

LUMI

A white wolf is the central focus, standing in a futuristic, blue-toned digital environment. The background is filled with vertical data streams, grid patterns, and glowing particles, creating a high-tech, cybernetic atmosphere. The wolf is looking slightly to the right of the viewer.

Containers on LUMI-C and LUMI-G

Kurt Lust
LUMI User Support Team (LUST)
University of Antwerp

April 2026

Containers

This is about containers on LUMI-C and LUMI-G, not LUMI-K!

- What can they do and what can't they do?
- Getting containers onto LUMI
- Running containers on LUMI
- Enhancements to the LUMI environment to help you
- Strategies for extending existing containers

- But remember: LUMI is an HPC infrastructure, not a container cloud!
 - HPC has its own container runtimes specifically for an HPC environment

What do containers not provide?

- **Full reproducibility of your science** is a myth
 - Only reproducibility of part of the software stack, not of the results
- **Performance portability:**
 - A container built from sources on one CPU will not be optimal for another one.
 - Containers built from downloaded binaries may not exploit all architectural features of the CPU.
 - No support for the LUMI interconnect may lead to fall-back to a slower protocol that works
- **Simply portability:** Not every container prepared on your Ubuntu or CentOS cluster or workstation will work on LUMI.
 - Containers that rely on certain hardware, drivers/kernel modules and/or kernel versions may fail.
 - Problem cases: GPU (driver version) and sometimes high-performance networking (MPI, RCCL, ...)

But what can they then do on LUMI?

L U M I

- **Containers are a software management instrument** to create a contained software installation
- **Storage manageability:** Lower pressure on the filesystems (for software frameworks that access hundreds of thousands of small files) for better I/O performance and management of your disk file quota.
 - E.g., conda installations are not appreciated straight on the Lustre file system
- **Software installation:** Can be a way to install software with an installation process that is not aware of multi-user HPC systems and is too complicated to recompile.
 - E.g., GUI applications that need a fat library stack
 - E.g., experiment with software that needs a newer version of ROCm, though with limitations
- **Isolation:** More important for services; often a pain instead
 - But still helps avoiding picking up the wrong libraries
- **But note:** You're the system administrator of your container, not LUST!

Profiling can reveal things we forget

- We were experimenting with Cray performance tools profiling the LUMI [MNIST image example](#)
- Using a venv for pytorch torchvision torchaudio with rocm 6.0 support
- Run on 4 nodes 32 GPUs
- Looking at the profile most of the I/O was in python files and read/open/dlopen
- Moved to putting the venv into the container:
Runtime changed from 100s to 52s

Managing containers

- Supported runtimes
 - Not all container runtimes are a good match with HPC systems
 - No elevated privileges for security reasons
 - No user namespaces on LUMI because of security issues in combination with networked filesystems
 - Docker is **NOT** directly available in the user environment (and will never be)
 - Singularity Community Edition is natively available (as a system command) on the login and compute nodes

Managing containers (2)

- But you can convert docker containers to singularity: Pulling containers
 - DockerHub and other registries (example: Julia container)
`singularity pull docker://julia`
 - Singularity uses a flat (single) sif file for storing the container and the pull command makes the conversion
 - Be carefull: cache in `.singularity` dir can easily exhaust your storage quota for larger images
 - May want to set `SINGULARITY_CACHEDIR` to move the cache

```
kulust@uan03.lumi.csc - ~/container-demo
kulust@uan03.lumi.csc - ~/container-demo (ssh) #1
[lumi][kulust@uan03-1016 container-demo]$ singularity pull docker://julia
INFO:   Converting OCI blobs to SIF format
INFO:   Starting build...
INFO:   Fetching OCI image...
8.8MiB / 8.8MiB [=====] 100 % 28.7 MiB/s 0s
28.4MiB / 28.4MiB [=====] 100 % 28.7 MiB/s 0s
280.0MiB / 280.0MiB [=====] 100 % 28.7 MiB/s 0s
INFO:   Extracting OCI image...
2026/04/16 18:56:06 warn rootless{usr/local/julia/lib/julia/libLLVM-18jl.so} ignoring (usually) harmless
EPERM on setxattr "user.rootlesscontainers"
2026/04/16 18:56:06 warn rootless{usr/local/julia/lib/julia/libLLVM.so} ignoring (usually) harmless EPERM
on setxattr "user.rootlesscontainers"
2026/04/16 18:56:08 warn rootless{usr/local/julia/lib/julia/libamd.so} ignoring (usually) harmless EPERM
on setxattr "user.rootlesscontainers"
2026/04/16 18:56:08 warn rootless{usr/local/julia/lib/julia/libamd.so.3} ignoring (usually) harmless EPERM
M on setxattr "user.rootlesscontainers"
2026/04/16 18:56:08 warn rootless{usr/local/julia/lib/julia/libatomic.so} ignoring (usually) harmless EPE
RM on setxattr "user.rootlesscontainers"
2026/04/16 18:56:08 warn rootless{usr/local/julia/lib/julia/libatomic.so.1} ignoring (usually) harmless E
PERM on setxattr "user.rootlesscontainers"
2026/04/16 18:56:08 warn rootless{usr/local/julia/lib/julia/libblastrampoline.so} ignoring (usually) harm
less EPERM on setxattr "user.rootlesscontainers"
2026/04/16 18:56:08 warn rootless{usr/local/julia/lib/julia/libblastrampoline.so.5.15.0} ignoring (usuall
```

```
kulust@uan03.lumi.csc - ~/container-demo ㉿#2
kulust@uan03.lumi.csc - ~/container-demo (ssh) #1 +
2026/04/16 18:56:22 warn rootless{usr/local/julia/share/julia/test/trimming/basic_jll.jl} ignoring (usually) harmless EPERM on setxattr "user.rootlesscontainers"
2026/04/16 18:56:22 warn rootless{usr/local/julia/share/julia/test/trimming/hello.jl} ignoring (usually) harmless EPERM on setxattr "user.rootlesscontainers"
2026/04/16 18:56:22 warn rootless{usr/local/julia/share/julia/test/trimming/trimmability.jl} ignoring (usually) harmless EPERM on setxattr "user.rootlesscontainers"
2026/04/16 18:56:22 warn rootless{usr/local/julia/share/julia/test/trimming/trimming.jl} ignoring (usually) harmless EPERM on setxattr "user.rootlesscontainers"
2026/04/16 18:56:22 warn rootless{usr/local/julia/share/julia/test/triplequote.jl} ignoring (usually) harmless EPERM on setxattr "user.rootlesscontainers"
2026/04/16 18:56:22 warn rootless{usr/local/julia/share/julia/test/tuple.jl} ignoring (usually) harmless EPERM on setxattr "user.rootlesscontainers"
2026/04/16 18:56:22 warn rootless{usr/local/julia/share/julia/test/unicode/utf8.jl} ignoring (usually) harmless EPERM on setxattr "user.rootlesscontainers"
2026/04/16 18:56:22 warn rootless{usr/local/julia/share/julia/test/vecelement.jl} ignoring (usually) harmless EPERM on setxattr "user.rootlesscontainers"
2026/04/16 18:56:22 warn rootless{usr/local/julia/share/julia/test/version.jl} ignoring (usually) harmless EPERM on setxattr "user.rootlesscontainers"
2026/04/16 18:56:22 warn rootless{usr/local/julia/share/julia/test/worlds.jl} ignoring (usually) harmless EPERM on setxattr "user.rootlesscontainers"
INFO:      Inserting Singularity configuration...
INFO:      Creating SIF file...
[lumi][kulust@uan03-1017 container-demo]$
```

```
kulust@uan03.lumi.csc - ~/.singularity
kulust@uan03.lumi.csc - ~/.singularity (ssh) #1
2026/04/16 18:56:22 warn rootless{usr/local/julia/share/julia/test/worlds.jl} ignoring (usually) harmless
EPERM on setattr "user.rootlesscontainers"
INFO:   Inserting Singularity configuration...
INFO:   Creating SIF file...
[lumi][kulust@uan03-1017 container-demo]$ cd ~/.singularity
[lumi][kulust@uan03-1018 .singularity]$ ls -la
total 12
drwx----- 3 kulust pepr_kulust 4096 Apr 16 18:55 .
drwx----- 41 kulust pepr_kulust 4096 Apr 16 16:12 ..
drwx----- 9 kulust pepr_kulust 4096 Apr 16 18:55 cache
[lumi][kulust@uan03-1019 .singularity]$ du -h
4.0K    ./cache/shub
318M    ./cache/blob/blobs/sha256
318M    ./cache/blob/blobs
318M    ./cache/blob
4.0K    ./cache/net
4.0K    ./cache/oras
4.0K    ./cache/oci-sif
4.0K    ./cache/library
312M    ./cache/oci-tmp
629M    ./cache
629M    .
[lumi][kulust@uan03-1020 .singularity]$
```

