



# LUMI

Extending containers  
with virtual environments  
for faster testing

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

# Motivation

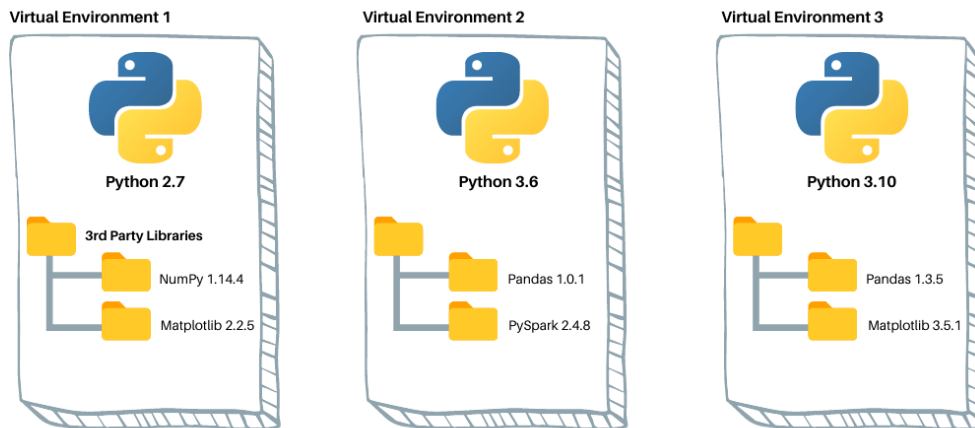


Containers is great, but building a container takes time --> not ideal for quick testing / iterating on your project

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

# 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/](http://www.dataquest.io/blog/a-complete-guide-to-python-virtual-environments/)

[dataquest.io](http://dataquest.io)

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 use /appl/local/training/modules/AI-20241126
```

```
module load cotainr
```

```
cotainr build minimal_pytorch.sif  
--base-image=/appl/local/containers/sif-images/lumi-rocm-  
rocm-6.0.3.sif --conda-env=minimal_pytorch.yml --accept-  
license
```

```
name: minimal_pytorch  
channels:  
- conda-forge  
dependencies:  
- filelock=3.15.4  
- fsspec=2024.9.0  
- jinja2=3.1.4  
- markupsafe=2.1.5  
- mpmath=1.3.0  
- networkx=3.3  
- numpy=2.1.1  
- pillow=10.4.0  
- pip=24.0  
- python=3.12.3  
- sympy=1.13.2  
- typing-extensions=4.12.2  
- pip:  
- --extra-index-url  
https://download.pytorch.org/whl/rocm6.0/  
- pytorch-triton-rocm==3.0.0  
- torch==2.4.1+rocm6.0  
- torchaudio==2.4.1+rocm6.0  
- torchvision==0.19.1+rocm6.0.6
```

# 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-20241126/  
module load singularity-userfilesystems  
singularity shell minimal_pytorch.sif
```

```
+ decristoforo singularity shell --bind /pfs,/scratch,/projappl,/project,/flash,/appl minimal  
_pytorch.sif  
Singularity> pip list  
Package Version  
-----  
filelock 3.15.4  
fsspec 2024.9.0  
gmpy2 2.1.5  
Jinja2 3.1.4  
MarkupSafe 2.1.5  
mpmath 1.3.0  
networkx 3.3  
numpy 2.1.1  
pillow 10.4.0  
pip 24.0  
pytorch-triton-rocm 3.0.0  
setuptools 75.6.0  
sympy 1.13.2  
torch 2.4.1+rocm6.0  
torchaudio 2.4.1+rocm6.0  
torchvision 0.19.1+rocm6.0  
typing_extensions 4.12.2  
wheel 0.45.1
```

