LUMI

Extending containers with virtual environments for faster testing

Gregor Decristoforo – LUMI User Support Team
Norwegian research infrastructure services (NRIS) – UiT, Norway

Motivation

Cotainr is great, but building a container takes time --> not ideal for quick testing / iterating on your project

Virtual environments offer a quick (and easy) way of installing additional packages to existing containers

WARNING

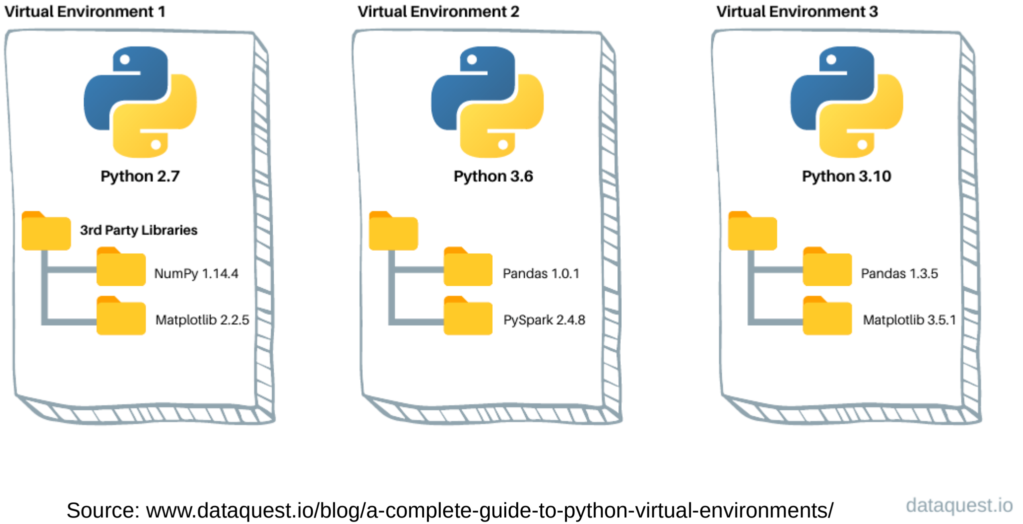
This should not be the default way of installing python packages as it puts a lot of strain

on the Lustre file system. Once you have a complete set of python packages and their versions,

always create a new container.

What are virtual environments

A virtual environment is a folder tree containing a specific Python version, third-party libraries, and other scripts.



Source: www.dataquest.io/blog/a-complete-guide-to-python-virtual-environments/

Virtual environments are conceptually similar to `conda` environments – just for `pip` only.

Requirements

We assume we already have a container built from a conda environment file. If not, we can build one via:

```
module load LUMI/23.03 cotainr
```

```
cotainr build minimal_pytorch.sif
--base-image=/appl/local/containers/sif-images/lumi-rocm-
rocm-5.6.1.sif --conda-env=minimal_pytorch.yml
```

```
name: minimal_pytorch
channels:
  - conda-forge
dependencies:
  - filelock=3.13.4
  - fsspec=2024.3.1
  - jinja2=3.1.3
  - markupsafe=2.1.5
  - mpmath=1.3.0
  - networkx=3.3
  - numpy=1.26.4
  - pillow=10.3.0
  - pip=24.0
  - python=3.11.9
  - sympy=1.12
  - typing-extensions=4.11.0
  - pip:
    - --extra-index-url
https://download.pytorch.org/whl/rocm5.6/
    - pytorch-triton-rocm==2.2.0
    - torch==2.2.2+rocm5.6
    - torchaudio==2.2.2+rocm5.6
    - torchvision==0.17.2+rocm5.6
```

