

How-to create a simple apptainer container

Consider the case where you need to create a working python3 distrib with pytorch installed for running a script using GPU resources on ChaCha or DISCO. On those machines, we use Apptainer for the containers and this documentation describes how to create, customize and run images using Apptainer. Launching Apptainer tasks using Slurm is defined elsewhere.

Defining the content of the container

Start by defining a container definition, such as in `rules.def` (this corresponds to a Dockerfile file):

```
Bootstrap: docker
From: nvidia/cuda:12.6.2-base-ubuntu22.04

%help
  This container provides a CUDA environment, with Python and
  Jupyternotebook.

%labels
  AUTHOR_NAME Pierre-André Mudry
  VERSION 1.0

%environment

%post -c /bin/bash
  export DEBIAN_FRONTEND=noninteractive
  apt-get -y update
  apt-get -y install git python3-pip python3-dev python3-opencv libglib2.0-0

%files
  requirements.txt requirements.txt

%post -c /bin/bash
  python3 -m pip install -r requirements.txt
  python3 -m pip install --upgrade pip
  pip3 install torch torchvision torchaudio -f
  https://download.pytorch.org/whl/cu111/torch_stable.html
```

This definition files is of a Debian distrib with CUDA, with python3 and pytorch installed and has other requirements as well. Those requirements are copied from you current directory `requirements.txt` file using the `%files` directive. In this example, we use for `requirements.txt`:

```
warp-lang
usd-core
matplotlib
```

```
pyglet
jupyter
python-language-server
```

Of course, you can tailor this to fit your needs.

Check the [complete Apptainer documentation](#) for more details.

Building the image

Once the definition is made, you can build the `test.sif` image with `apptainer build test.sif rules.def`. The image is created in your own directory. By default, when you run the image, it has access to the host filesystem. To run the image interactively (or not), you then simply run

Running the image

```
apptainer run --nv ./test.sif
```

The `--nv` flag must be used to use NVidia resources. If you run this script interactively, you then have access to the virtual environment. You can launch for instance `nvidia-smi` or `jupyter notebook --ip=0.0.0.0 --port=8888 --no-browser --allow-root` to create a Jupyter notebook system.

Migrating from Docker

When you already have a Docker image for your job, you have several options to migrate to Apptainer :

- You can convert your image to an Apptainer `.sif` on your local laptop before uploading your `.sif` to the server :

1. From local docker images :

```
sudo apptainer build yourapplication.sif docker-
daemon://yourapplication:latest

[sudo] password for user.name:
INFO: Starting build...
Copying blob 27123a71e85e done |
Copying config 388ea0fa6e done |
Writing manifest to image destination
2025/03/03 10:43:58 info unpack layer:
sha256:bec6fac0671f72331024c0537184f7213dd7940aa9f6bf20cb8a30bdb2e35b1d
INFO: Creating SIF file...
INFO: Build complete: yourapplication.sif
```

```
scp yourapplication.sif user.name@chacha:/home/user.name/datasets/
```

- You can build the .sif from a docker archive :

```
sudo docker save yourapplication:latest -o yourapplication.tar
```

```
sudo chown user.name: yourapplication.tar
```

```
apptainer build yourapplication.sif docker-archive://yourapplication.tar
```

```
INFO: Starting build...
```

```
Copying blob 27123a71e85e done |
```

```
Copying config 388ea0fa6e done |
```

```
Writing manifest to image destination
```

```
2025/03/03 10:54:47 info unpack layer:
```

```
sha256:bec6fac0671f72331024c0537184f7213dd7940aa9f6bf20cb8a30bdb2e35b1d
```

```
INFO: Creating SIF file...
```

```
INFO: Build complete: yourapplication.sif
```

```
scp yourapplication.sif user.name@chacha:/home/user.name/datasets/
```

From:

<https://wiki.isc-vs.ch/> - **The ISC wiki**

Permanent link:

https://wiki.isc-vs.ch/doku.php?id=infra:howto:apptainer_sample&rev=1740996083

Last update: **2025/03/03 10:01**