Managing containers (3)

- Building containers
 - Support for building containers is very limited on LUMI: No elevated privileges but also no user namespaces and no fakeroot.
 - One option is to pull or copy containers from outside.
 - We can support **proot** though and SingularityCE has a build process based on this. With that, singularity can build from existing (base) container in some cases.
 - Build type called “[Unprivileged proot builds](#)” in the Singularity CE manual.
 - Needs **proot** from the **PRoot** module in CrayEnv or the LUMI stacks.
 - We provide some base images adapted for LUMI.

Interacting with containers

- Accessing a container with the `shell` command
`singularity shell container.sif`

singularity shell julia_latest.sif

LUMI



A terminal window titled "kulust@uan03.lumi.csc - ~/container-demo" with a terminal address bar showing "kulust@uan03.lumi.csc - ~/container-demo (ssh) #1". The terminal output shows the following sequence of commands and results:

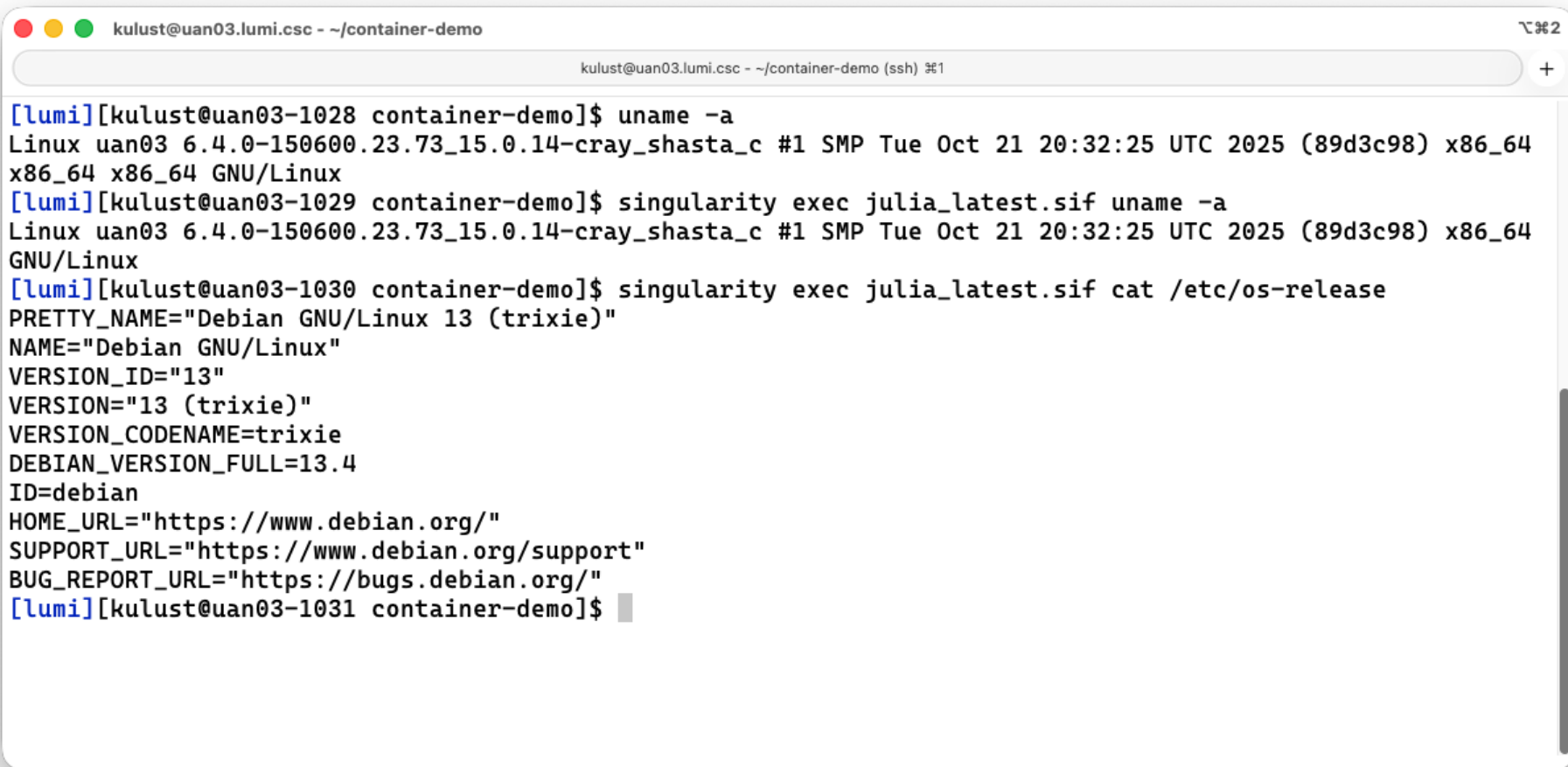
```
[lumi][kulust@uan03-1025 container-demo]$ ls /opt
admin-pe AMD amdgpu cray e-sms modulefiles rocm rocm-6.3.4 slingshot
[lumi][kulust@uan03-1026 container-demo]$ singularity shell julia_latest.sif
Singularity> ls /opt
Singularity> cat /etc/os-release
PRETTY_NAME="Debian GNU/Linux 13 (trixie)"
NAME="Debian GNU/Linux"
VERSION_ID="13"
VERSION="13 (trixie)"
VERSION_CODENAME=trixie
DEBIAN_VERSION_FULL=13.4
ID=debian
HOME_URL="https://www.debian.org/"
SUPPORT_URL="https://www.debian.org/support"
BUG_REPORT_URL="https://bugs.debian.org/"
Singularity> exit
exit
[lumi][kulust@uan03-1027 container-demo]$
```