# 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-1.6.0-py3-none-any.whl.metadata (20 kB)
Requirement already satisfied: numpy>1.20.0 in /opt/conda/envs/conda_container_env/lib/python3.12/site-packages (from torchmetrics) (2.1.1)
Collecting packaging>17.1 (from torchmetrics)
  Downloading packaging-24.2-py3-none-any.whl.metadata (3.2 kB)
Requirement already satisfied: torch>=2.0.0 in /opt/conda/envs/conda_container_env/lib/python3.12/site-packages (from torchmetrics) (2.4.1+rocm6.0)
Collecting lightning-utilities>=0.8.0 (from torchmetrics)
  Downloading lightning_utilities-0.11.9-py3-none-any.whl.metadata (5.2 kB)
Requirement already satisfied: setuptools in /opt/conda/envs/conda_container_env/lib/python3.12/site-packages (from lightning-utilities>=0.8.0->torchmetrics) (75.6.0)
Requirement already satisfied: typing-extensions in /opt/conda/envs/conda_container_env/lib/python3.12/site-packages (from lightning-utilities>=0.8.0->torchmetrics) (4.12.2)
Requirement already satisfied: filelock in /opt/conda/envs/conda_container_env/lib/python3.12/site-packages (from torch>=2.0.0->torchmetrics) (3.15.4)
Requirement already satisfied: sympy in /opt/conda/envs/conda_container_env/lib/python3.12/site-packages (from torch>=2.0.0->torchmetrics) (1.13.2)
Requirement already satisfied: networkx in /opt/conda/envs/conda_container_env/lib/python3.12/site-packages (from torch>=2.0.0->torchmetrics) (3.3)
Requirement already satisfied: Jinja2 in /opt/conda/envs/conda_container_env/lib/python3.12/site-packages (from torch>=2.0.0->torchmetrics) (3.1.4)
Requirement already satisfied: fsspec in /opt/conda/envs/conda_container_env/lib/python3.12/site-packages (from torch>=2.0.0->torchmetrics) (2024.9.0)
Requirement already satisfied: pytorch-triton-rocm==3.0.0 in /opt/conda/envs/conda_container_env/lib/python3.12/site-packages (from torch>=2.0.0->torchmetrics) (3.0.0)
Requirement already satisfied: MarkupSafe>=2.0 in /opt/conda/envs/conda_container_env/lib/python3.12/site-packages (from Jinja2->torch>=2.0.0->torchmetrics) (2.1.5)
Requirement already satisfied: mpmath<1.4, >=1.1.0 in /opt/conda/envs/conda_container_env/lib/python3.12/site-packages (from sympy->torch>=2.0.0->torchmetrics) (1.3.0)
Downloading torchmetrics-1.6.0-py3-none-any.whl (926 kB)
 926.4/926.4 kB 6.9 MB/s eta 0:00:00
Downloading lightning_utilities-0.11.9-py3-none-any.whl (28 kB)
Downloading packaging-24.2-py3-none-any.whl (65 kB)
 65.5/65.5 kB 2.6 MB/s eta 0:00:00
Installing collected packages: packaging, lightning-utilities, torchmetrics
Successfully installed lightning-utilities-0.11.9 packaging-24.2 torchmetrics-1.6.0
```

# Location of installed packages



We can check the location of the installed files via

```
(myenv) Singularity> python
Python 3.12.3 | packaged by conda-forge | (main, Apr 15 2024, 18:38:13) [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/python3.12/site-packages/torchvision/__init__.py'
>>> os.path.abspath(torchmetrics.__file__)
'/pfs/lustrep3/projappl/project_462000002/decrisoforo/myenv/lib/python3.12/site-packages/torchmetrics/__init__.py'
```

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



## Warning

You should not stop here, as this way of installing python packages creates typically thousands of small files. This puts a lot of strain on the Lustre file system and might exceed your file quota.

Once you have a complete set of python packages and their versions, choose one of the following options:

- Create a new container with `containr` and delete virtual environment
- Turn `myenv` into a SquashFS file and bind mount it to the container

# Option 1: Create a new container with cotainr



After having found all needed packages, add them to the conda environment file and create a new container:

```
module use /appl/local/training/modules/AI-20241126
module load cotainr

cotainr build updated_pytorch.sif
  --base-image=/appl/local/containers/sif-images/lumi-rocm-
  rocm-6.0.3.sif --conda-env=updated_pytorch.yml --accept-
  license
```

The virtual environment should then be deleted:

```
rm -rf myenv
```

```
name: updated_pytorch
channels:
  - conda-forge
dependencies:
  - filelock=3.15.4
  - fsspec=2024.9.0
  - jinja2=3.1.4
  - markupsafe=2.1.5
  - mpmath=1.3.0
  - networkx=3.3
  - numpy=2.1.1
  - pillow=10.4.0
  - pip=24.0
  - python=3.12.3
  - sympy=1.13.2
  - typing-extensions=4.12.2
  - pip:
    - --extra-index-url
      https://download.pytorch.org/whl/rocm6.0/
    - pytorch-triton-rocm==3.0.0
    - torch==2.4.1+rocm6.0
    - torchaudio==2.4.1+rocm6.0
    - torchvision==0.19.1+rocm6.0.6
    - torchmetrics==1.6.0
```

## Option 2: Turn myenv into a SquashFS file

Turn the myenv directory into a SquashFS file and bind mount it to the container:

```
mksquashfs myenv myenv.sqsh
```

```
singularity exec -B myenv.sqsh:/user-software:image-src=/ minimal_pytorch.sif  
  bash -c 'source /user-software/bin/activate && python my_script.py'
```

This is much better for the file system as it regards the myenv.sqsh as a single file.

For advanced users:

This approach is compatible with packages that cannot be installed via `cotainr` (e.g. packages that require manual compilation)

# LUMI application containers



venv approach may also be used with the LUMI application containers that are not built with `cotainr`, e.g. `/appl/local/containers/sif-images/lumi-pytorch-rocm-6.2.1-python-3.12-pytorch-20240918-vllm-4075b35.sif`

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

```
CONTAINER=/appl/local/containers/sif-images/lumi-pytorch-rocm-6.2.1-python-3.12-pytorch-20240918-vllm-4075b35.sif
```

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

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 and exceed your file limit (if SquashFS approach is not used)
- 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

## 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
```