Run a shell inside the container

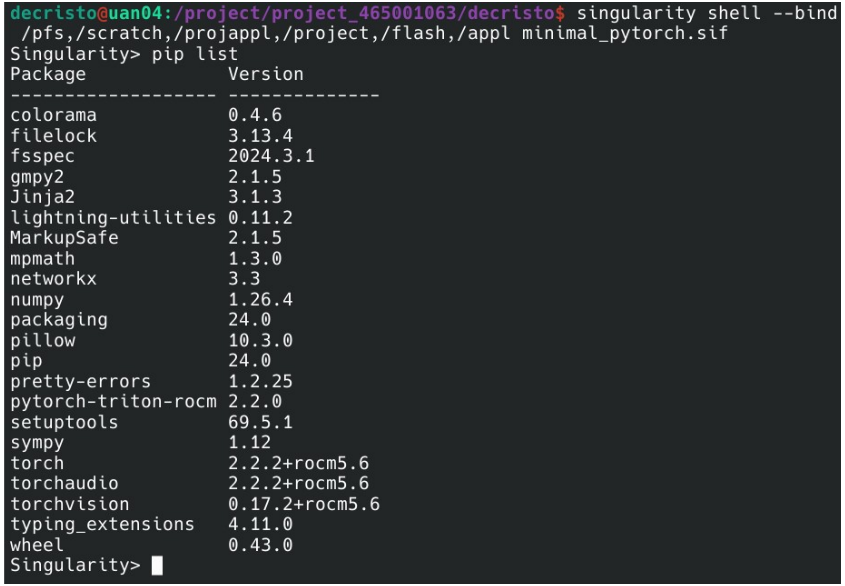
```
singularity shell --bind
/pfs,/scratch,/projappl,/project,/flash,/appl
minimal_pytorch.sif
```

Instead of setting `--bind` manually, one achieves the same with

```
module use /appl/local/training/modules/AI-
20240529/
```

```
module load singularity-userfilesystems
```

```
singularity shell minimal_pytorch.sif
```



```
decristo@uan04:/project/project_465001063/decristo$ singularity shell --bind
 /pfs,/scratch,/projappl,/project,/flash,/appl minimal_pytorch.sif
Singularity> pip list
Package             Version
------------------- -------------
colorama            0.4.6
filelock            3.13.4
fsspec              2024.3.1
gmpy2               2.1.5
Jinja2              3.1.3
lightning-utilities 0.11.2
MarkupSafe          2.1.5
mpmath              1.3.0
networkx            3.3
numpy               1.26.4
packaging           24.0
pillow              10.3.0
pip                 24.0
pretty-errors       1.2.25
pytorch-triton-rocm 2.2.0
setuptools          69.5.1
sympy               1.12
torch               2.2.2+rocm5.6
torchaudio          2.2.2+rocm5.6
torchvision         0.17.2+rocm5.6
typing_extensions   4.11.0
wheel               0.43.0
Singularity> █
```

Create a virtual environment via venv

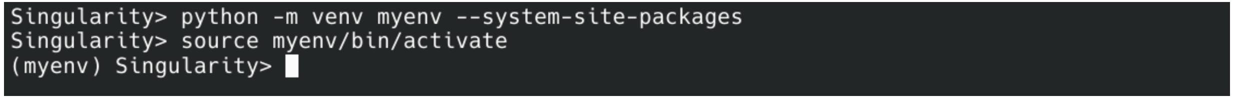
Inside the container, create a virtual environment via venv

```
python -m venv myenv --system-site-packages
```

The --system-site-packages flag gives the virtual environment access to the packages inside the container.