Interacting with containers

- Accessing a container with the `shell` command
`singularity shell container.sif`
- Executing a command in the container with `exec`
`singularity exec container.sif uname -a`

singularity exec julia_latest.sif uname -a

LUMI



```
kulust@uan03.lumi.csc - ~/container-demo
kulust@uan03.lumi.csc - ~/container-demo (ssh) #1
[lumi][kulust@uan03-1028 container-demo]$ uname -a
Linux uan03 6.4.0-150600.23.73_15.0.14-cray_shasta_c #1 SMP Tue Oct 21 20:32:25 UTC 2025 (89d3c98) x86_64
x86_64 x86_64 GNU/Linux
[lumi][kulust@uan03-1029 container-demo]$ singularity exec julia_latest.sif uname -a
Linux uan03 6.4.0-150600.23.73_15.0.14-cray_shasta_c #1 SMP Tue Oct 21 20:32:25 UTC 2025 (89d3c98) x86_64
GNU/Linux
[lumi][kulust@uan03-1030 container-demo]$ singularity exec julia_latest.sif cat /etc/os-release
PRETTY_NAME="Debian GNU/Linux 13 (trixie)"
NAME="Debian GNU/Linux"
VERSION_ID="13"
VERSION="13 (trixie)"
VERSION_CODENAME=trixie
DEBIAN_VERSION_FULL=13.4
ID=debian
HOME_URL="https://www.debian.org/"
SUPPORT_URL="https://www.debian.org/support"
BUG_REPORT_URL="https://bugs.debian.org/"
[lumi][kulust@uan03-1031 container-demo]$
```

Interacting with containers

- Accessing a container with the `shell` command
`singularity shell container.sif`
- Executing a command in the container with `exec`
`singularity exec container.sif uname -a`
- "Running" a container
`singularity run container.sif`
will usually execute a default command and pass its arguments to that command
- Inspecting run definition script
`singularity inspect --runscript container.sif`

Interacting with containers

- Accessing a container with the `shell` command
`singularity shell container.sif`
- Executing a command in the container with `exec`
`singularity exec container.sif uname -a`
- "Running" a container
`singularity run container.sif`
- Inspecting run definition script
`singularity inspect --runscript container.sif`
- Accessing host filesystem with bind mounts
 - Singularity will mount `$HOME`, `/tmp`, `/proc`, `/sys`, `/dev` into container by default
 - Use `--bind src1:dest1,src2:dest2` or the `SINGULARITY_BIND(PATH)` environment variable to mount other host directories (like `/project` or `/appl`).
On LUMI you need `--bind /pfs,/scratch,/projappl,/project,/flash`

Running containers on LUMI

- Use SLURM to run containers on compute nodes
- Use srun to execute MPI containers

```
srun singularity exec --bind ${BIND_ARGS} \  
${CONTAINER_PATH} my_mpi_binary ${APP_PARAMS}
```
- **Be aware your container must be compatible with Cray MPI** (MPICH ABI compatible) or use the proper libfabric library with libcx1 for good performance
 - Configure suggestion: see next slide
- Open MPI based containers may need workarounds and are not well supported on LUMI at the moment (and even more problematic for the GPU)

Environment enhancements (1)

- LUMI specific tools for container interaction provided as modules
- **singularity-bindings/system** (available via easyconfig)
 - Sets the environment to use Cray MPICH provided outside the container
 - Requires a LUMI software stack
 - Use EasyBuild-user module and `eb --search singularity-bindings` to find the easyconfig or copy from our [LUMI Software Library web site](#)
 - Provides basic bind mounts for using the host MPI in the container setting `SINGULARITY_BIND` and `SINGULARITY_LD_LIBRARY_PATH`
 - May need customisation for your specific container

Environment enhancements (2)

- **lumi-aif-singularity-bindings** (available via [Local-LAIF](#) module or [/appl/local/laifs/modules](#))
 - Bindings for the AI containers provided by the LUMI AI Factory
 - Provide all your filesystems and a directory needed for Slurm support
 - Sets an environment variable so that its MPI can talk to Slurm
- **singularity-AI-bindings** (easyconfig or [/appl/local/containers/ai-modules](#))
 - Bindings for some of the AI containers that LUST provided
 - But not a generic binding that will work for all containers!

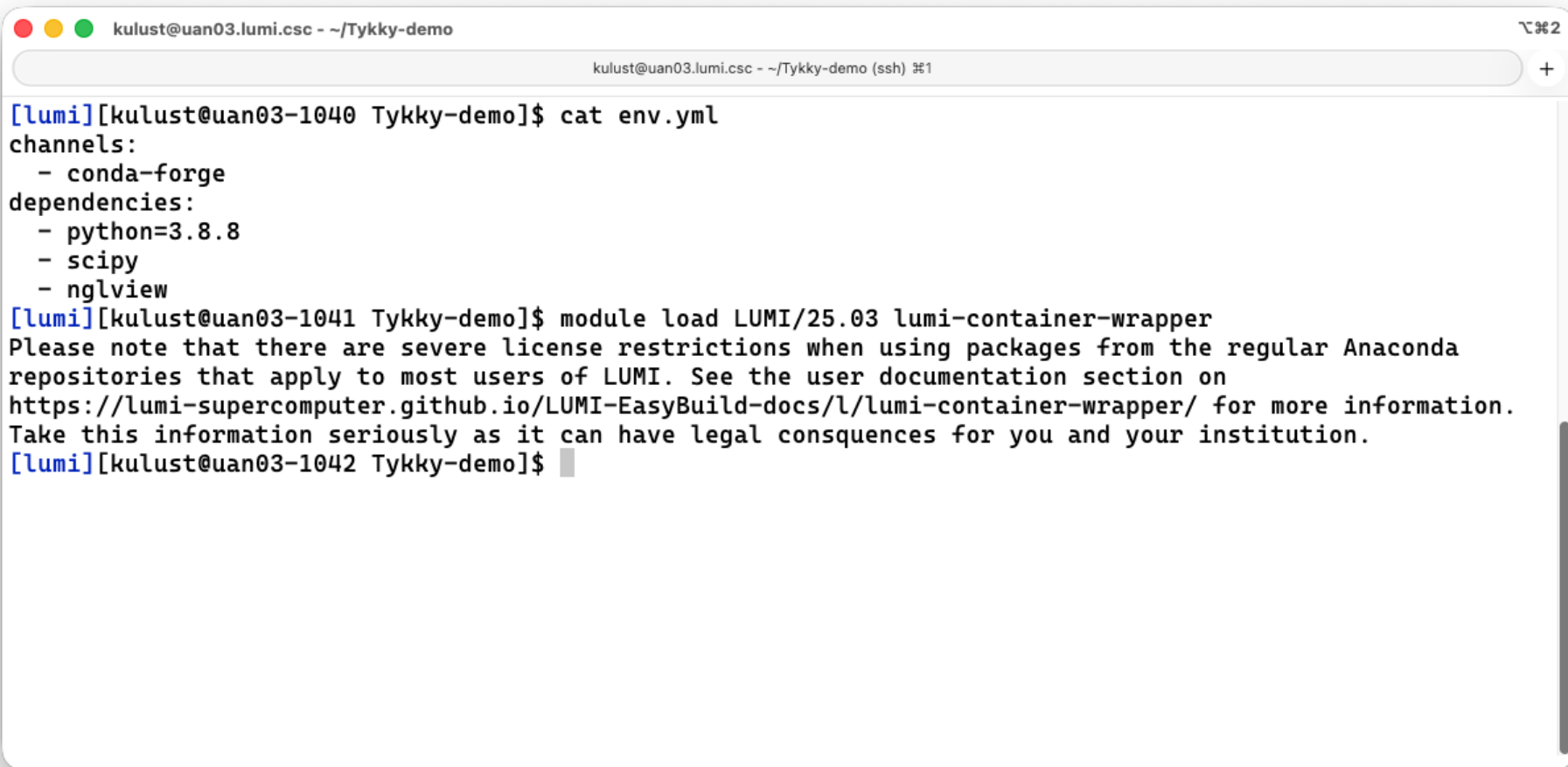
Environment enhancements (3)

