Containers in HPC – Pyxis

UManitoba Fall 2025
High-Performance Computing And Cloud Workshop

Stefano Ansaloni

University of Manitoba

October 15, 2025

About me

Stefano Ansaloni

Cloud Computing Specialist at University of Manitoba
(part of the Advanced Research Computing team)

Software Developer and DevOps Specialist since 2017

Linux User/Admin since 2005

What is a container

From Wikipedia (Containerization):

> *Containerization is operating-system–level virtualization or application-level virtualization so that softwares can run in isolated user spaces called "containers" in any cloud or non-cloud environment, regardless of type or vendor.*

Properties of containers

- Each container is a fully functional and portable computing environment surrounding the application and keeping it independent of other environments running in parallel
- Each container simulates a different software application and runs isolated processes (including configurations, libraries, and dependencies)
- Multiple containers share a common operating system kernel (operative system)

Terminology

Image	Archive (or number of archives – i.e. “layers”) of a filesystem tree along with metadata
Containerfile	Recipe for building an image, including OS and software within the image (e.g. *Dockerfile*)
Container	A running instance of an image (can be a computing process, or a service daemon)
Container Runtime	Lower level component responsible for reading the image and communicating with the host kernel to start containerized processes (e.g. *runc*, *crun*)
OCI (Open Container Initiative)	Open governance structure that creates open industry standards for container formats and runtimes
Registry	An online storage area for images (e.g. *DockerHub*, *Quay.io*)

Focusing On Pyxis

What is Pyxis

From GitHub (NVIDIA/pyxis):

> *Pyxis is a SPANK plugin for the Slurm Workload Manager. It allows unprivileged cluster users to run containerized tasks.*
>
> *A pyxis is an ancient small box or container.*

Features

Pyxis is currently only available on Grex, providing the following benefits:

- Seamlessly execute Slurm jobs in unprivileged containers
- Simple command-line interface
- Support for OCI image registries
- Support for layers caching and layers sharing across nodes
- Supports multi-node MPI jobs through PMI2 or PMIx
- Allows users to install packages inside the container

Command line options

Pyxis introduces some new command line options to the Slurm default job submission tools (i.e. *salloc*, *sbatch*, *srun*):

`--container-image=IMAGE[:TAG]`	Image to use for the container filesystem
`--container-mounts=SRC:DST[:OPTS][,SRC:DST]`	Bind mount(s) directories inside the container
`--container-workdir=PATH`	Working directory inside the container
`--container-remap-root`	Ask to be remapped to root inside the container
`--container-entrypoint`	Execute the entrypoint from the container image
`--container-env=NAME[,NAME]`	Names of environment variables to override with the host environment

The full list of options can be printed using the "`--help`" flag.

Example

Using `salloc`

To submit a test job using “`salloc`” and Pyxis, it is enough to specify the “`--container-image`” option:

```
salloc --partition=skylake --container-image=debian:stable-slim
```

When using Pyxis through “`salloc`”, Slurm will start an interactive job and return a shell from inside the container.

Example

Using sbatch

```
#!/bin/bash

#SBATCH --partition=lgpu
#SBATCH --nodes=1
#SBATCH --gpus=2
#SBATCH --ntasks=4
#SBATCH --cpus-per-task=8
#SBATCH --mem-per-cpu=8G
#SBATCH --container-image=nvcr.io/hpc/gromacs:2023.2
#SBATCH --container-mounts=/PATH/TO/INPUT:/host_pwd
#SBATCH --container-workdir=/host_pwd
#SBATCH --container-entrypoint

gmx mdrun -ntmpi 4 -ntomp 8 -nb gpu -pme gpu -npme 1 -update gpu -bonded gpu \
    -nsteps 100000 -resetstep 90000 -noconfout -dlb no -nstlist 300 -pin on \
    -v -gpu_id 01
```

University of Manitoba

Demo

Running a GROMACS benchmark with Pyxis

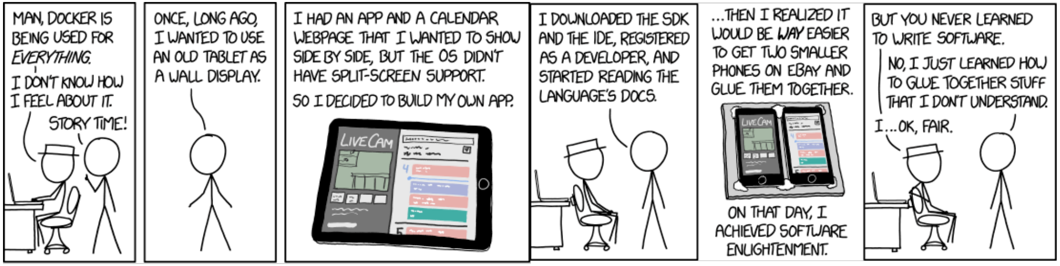
Running a GROMACS benchmark with Pyxis

Specifications

Container Engine	⇒	Pyxis/Enroot (on Grex)
Image	⇒	nvcr.io/hpc/gromacs:2023.2 (from the nVidia NGC Catalog)
Software	⇒	GROMACS
Benchmark	⇒	STMV

Is Pyxis a silver bullet?

- Always check if the same software is already provided via modules-based HPC software stack
- It requires *well-built* images
  - Making or finding a suitable image is a bit of work
  - Bleeding-edge versions of programs could be poorly maintained/tested (including their images)
- Useful to encapsulate software and sometimes data to reduce number of files (e.g. python or conda based programs)


MAN, DOCKER IS BEING USED FOR *EVERYTHING*.
I DON'T KNOW HOW I FEEL ABOUT IT.
STORY TIME!
ONCE, LONG AGO, I WANTED TO USE AN OLD TABLET AS A WALL DISPLAY.
I HAD AN APP AND A CALENDAR WEBPAGE THAT I WANTED TO SHOW SIDE BY SIDE, BUT THE OS DIDN'T HAVE SPLIT-SCREEN SUPPORT.
SO I DECIDED TO BUILD MY OWN APP.
LIVE CAM
I DOWNLOADED THE SDK AND THE IDE, REGISTERED AS A DEVELOPER, AND STARTED READING THE LANGUAGE'S DOCS.
...THEN I REALIZED IT WOULD BE *WAY* EASIER TO GET TWO SMALLER PHONES ON EBAY AND GLUE THEM TOGETHER.
ON THAT DAY, I ACHIEVED SOFTWARE ENLIGHTENMENT.
BUT YOU NEVER LEARNED TO WRITE SOFTWARE.
NO, I JUST LEARNED HOW TO GLUE TOGETHER STUFF THAT I DON'T UNDERSTAND.
I...OK, FAIR.

XKCD [1988]

Questions?

Thank you

University of Manitoba