don't need them anymore please **remove** any old environments:

```
1 (base) zen trainee00@l55:~$ conda env remove -n myenv
2
3 Remove all packages in environment /home/fs70824/trainee00/.conda/env
4 s/myenv:
5
6 Everything found within the environment (/home/fs70824/trainee00/.con
7 da/envs/myenv), including any conda environment configurations and an
8 y non-conda files, will be deleted. Do you wish to continue?
9 (y/[n])?
```

Conda creates **many small files** and might fill up your **\$HOME** file system quickly.

- ▶ make a new empty directory **contest** to play around in.
- ▶ load **miniconda3**.
- ▶ write your own **conda.yaml** and install **numpy**.
- ▶ create a new env based on your **conda.yaml**.
- ▶ activate your env.
- ▶ find out what version of python is running there.
- ▶ submit a job to **slurm** printing the python version and using **numpy**.
- ▶ deactivate and remove your environment.

- ▶ There is more info about python, conda and jupyter at our **python4hpc** courses, for the next course visit our training calendar:

<https://vsc.ac.at/training>

- ▶ In depth **jupyter** notebooks (**ipynb**) from the python4hpc course:

https://gitlab.tuwien.ac.at/vsc-public/training/python4hpc/-/blob/6d3d51eab3eff2507b7addb501753c2fb1382683/D1_05_Conda.ipynb

- ▶ Some basic usage like this example in our documentation:

<https://docs.vsc.ac.at/software/python/conda>