Containerising tools

- **cotainr** (LUMI and CrayEnv software stacks)
 - A tool to pack conda installations in a singularity container
 - Use the singularity commands as shown on earlier slides to run
- **lumi-container-wrapper** (LUMI and CrayEnv software stacks)
 - Supports conda and pip environments
 - With pip: Python provided by the cray-python module (so there is an optimised NumPy etc.)
 - Software installation in two parts: a base container and a SquashFS file which is mounted in that container with the conda/pip environment
 - Provides wrappers to encapsulate your custom environment in a container (so you don't use singularity commands directly)
 - Can even create wrappers for commands in an existing container
 - Still helps with quota on the number of files in your project and I/O performance

lumi-container-wrapper (1)

L U M I



```
kulust@uan03.lumi.csc - ~/Tykky-demo
kulust@uan03.lumi.csc - ~/Tykky-demo (ssh) #1
[lumi][kulust@uan03-1040 Tykky-demo]$ cat env.yml
channels:
- conda-forge
dependencies:
- python=3.8.8
- scipy
- nglview
[lumi][kulust@uan03-1041 Tykky-demo]$ module load LUMI/25.03 lumi-container-wrapper
Please note that there are severe license restrictions when using packages from the regular Anaconda
repositories that apply to most users of LUMI. See the user documentation section on
https://lumi-supercomputer.github.io/LUMI-EasyBuild-docs/l/lumi-container-wrapper/ for more information.
Take this information seriously as it can have legal consequences for you and your institution.
[lumi][kulust@uan03-1042 Tykky-demo]$
```

lumi-container-wrapper (2)

```
kulust@uan03.lumi.csc - ~/Tykky-demo
kulust@uan03.lumi.csc - ~/Tykky-demo (ssh) #1
[lumi][kulust@uan03-1042 Tykky-demo]$ conda-containerize new --prefix ./conda-cont-1 env.yml
[ INFO ] Constructing configuration
[ INFO ] Using /tmp/kulust/cw-4TASAY as temporary directory
[ INFO ] Installation dir ./conda-cont-1 does not exist, creating it for you
[ INFO ] Fetching container docker://opensuse/leap:15.6
[ INFO ] Running installation script
[ INFO ] Using miniforge version Miniforge3-26.1.1-3-Linux-x86_64
[ INFO ] Installing miniforge
=====
Linking mamba-2.5.0-h9835478_0
Linking ruamel.yaml-0.18.17-py313h54dd161_2
Linking libmambapy-2.5.0-py313h4616538_0
Linking urllib3-2.6.3-pyhd8ed1ab_0
Linking zstandard-0.25.0-py313h54dd161_1
Linking conda-package-streaming-0.12.0-pyhd8ed1ab_0
Linking requests-2.32.5-pyhcf101f3_1
Linking conda-libmamba-solver-25.11.0-pyhd8ed1ab_1
done
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
#
# To activate this environment, use
```

lumi-container-wrapper (3)

```
kulust@uan03.lumi.csc - ~/Tykky-demo ⌘#2  
kulust@uan03.lumi.csc - ~/Tykky-demo (ssh) #1 +  
[ INFO ] Creating sqfs image  
Parallel mksquashfs: Using 8 processors##7 | 67%  
Creating 4.0 filesystem on _deploy/img.sqfs, block size 131072.  
[=====] 46353/46353 100%  
setuptools-75.3.0 | 761 KB | ##### | 100%  
Exportable Squashfs 4.0 filesystem, gzip compressed, data block size 131072  
  compressed data, compressed metadata, compressed fragments,  
  compressed xattrs, compressed ids  
  duplicates are removed  
Filesystem size 600799.81 Kbytes (586.72 Mbytes)  
  34.81% of uncompressed filesystem size (1725960.93 Kbytes)  
Inode table size 516727 bytes (504.62 Kbytes)  
  23.39% of uncompressed inode table size (2209519 bytes)  
Directory table size 744953 bytes (727.49 Kbytes)  
  41.39% of uncompressed directory table size (1799979 bytes)  
Number of duplicate files found 5663  
Number of inodes 47981  
Number of files 36284  
Number of fragments 2402  
Number of symbolic links 4129  
Number of device nodes 0  
Number of fifo nodes 0  
Number of socket nodes 0
```

lumi-container-wrapper (4)

```
kulust@uan03.lumi.csc - ~/Tykky-demo  ̣#2
kulust@uan03.lumi.csc - ~/Tykky-demo (ssh) #1  +

    41.39% of uncompressed directory table size (1799979 bytes)
Number of duplicate files found 5663
Number of inodes 47981
Number of files 36284
Number of fragments 2402
Number of symbolic links 4129
Number of device nodes 0
Number of fifo nodes 0
Number of socket nodes 0
Number of directories 7568
Number of hard-links 26930
Number of ids (unique uids + gids) 1
Number of uids 1
    kulust (327000143)
Number of gids 1
    pepr_kulust (327000143)
[ INFO ] Creating wrappers
[ INFO ] Installing to ./conda-cont-1
[ INFO ] Done, duration: 107s
[ INFO ] Program has been installed to ./conda-cont-1
        To use add the bin folder to your path e.g:
        export PATH="/users/kulust/Tykky-demo/conda-cont-1/bin:$PATH"
[lumi][kulust@uan03-1043 Tykky-demo]$
```

lumi-container-wrapper (5)

```
kulust@uan03.lumi.csc - ~/Tykky-demo
kulust@uan03.lumi.csc - ~/Tykky-demo (ssh) #1

[lumi][kulust@uan03-1043 Tykky-demo]$ ls conda-cont-1/
_bin bin common.sh container.sif img.sqfs share
[lumi][kulust@uan03-1044 Tykky-demo]$ ls conda-cont-1/bin
2to3                jsonschema          lzegrep              python3              wsdump
2to3-3.8            jupyter             lzfgrep              python3.8            x86_64-conda-linux-gnu-ld
captaininfo         jupyter-dejavu      lzgrep               python3.8-config    xz
clear               jupyter-events      lzless               python3-config       xzcat
c_rehash            jupyter-execute     lzma                  reset                xzcmp
curve_keygen        jupyter-kernel      lzmadec              send2trash           xzdec
_debug_exec         jupyter-kernelspec lzmainfo              sqlite3              xzdiff
debugpy             jupyter-lab         lzmore                sqlite3_analyzer     xzegrep
_debug_shell        jupyter-labextension ncurses6-config      tabs                  xzfgrep
f2py                jupyter-labhub      ncursesw6-config    tclsh                 xzgrep
f2py3               jupyter-migrate     normalizer            tclsh8.6             xzless
f2py3.8             jupyter-nbconvert   openssl              tic                    xzmore
httpx               jupyter-notebook    pip                   toe                    zstd
idle3                jupyter-run         pip3                  tput                  zstdcat
idle3.8             jupyter-server      pybabel              tset                  zstdgrep
infocmp             jupyter-troubleshoot pydoc                 unlzma                zstdless
infotocap           jupyter-trust       pydoc3                unxz                   zstdmt
ipython             list-packages       pydoc3.8              unzstd                 wheel
ipython3            lzcat               pygmentize            wish
jlp                 lzcmp               pyjson5
```

