Jobs@IP2I : using containers









What is a container Creating a container Using a container



3rd March 2022



A container is a set of softwares and libraries directly interacting with the host kernel

HOST	
Container 2	Container 3
<ul> <li>Debian 11</li> <li>Python 3.6</li> <li>GCC 9</li> </ul>	<ul> <li>Ubuntu 18</li> <li>TensorFlow 2.4</li> <li>Python 3.6</li> </ul>
LINUX KERNEL	
	HOST Container 2 . Debian 11 . Python 3.6 . GCC 9 LINUX KERNEL



A container is created from an *image*. With Docker, we can either use an existing image (https://hub.docker.com/) or create our own.

We currently have 2 existing images on the computing group gitlab repository https://gitlab.in2p3.fr/ip2i/calcul/docker\_images

The 2 images are available from the CC IN2P3 container registry :

- gitlab-registry.in2p3.fr/ip2i/calcul/docker\_images:cosmo
- gitlab-registry.in2p3.fr/ip2i/calcul/docker\_images:tensorflow

They are documented on https://ip2i.pages.in2p3.fr/calcul/documentation/ML/IP2I\_GPU\_Server/#using-sing ularity



We have docker images ... but docker is not installed on the SLURM cluster!

We have *singularity*, which can use docker images and is less permissive in a shared environment

## Conversion of a docker image to a singularity image :

export SINGULARITY\_TMPDIR=/scratch/ export SINGULARITY\_CACHEDIR=/scratch/ singularity build ./<image name>.sif docker://<address>/<image name>:<tag>

Existing singularity images are stored in /gridgroup/calcul/singularity/

Creation of a container from a singularity image :

singularity run *path/to/image.sif* or singularity exec path/to/image.sif ./myscript.sh

If you need access to a GPU card :

export SINGULARITYENV\_CUDA\_VISIBLE\_DEVICES=\$CUDA\_VISIBLE\_DEVICES singularity run --nv /path/to/image/tensorflow.sif

3rd March 2022