Activate the environment via

```
source myenv/bin/activate
```

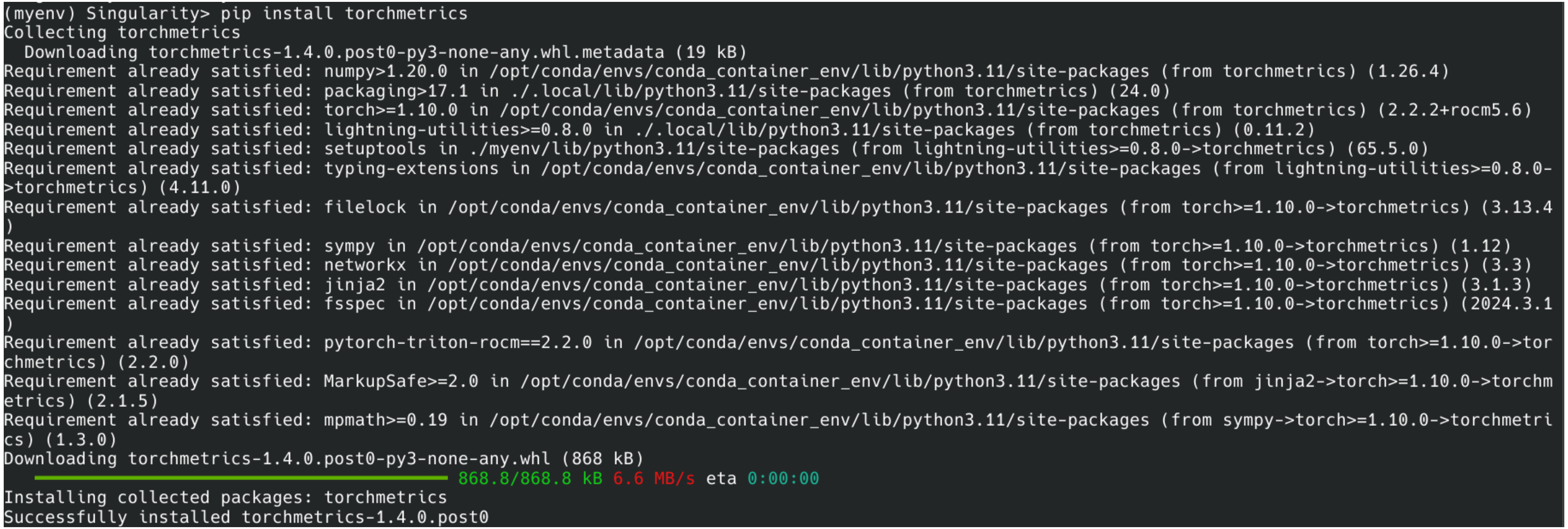


```
Singularity> python -m venv myenv --system-site-packages
Singularity> source myenv/bin/activate
(myenv) Singularity> █
```

Install custom packages via `pip`

`pip install torchmetrics`

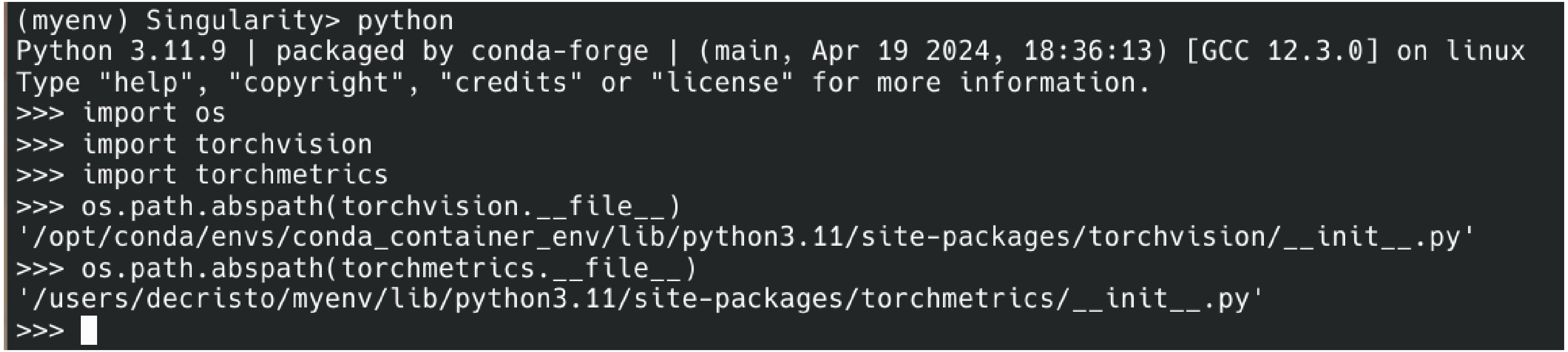
The new package will then be available alongside the packages in the container



```
(myenv) Singularity> pip install torchmetrics
Collecting torchmetrics
  Downloading torchmetrics-1.4.0.post0-py3-none-any.whl.metadata (19 kB)
Requirement already satisfied: numpy>1.20.0 in /opt/conda/envs/conda_container_env/lib/python3.11/site-packages (from torchmetrics) (1.26.4)
Requirement already satisfied: packaging>17.1 in ./.local/lib/python3.11/site-packages (from torchmetrics) (24.0)
Requirement already satisfied: torch>=1.10.0 in /opt/conda/envs/conda_container_env/lib/python3.11/site-packages (from torchmetrics) (2.2.2+rocm5.6)
Requirement already satisfied: lightning-utilities>=0.8.0 in ./.local/lib/python3.11/site-packages (from torchmetrics) (0.11.2)
Requirement already satisfied: setuptools in ./myenv/lib/python3.11/site-packages (from lightning-utilities>=0.8.0->torchmetrics) (65.5.0)
Requirement already satisfied: typing-extensions in /opt/conda/envs/conda_container_env/lib/python3.11/site-packages (from lightning-utilities>=0.8.0-
>torchmetrics) (4.11.0)
Requirement already satisfied: filelock in /opt/conda/envs/conda_container_env/lib/python3.11/site-packages (from torch>=1.10.0->torchmetrics) (3.13.4
)
Requirement already satisfied: sympy in /opt/conda/envs/conda_container_env/lib/python3.11/site-packages (from torch>=1.10.0->torchmetrics) (1.12)
Requirement already satisfied: networkx in /opt/conda/envs/conda_container_env/lib/python3.11/site-packages (from torch>=1.10.0->torchmetrics) (3.3)
Requirement already satisfied: jinja2 in /opt/conda/envs/conda_container_env/lib/python3.11/site-packages (from torch>=1.10.0->torchmetrics) (3.1.3)
Requirement already satisfied: fsspec in /opt/conda/envs/conda_container_env/lib/python3.11/site-packages (from torch>=1.10.0->torchmetrics) (2024.3.1
)
Requirement already satisfied: pytorch-triton-rocm==2.2.0 in /opt/conda/envs/conda_container_env/lib/python3.11/site-packages (from torch>=1.10.0->tor
chmetrics) (2.2.0)
Requirement already satisfied: MarkupSafe>=2.0 in /opt/conda/envs/conda_container_env/lib/python3.11/site-packages (from jinja2->torch>=1.10.0->torchm
etrics) (2.1.5)
Requirement already satisfied: mpmath>=0.19 in /opt/conda/envs/conda_container_env/lib/python3.11/site-packages (from sympy->torch>=1.10.0->torchmetri
cs) (1.3.0)
Downloading torchmetrics-1.4.0.post0-py3-none-any.whl (868 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 868.8/868.8 kB 6.6 MB/s eta 0:00:00
Installing collected packages: torchmetrics
Successfully installed torchmetrics-1.4.0.post0
```