lumi-container-wrapper (6)

```
kulust@uan03.lumi.csc - ~/Tykky-demo/conda-cont-1/bin ⌘ #2  
kulust@uan03.lumi.csc - ~/Tykky-demo/conda-cont-1/bin (ssh) #1 +  
  
curve_keygen  jupyter-kernel      lzmadec      send2trash    xzdec  
_debug_exec   jupyter-kernelspec   lzmainfo     sqlite3        xzdiff  
debugpy       jupyter-lab          lzmore       sqlite3_analyzer xzegrep  
_debug_shell  jupyter-labextension ncurses6-config tabs           xzfgrep  
f2py          jupyter-labhub       ncursesw6-config tclsh         xzgrep  
f2py3         jupyter-migrate      normalizer   tclsh8.6      xzless  
f2py3.8      jupyter-nbconvert    openssl      tic           xzmore  
httpx         jupyter-notebook     pip          toe           zstd  
idle3         jupyter-run          pip3         tput         zstdcat  
idle3.8      jupyter-server       pybabel     tset         zstdgrep  
infocmp      jupyter-troubleshoot pydoc        unlzma        zstdless  
infotocap    jupyter-trust        pydoc3      unxz         zstdmt  
ipython      list-packages        pydoc3.8    unzstd         
ipython3     lzcat                pygmentize  wheel          
jlp          lzcmp                pyjson5     wish           
jsonpointer  lzdif                python      wish8.6        
[lumi][kulust@uan03-1045 Tykky-demo]$ cd conda-cont-1/bin  
[lumi][kulust@uan03-1046 bin]$ ./python3  
Python 3.8.8 | packaged by conda-forge | (default, Feb 20 2021, 16:22:27)  
[GCC 9.3.0] on linux  
Type "help", "copyright", "credits" or "license" for more information.  
>>> import numpy  
>>> 
```

Environment enhancements (4): Non-AI containers

- **lumi-vnc** (LUMI and CrayEnv software stacks)
 - Provides basic VNC virtual desktop for interacting with graphical interfaces via a web browser or VNC client
 - Open OnDemand a better alternative for many
- **ccpe**: Containerised Cray Programming Environment
 - For advanced users only
 - User-installable as often customisations are needed
 - Experiment with newer versions of the Cray PE
 - Functionality may be limited due to ROCm driver compatibility and compatibility of MPI with the network software stack

Environment enhancements (5): AI containers

- AI containers built by the LUMI AI Factory
 - Currently PyTorch-based
 - Hierarchy of containers:
 - `lumi-multitorch-rocm-*`: Base container based on Ubuntu with ROCm
 - `lumi-multitorch-libfabric-*`: Adding libfabric on top of the previous one
 - `lumi-multitorch-mpich-*`: Adding the MPICH MPI library with GPU support on top of the previous one
 - `lumi-multitorch-torch-*`: Adding PyTorch on top of the previous one
 - `lumi-multitorch-full-*`: Adding a selection of additional AI and ML libraries on top of the previous one
 - `/appl/local/laiifs/containers`
- Older containers still available in `/appl/local/containers/sif-images`
 - And EasyBuild recipes for a module providing bindings and some wrapper scripts

Extending containers 1: Virtual environment

- This strategy does not require rebuilding the container.
- Python has a mechanism to extend a read-only installation: virtual environments.
- Idea:
 - Create a bind mount for the virtual environment
 - Want a fixed path to be able to share with other projects
 - Create the virtual environment and install packages from within the container
 - Problem: The many-small-files problem returns...
 - Make a SquashFS file from that virtual environment
 - For further use, bind mount that SquashFS file instead of the directory
 - So the many-small-files problem is solved again

Step 1: Preparation

```
kulust@uan01.lumi.csc --
kulust@uan01.lumi.csc -- (ssh) #1

** `--*****-----***-----*****-----`

Did you know?
*****
With software built with EasyBuild (as is most software in the LUMI stack),
it is very easy to find the location of the software binaries. After loading
the module, an environment variable based on the name of the module will be
defined. E.g., after loading one of the Boost modules, the location of
Boost will be given by the EBR00TBOOST environment variable. Try
  env | egrep ^EBR00T
in a shell to see all EBR00T variables that are defined after loading some
modules.

[lumi][kulust@uan01-1001 ~]$ mkdir -p /tmp/$USER/user-software
[lumi][kulust@uan01-1002 ~]$ export SIF='/appl/local/laiifs/containers/lumi-multitorch-u24r64f21m43t29-2026
0225_144743/lumi-multitorch-full-u24r64f21m43t29-20260225_144743.sif'
[lumi][kulust@uan01-1003 ~]$ module load Local-LAIF lumi-aif-singularity-bindings
This software collection is provided and supported by the LUMI AI Factory.
Run `module help Local-LAIF` for more information about support.
Not all software may be compatible with software in other stacks,
so mix at your own risk.
[lumi][kulust@uan01-1004 ~]$ touch /tmp/$USER/user-software/demofile
[lumi][kulust@uan01-1005 ~]$
```

Our work directory

Save some typing later

Take care of bindings

Just for demo purposes

Step 2: Enter the container

kulust@uan01.lumi.csc --

⌘#2

kulust@uan01.lumi.csc -- (ssh) #1

it is very easy to find the location of the software binaries. After loading the module, an environment variable based on the name of the module will be defined. E.g., after loading one of the Boost modules, the location of Boost will be given by the EBROOTBOOST environment variable. Try

```
env | egrep ^EBROOT
```

in a shell to see all EBROOT variables that are defined after loading some modules.

```
[lumi][kulust@uan01-1001 ~]$ mkdir -p /tmp/$USER/user-software
```

```
[lumi][kulust@uan01-1002 ~]$ export SIF='/appl/local/laifs/containers/lumi-multitorch-u24r64f21m43t29-20260225_144743/lumi-multitorch-full-u24r64f21m43t29-20260225_144743.sif'
```

```
[lumi][kulust@uan01-1003 ~]$ module load Local-LAIF lumi-aif-singularity-bindings
```

This software collection is provided and supported by the LUMI AI Factory.

Run `'module help Local-LAIF'` for more information about support.

Not all software may be compatible with software in other stacks, so mix at your own risk.

```
[lumi][kulust@uan01-1004 ~]$ touch /tmp/$USER/user-software/demofile
```

```
[lumi][kulust@uan01-1005 ~]$ singularity run -B /tmp/$USER/user-software:/user-software $SIF bash -l
```

```
Singularity> ls -l /user-software
```

```
total 0
```

```
-rw-rw---- 1 kulust pepr_kulust 0 Mar 19 16:26 demofile
```

```
Singularity> rm -f /user-software/demofile
```

```
Singularity>
```

Start shell in container

Take care of bindings

Clean-up

Step 3: Create the venv and install (1)

```
kulust@uan01.lumi.csc --
kulust@uan01.lumi.csc -- (ssh) #1

Boost will be given by the EBR00TBOOST environment variable. Try
  env | egrep ^EBR00T
in a shell to see all EBR00T variables that are defined after loading some
modules.

[lumi][kulust@uan01-1001 ~]$ mkdir -p /tmp/$USER/user-software
[lumi][kulust@uan01-1002 ~]$ export SIF='/appl/local/laifs/containers/lumi-m
0225_144743/lumi-multitorch-full-u24r64f21m43t29-20260225_144743.sif'
[lumi][kulust@uan01-1003 ~]$ module load Local-LAIF lumi-aif-singularity-bin
This software collection is provided and supported by the LUMI AI Factory.
Run `module help Local-LAIF` for more information about support.
Not all software may be compatible with software in other stacks,
so mix at your own risk.
[lumi][kulust@uan01-1004 ~]$ touch /tmp/$USER/user-software/demofile
[lumi][kulust@uan01-1005 ~]$ singularity run -B /tmp/$USER/user-software:/user-software $SIF bash -l
Singularity> ls -l /user-software
total 0
-rw-rw---- 1 kulust pepr_kulust 0 Mar 19 16:26 demofile
Singularity> rm -f /user-software/demofile
Singularity> cd /user-software
Singularity> python -m venv mace-demo --system-site-packages
Singularity> source mace-demo/bin/activate
(mace-demo) Singularity> pip install mace-torch
```

Go to directory

Create environment

Activate environment

Install mace

Step 3: Create the venv and install (2)

```
kulust@uan01.lumi.csc --
kulust@uan01.lumi.csc -- (ssh) #1
----- 325.9/325.9 kB 1.2 MB/s eta 0:00:00
Downloading matscipy-1.2.0-cp312-cp312-manylinux_2_24_x86_64.manylinux_2_28_x86_64.whl (453 kB)
----- 453.1/453.1 kB 1.6 MB/s eta 0:00:00
Downloading opt_einsum-3.4.0-py3-none-any.whl (71 kB)
----- 71.9/71.9 kB 93.8 kB/s eta 0:00:00
Downloading orjson-3.11.7-cp312-cp312-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (133 kB)
----- 133.4/133.4 kB 582.0 kB/s eta 0:00:00
Downloading prettytable-3.17.0-py3-none-any.whl (34 kB)
Downloading torch_ema-0.3-py3-none-any.whl (5.5 kB)
Downloading opt_einsum_fx-0.1.4-py3-none-any.whl (13 kB)
Building wheels for collected packages: python-hostlist
  Building wheel for python-hostlist (setup.py) ... done
  Created wheel for python-hostlist: filename=python_hostlist-2.3.0-py3-none-any.whl size=39521 sha256=968
  cleaad6e80ed642bb3ec5d10085eca4b5fa4b330bc2157e2ae460bd3eaa11
  Stored in directory: /users/kulust/.cache/pip/wheels/02/e4/34/75fc0cd5b7889d8cc4ce6fb2f74c9fd17b3c6138cb
  03832481
Successfully built python-hostlist
Installing collected packages: python-hostlist, lmbd, prettytable, orjson, opt_einsum, h5py, configargpars
e, torch-ema, opt-einsum-fx, ase, matscipy, e3nn, mace-torch
Successfully installed ase-3.28.0 configargparse-1.7.5 e3nn-0.4.4 h5py-3.16.0 lmbd-2.1.1 mace-torch-0.3.15
matscipy-1.2.0 opt-einsum-fx-0.1.4 opt_einsum-3.4.0 orjson-3.11.7 prettytable-3.17.0 python-hostlist-2.3.
0 torch-ema-0.3
http://MB/s mo) Singularity>
```

Step 3: Create the venv and install (3)

L U M I

```
kulust@uan01.lumi.csc --  
kulust@uan01.lumi.csc -- (ssh) #1  
(mace-demo) Singularity> ls mace-demo/bin  
activate  
activate.csh  
activate.fish  
Activate.ps1  
ase  
dbuck  
hostgrep  
hostlist  
mace_active_learning_md  
(mace-demo) Singularity> ls mace-demo/lib/python3.12/site-packages/  
ase  
ase-3.28.0.dist-info  
configargparse-1.7.5.dist-info  
configargparse.py  
e3nn  
e3nn-0.4.4.dist-info  
h5py  
h5py-3.16.0.dist-info  
h5py.libs  
hostlist.py  
lmbd  
(mace-demo) Singularity> find . | wc -l  
4903  
(mace-demo) Singularity>
```

mace_convert_device	mace_select_head	matscipy-stericify
mace_create_lammps_model	matscipy-average-eam-potential	pip
mace_cueq_to_e3nn	matscipy-continuous2discrete	pip3
mace_e3nn_cueq	matscipy-poisson-nernst-planck	pip3.12
mace_eval_configs	matscipy-quasistatic-crack	pshbak
mace_finetuning_select	matscipy-quench	python
mace_plot_train	matscipy-rms	python3
mace_prepare_data	matscipy-sinclair-continuation	python3.12
mace_run_train	matscipy-sinclair-crack	

lmbd-2.1.1.dist-info	pip
mace	pip-24.0.dist-info
mace_torch-0.3.15.dist-info	prettytable
matscipy	prettytable-3.17.0.dist-info
matscipy-1.2.0.dist-info	__pycache__
opt_einsum	python_hostlist-2.3.0.dist-info
opt_einsum-3.4.0.dist-info	scripts
opt_einsum_fx	tests
opt_einsum_fx-0.1.4.dist-info	torch_ema
orjson	torch_ema-0.3.dist-info
orjson-3.11.7.dist-info	

Step 3: Create the venv and install (4)

L U M I

```
kulust@uan01.lumi.csc ~ - -
kulust@uan01.lumi.csc ~ - - (ssh) #1

[lumi][kulust@uan01-1006 ~]$ ls /tmp/$USER/user-software/mace-demo/bin
activate                mace_convert_device      mace_select_head         matscipy-stericify
activate.csh            mace_create_lammps_model matscipy-average-eam-potential pip
activate.fish          mace_cueq_to_e3nn        matscipy-continuous2discrete pip3
Activate.ps1           mace_e3nn_cueq           matscipy-poisson-nernst-planck pip3.12
ase                     mace_eval_configs        matscipy-quasistatic-crack pshbak
dbuck                   mace_finetuning_select  matscipy-quench          python
hostgrep                mace_plot_train          matscipy-rms             python3
hostlist                mace_prepare_data        matscipy-sinclair-continuation python3.12
mace_active_learning_md mace_run_train           matscipy-sinclair-crack

[lumi][kulust@uan01-1007 ~]$ ls /tmp/$USER/user-software/mace-demo/lib/python3.12/site-packages/
ase                      lmbd-2.1.1.dist-info      pip
ase-3.28.0.dist-info    mace                       pip-24.0.dist-info
configargparse-1.7.5.dist-info mace_torch-0.3.15.dist-info prettytable
configargparse.py      matscipy                   prettytable-3.17.0.dist-info
e3nn                    matscipy-1.2.0.dist-info  __pycache__
e3nn-0.4.4.dist-info   opt_einsum                 python_hostlist-2.3.0.dist-info
h5py                    opt_einsum-3.4.0.dist-info scripts
h5py-3.16.0.dist-info  opt_einsum_fx              tests
h5py.libs               opt_einsum_fx-0.1.4.dist-info torch_ema
hostlist.py             orjson                     torch_ema-0.3.dist-info
lmbd                    orjson-3.11.7.dist-info

[lumi][kulust@uan01-1008 ~]$ find /tmp/$USER/user-software | wc -l
4903

[lumi][kulust@uan01-1009 ~]$
```

Step 4: Create the SquashFS file (1)

```
kulust@uan01.lumi.csc - /tmp/kulust
kulust@uan01.lumi.csc - /tmp/kulust (ssh) #1
[lumi][kulust@uan01-1009 ~]$ cd /tmp/$USER
[lumi][kulust@uan01-1010 kulust]$ mksquashfs user-software /project/project_465000095/$USER/user-software-
mace.sqsh -processors 1 -all-root -action "chmod(a+rX) @true"
Parallel mksquashfs: Using 1 processor
Creating 4.0 filesystem on /project/project_465000095/kulust/user-software-mace.sqsh, block size 131072.
[=====] 4402/4402 100%

Exportable Squashfs 4.0 filesystem, gzip compressed, data block size 131072
  compressed data, compressed metadata, compressed fragments,
  compressed xattrs, compressed ids
  duplicates are removed
Filesystem size 18990.67 Kbytes (18.55 Mbytes)
  29.66% of uncompressed filesystem size (64035.93 Kbytes)
Inode table size 45729 bytes (44.66 Kbytes)
  28.97% of uncompressed inode table size (157834 bytes)
Directory table size 49260 bytes (48.11 Kbytes)
  35.78% of uncompressed directory table size (137656 bytes)
Number of duplicate files found 146
Number of inodes 4903
Number of files 4334
Number of fragments 347
Number of symbolic links 4
Number of device nodes 0
```