Location of installed packages

We can check the location of the installed files via



```
(myenv) Singularity> python
Python 3.11.9 | packaged by conda-forge | (main, Apr 19 2024, 18:36:13) [GCC 12.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import os
>>> import torchvision
>>> import torchmetrics
>>> os.path.abspath(torchvision.__file__)
'/opt/conda/envs/conda_container_env/lib/python3.11/site-packages/torchvision/__init__.py'
>>> os.path.abspath(torchmetrics.__file__)
'/users/decristo/myenv/lib/python3.11/site-packages/torchmetrics/__init__.py'
>>> 
```

The new package is installed in our virtual environment whereas the other packages are installed in the container.

Cleaning up

After having found all needed packages, create a new container with an updated `conda` environment file:

```
module load LUMI/23.03 cotainr

cotainr build minimal_pytorch.sif --base-image=/appl/local/containers/sif-
  images/lumi-rocm-rocm-5.6.1.sif -conda-env=updated_environment.yml
```

The virtual environment should then be deleted:

```
rm -rf myenv
```

LUMI application containers

`venv` approach may also be used with the LUMI application containers which are built via `EasyBuild`, e.g. `/appl/local/containers/sif-images/lumi-pytorch-rocm-5.6.1-python-3.10-pytorch-v2.2.2.sif`

For these containers it is required to activate the conda environment (`$WITH_CONDA`) before creating the `venv`

Building a (final) container from LUMI application containers + a `venv` is not directly supported by `cotainr`

Pros and Cons

Pros:

- Quick (and easy) approach for installing additional packages to existing containers

Cons:

- Additional packages are installed directly on Lustre file system which can lead to bad performance
- Required to keep manually track of which `venv` matches which container for which use case
- Necessary to source the `venv` every time you run the container to get access to the packages in the virtual environment:

```
singularity exec $CONTAINER bash -c "source myenv/bin/activate && python
my_script.py"
```

Summary of steps

Open shell inside container

```
singularity shell --bind /pfs,/scratch,/projappl,/project,/flash,/appl
container_image.sif
```

If no virtual environment present, create a new one

```
python -m venv myenv --system-site-packages
```

Activate virtual environment

```
source myenv/bin/activate
```

Install custom packages

```
pip install new_package
```