Change work directory

Make SquashFS file,

set uid and gid to 0 (root),

correct all permissions

Step 4: Create the SquashFS file (2)

```
kulust@uan01.lumi.csc - /tmp/kulust
kulust@uan01.lumi.csc - /tmp/kulust (ssh) #1
duplicates are removed
Filesystem size 18990.67 Kbytes (18.55 Mbytes)
  29.66% of uncompressed filesystem size (64035.93 Kbytes)
Inode table size 45729 bytes (44.66 Kbytes)
  28.97% of uncompressed inode table size (157834 bytes)
Directory table size 49260 bytes (48.11 Kbytes)
  35.78% of uncompressed directory table size (137656 bytes)
Number of duplicate files found 146
Number of inodes 4903
Number of files 4334
Number of fragments 347
Number of symbolic links 4
Number of device nodes 0
Number of fifo nodes 0
Number of socket nodes 0
Number of directories 565
Number of hard-links 0
Number of ids (unique uids + gids) 1
Number of uids 1
  root (0)
Number of gids 1
  root (0)
[lumi][kulust@uan01-1011 kulust]$
```

Step 4: Create the SquashFS file (3)

```
kulust@uan01.lumi.csc - /tmp/kulust
kulust@uan01.lumi.csc - /tmp/kulust (ssh) #1
[lumi][kulust@uan01-1016 kulust]$ rm -rf /tmp/$USER/user-software
[lumi][kulust@uan01-1017 kulust]$ unsquashfs -d ./user-software -processors 1 /project/project_465000095/kulust/user-software-mace.sqsh
Parallel unsquashfs: Using 1 processor
4338 inodes (4402 blocks) to write

[=====|] 8740/8740 100%

created 4334 files
created 565 directories
created 4 symlinks
created 0 devices
created 0 fifos
created 0 sockets
created 0 hardlinks
[lumi][kulust@uan01-1018 kulust]$ ls user-software/mace-demo/
bin include lib lib64 pyenv.cfg share
[lumi][kulust@uan01-1019 kulust]$
```

Clean up

Recover files from SquashFS file

See they are back

Step 5: Use the software

```
kulust@uan01.lumi.csc - /tmp/kulust
kulust@uan01.lumi.csc - /tmp/kulust (ssh) #1

[lumi][kulust@uan01-1021 kulust]$ ls user-software/mace-demo/
bin include lib lib64 pyvenv.cfg share
[lumi][kulust@uan01-1022 kulust]$ singularity run -B /project/project_465000095/kulust/user-software-mace.
sqsh /user-software:/image-src=/ $SIF bash -l
Singularity> ls /user-software/mace-demo
bin include lib lib64 pyvenv.cfg share
Singularity> source /user-software/mace-demo/bin/activate
(mace-demo) Singularity> python -c 'import mace; print( mace.__version__ )'
0.3.15
(mace-demo) Singularity>
```

Outside the container

Create environment

Inside the container

Activate environment

Test: import mace

SquashFS file

Mounted where in the container

Treat as a filesystem, not a file, and subdirectory that should be used

Extending containers 2: Unprivileged PRoot build L U M I

- With this strategy, we build a new container from an existing one.
- What can be done on LUMI is a bit limited as only the PRoot process can be used.
- But we can:
 - Build from an existing container on the filesystem or from dockerhub, but not all bootstrap options work
 - Copy files and directories from the host so that we don't need to bind them
 - Add preset environment variables
 - Install additional OS packages, compile software, install additional Python packages, ...
 - Change the runscript
- Steered by a definition file
 - See the examples on the next slides
 - Header, `%files`, `%environment`, `%post`, `%runscript`, ...

Example 1: Python package mace-torch

L U M I

mace.def

Header

```
Bootstrap: localimage  
From: {{ sif }}
```

Template, replace with concrete filename at build time

```
%environment  
    export NCCL_SOCKET_IFNAME='hsn0,hsn1,hsn2,hsn3'
```

```
%post  
    pip install mace-torch
```

Initialise this variable when starting the container

Commands to execute. The virtual environment /opt/venv is already initialised when running these commands.

Example 1: Python package mace-torch (2)

- Build new container without using the bindings module:

```
srun -pstandard -N1 -t2:00:00 --pty \  
singularity build \  
  --build-arg sif=$SIF \  
  lumi-multitorch-mace-u24r64f21m43t29-20260423.sif mace.def
```

- `--build-arg sif=$SIF` : Set the value for the template `{{ sif }}`
- Memory-intensive and long build time
- To do a build with your regular filesystems mounted: See the bonus examples in the notes.
- Demo:

Big container, so using a compute node

Define the template in the definition file

Name for the new container

Definition file

Example 1: Python package mace-torch (3)

L U M I

```
kulust@uan01.lumi.csc - /scratch/project_465000095/kulust
kulust@uan01.lumi.csc - /scratch/project_465000095/kulust (ssh) #1

[lumi][kulust@uan01-1033 kulust]$ cat mace.def
Bootstrap: localimage
From: {{ sif }}

%environment
    export NCCL_SOCKET_IFNAME='hsn0,hsn1,hsn2,hsn3'

%post
    pip install mace-torch
[lumi][kulust@uan01-1034 kulust]$ export SIF='/appl/local/laifs/containers/lumi-multitorch-u24r64f21m43t29-20260225_144743/lumi-multitorch-full-u24r64f21m43t29-20260225_144743.sif'
[lumi][kulust@uan01-1035 kulust]$ module unload lumi-aif-singularity-bindings
[lumi][kulust@uan01-1036 kulust]$ module load CrayEnv PRoot
[lumi][kulust@uan01-1037 kulust]$ srun -pstandard -N1 -t2:00:00 -A project_465000095 --pty \
> singularity build \
>     --build-arg sif=$SIF \
>     lumi-multitorch-mace-u24r64f21m43t29-20260423.sif mace.def
srun: job 17586474 queued and waiting for resources
srun: job 17586474 has been allocated resources
INFO: Using proot to build unprivileged. Not all builds are supported. If build fails, use --remote or --fakeroot.
WARNING: 'nodev' mount option set on /tmp, it could be a source of failure during build process
INFO: Starting build...
```

Example 1: Python package mace-torch (4)

L U M I

```
kulust@uan01.lumi.csc - /scratch/project_465000095/kulust
kulust@uan01.lumi.csc - /scratch/project_465000095/kulust (ssh) #1

----- 453.1/453.1 kB 82.3 MB/s eta 0:00:00
Downloading opt_einsum-3.4.0-py3-none-any.whl (71 kB)
----- 71.9/71.9 kB 35.4 MB/s eta 0:00:00
Downloading orjson-3.11.7-cp312-cp312-manylinux_2_17_x86_64_musl-manylinux2014_x86_64.whl (133 kB)
----- 133.4/133.4 kB 64.3 MB/s eta 0:00:00
Downloading prettytable-3.17.0-py3-none-any.whl (34 kB)
Downloading torch_ema-0.3-py3-none-any.whl (5.5 kB)
Downloading opt_einsum_fx-0.1.4-py3-none-any.whl (13 kB)
Building wheels for collected packages: python-hostlist
  Building wheel for python-hostlist (setup.py) ... done
  Created wheel for python-hostlist: filename=python_hostlist-2.3.0-py3-none-any.whl size=39521 sha256=a0d8a072f3798750115a0202147144fa50de512288d88cd5941aedd792dc7726
  Stored in directory: /users/kulust/.cache/pip/wheels/02/e4/34/75fc0cd5b7889d8cc4ce6fb2f74c9fd17b3c6138cb03832481
Successfully built python-hostlist
Installing collected packages: python-hostlist, lmbd, prettytable, orjson, opt_einsum, h5py, configargparse, torch-ema, opt-einsum-fx, ase, matscipty, e3nn, mace-torch
Successfully installed ase-3.28.0 configargparse-1.7.5 e3nn-0.4.4 h5py-3.16.0 lmbd-2.1.1 mace-torch-0.3.15 matscipty-1.2.0 opt-einsum-fx-0.1.4 opt_einsum-3.4.0 orjson-3.11.7 prettytable-3.17.0 python-hostlist-2.3.0 torch-ema-0.3
INFO: Adding environment to container
INFO: Creating SIF file...
```

A lot of output not shown...

And then it seems stuck for an hour...

Example 1: Python package mace-torch (5)

L U M I

```
kulust@uan01.lumi.csc - /scratch/project_465000095/kulust
kulust@uan01.lumi.csc - /scratch/project_465000095/kulust (ssh) #1

03832481
Successfully built python-hostlist
Installing collected packages: python-hostlist, lmbd, prettytable, orjson, opt_einsum, h5py, configargpars
e, torch-ema, opt-einsum-fx, ase, matscipty, e3nn, mace-torch
Successfully installed ase-3.28.0 configargparse-1.7.5 e3nn-0.4.4 h5py-3.16.0 lmbd-2.1.1 mace-torch-0.3.15
matscipty-1.2.0 opt-einsum-fx-0.1.4 opt_einsum-3.4.0 orjson-3.11.7 prettytable-3.17.0 python-hostlist-2.3.
0 torch-ema-0.3
INFO: Adding environment to container
INFO: Creating SIF file...
INFO: Build complete: lumi-multitorch-mace-u24r64f21m43t29-20260423.sif
[lumi][kulust@uan01-1038 kulust]$ singularity run lumi-multitorch-mace-u24r64f21m43t29-20260423.sif bash -
l
Singularity> ls -l /opt/venv/bin/mace_run_train
-rwxrwxr-x 1 root root 225 Mar 20 17:42 /opt/venv/bin/mace_run_train
Singularity> ls /opt/venv/lib/python3.12/site-packages/mace*
/opt/venv/lib/python3.12/site-packages/mace:
calculators cli data __init__.py modules __pycache__ py.typed tools __version__.py

/opt/venv/lib/python3.12/site-packages/mace_torch-0.3.15.dist-info:
entry_points.txt INSTALLER licenses METADATA RECORD REQUESTED top_level.txt WHEEL
Singularity> python -c 'import mace; print(mace.__version__)'
0.3.15
Singularity>
```

Example 2: Install Ubuntu application (1)

L U M I

mace.def

```
Bootstrap: docker
From: docker.io/lumiaifactory/lumi-multitorch:mpich-u24r64f21m43t29-20260225_144743
```

```
%post
```

```
apt update
```

```
apt install -y vim
```

Refresh the package list

Install vim and dependencies
Note -y

Installing from docker

Example 2: Install Ubuntu application (2)

```
kulust@uan02.lumi.csc -- ~
kulust@uan02.lumi.csc -- (ssh) #1

[lumi][kulust@uan02-1004 ~]$ cat userexts.def
Bootstrap: docker
From: docker.io/lumiaifactory/lumi-multitorch:mpich-u24r64f21m43t29-20260225_144743

%post
  apt update
  apt install -y vim
[lumi][kulust@uan02-1005 ~]$ module load LUMI PRoot
[lumi][kulust@uan02-1006 ~]$ singularity build \
> lumi-multitorch-user-u24r64f21m43t29-20260423.sif userexts.def
INFO: Using proot to build unprivileged. Not all builds are supported. If build fails, use --remote or
--fakeroot.
INFO: Starting build...
INFO: Fetching OCI image...
51.9MiB / 51.9MiB [=====] 100 % 57.8 MiB/s 0s
51.3MiB / 51.3MiB [=====] 100 % 57.8 MiB/s 0s
5.4GiB / 5.4GiB [=====] 100 % 57.8 MiB/s 0s
28.3MiB / 28.3MiB [=====] 100 % 57.8 MiB/s 0s
3.8MiB / 3.8MiB [=====] 100 % 57.8 MiB/s 0s
55.7MiB / 55.7MiB [=====] 100 % 57.8 MiB/s 0s
INFO: Extracting OCI image...
INFO: Inserting Singularity configuration...
INFO: Running post scriptlet
```

Example 2: Install Ubuntu application (3)

```
kulust@uan02.lumi.csc -- ~
kulust@uan02.lumi.csc -- (ssh) #1
Unpacking vim (2:9.1.0016-1ubuntu7.10) ...
Setting up libsodium23:amd64 (1.0.18-1ubuntu0.24.04.1) ...
Setting up libgpm2:amd64 (1.20.7-11) ...
Setting up xxd (2:9.1.0016-1ubuntu7.10) ...
Setting up vim-common (2:9.1.0016-1ubuntu7.10) ...
Setting up vim-runtime (2:9.1.0016-1ubuntu7.10) ...
Setting up vim (2:9.1.0016-1ubuntu7.10) ...
Processing triggers for man-db (2.12.0-4build2) ...
setpriv: setresgid failed: Operation not permitted
Processing triggers for libc-bin (2.39-0ubuntu8.7) ...
INFO: Creating SIF file...
INFO: Build complete: lumi-multitorch-user-u24r64f21m43t29-20260423.sif
[lumi][kulust@uan02-1007 ~]$ singularity run lumi-multitorch-user-u24r64f21m43t29-20260423.sif \
> bash -c 'vim --version'
NVIM v0.9.5
Build type: Release
LuaJIT 2.1.1703358377

system vimrc file: "$VIM/sysinit.vim"
fall-back for $VIM: "/usr/share/nvim"

Run :checkhealth for more info
[lumi][kulust@uan02-1008 ~]$
```

A lot of output not shown...

Bonus examples in the notes

- What if we want bind mounts during the `%post` processing?
 - Matching file or directory needed already
 - Trick: Abuse `%files` to copy an empty file or directory
- Can we call slurm commands in the container?
 - Yes, but it can be highly non-trivial, certainly if the OS in the container doesn't match the OS on LUMI
 - It may be dangerous to use a version of the client tools that does not match the version on LUMI
- What if installing a package does not work (example: FFmpeg)?
 - You'll have to be very creative
 - Look for other sources of precompiled binaries
 - Or install those dependencies that you can install, and proceed with manual compiles...
- Installing Python from scratch
 - Example with installing a package from sources
 - Relevant if you want to compile an application yourself also

Container limitations on LUMI

- “Bring your own userland and run on a system-optimised kernel” and you’ll be fine is nonsense.
- Containers use the host’s operating system kernel which may be different from what the container expects. Containers also do not abstract hardware.
 - This is particularly true for ROCm as each version works only with certain driver versions
- Much of system-specific optimisation is done in userland:
 - Optimise the 99%, not the 1%
 - Optimisations for a specific CPU and GPU instruction set
 - A generic container may not offer sufficiently good support for the Slingshot 11 interconnect on LUMI and fall back to TCP sockets resulting in poor performance, or not work at all.
 - Solution: inject Cray MPICH, but only for containers with ABI compatibility with MPICH.
 - Distributed AI: Need to inject the proper RCCL plugin.
- Only limited support for building containers on LUMI due to security concerns.
 - Yet we can still do a lot.

Questions?

