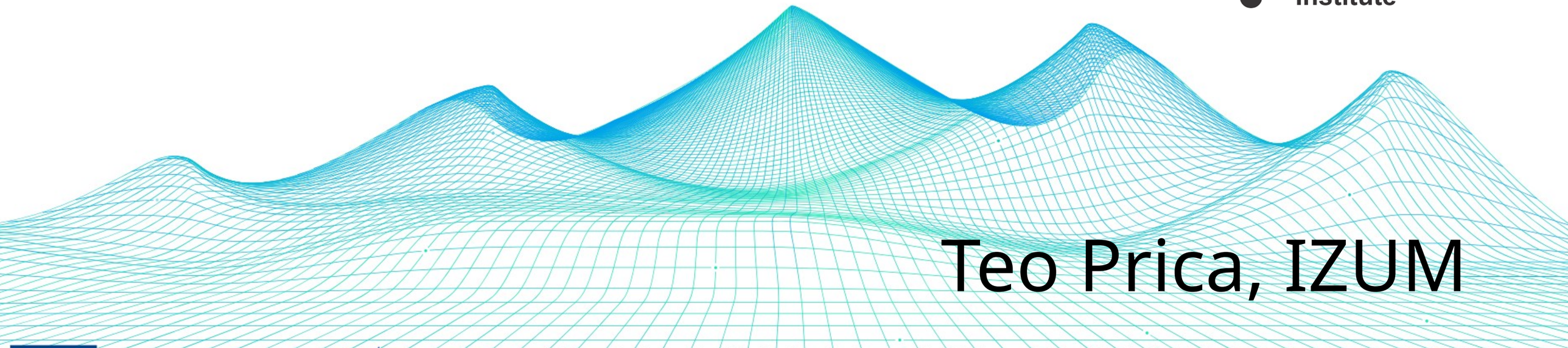
EPICURE

VEGA HPC

IZUM
Institute of Information Science, Maribor, Slovenia

Jožef Stefan Institute

Containerization in HPC Environments



Teo Prica, IZUM

Co-funded by the European Union

EuroHPC Joint Undertaking

This project has received funding from the European High Performance Computing Joint Undertaking under grant agreementNo.101139786. Views and opinions expressed are, however, those of the author(s) only and do not necessarily reflect those of the European Union orEuroHPC Joint Undertaking. Neither the European Union nor the granting authority can be held responsible for them.

Agenda

1. Context
2. Software
3. Containerization
4. Workloads
5. Security
6. Containers at HPC Vega

Introduction

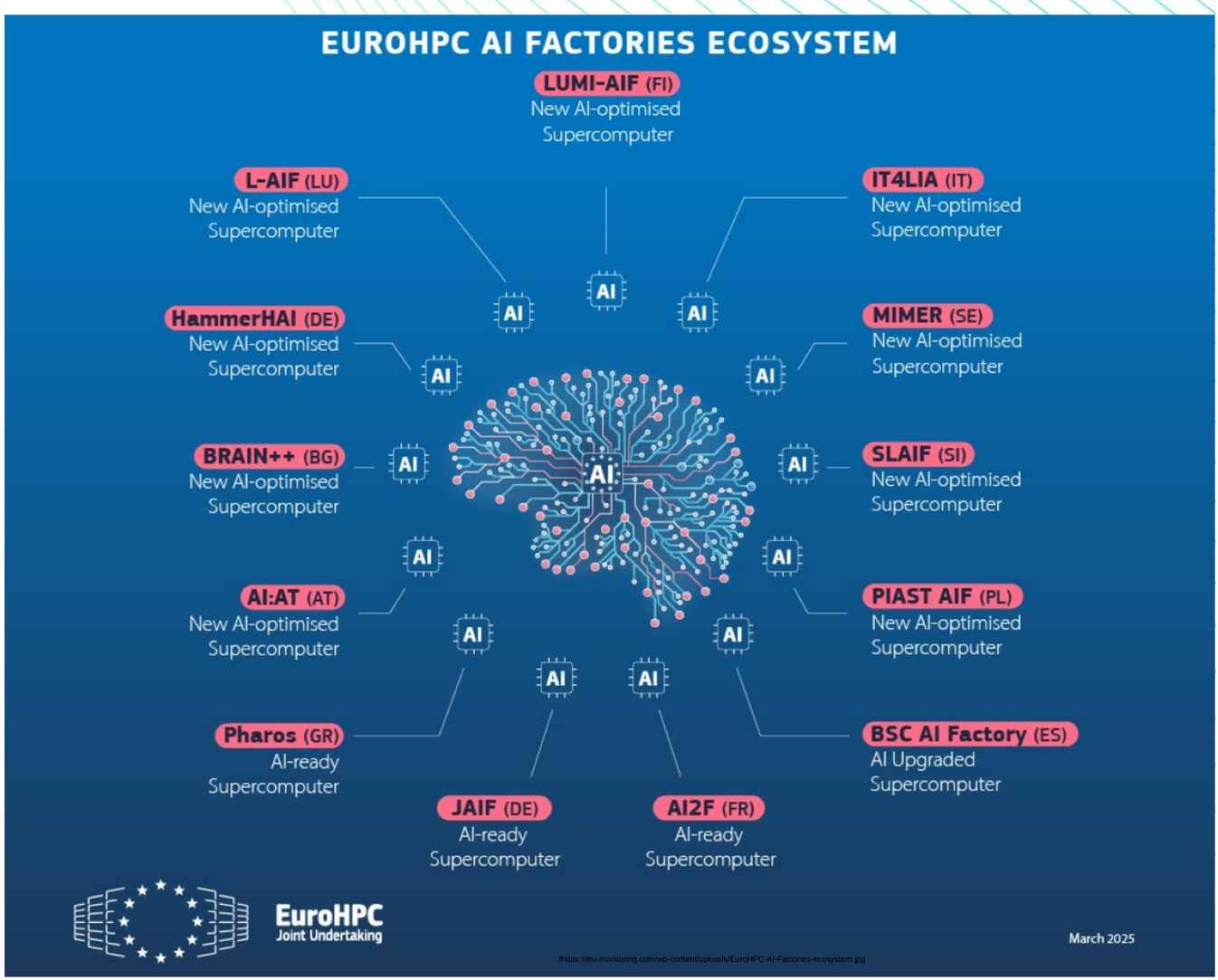
- The webinar on basic containerization in HPC environments.
- An overview of key tools and techniques.
- Use of Singularity/Apptainer containers:
  - Parallelisation wit MPI,
  - CUDA for GPU-accelerated workloads,
  - managing smaller and lightweight environments.

Artificial Intelligence

- AI in focus.
- EuroHPC AI Factories.
  - Future AI-Optimized HPC.
  - Specialized hardware requirements.
  - Industrial users (SMEs).
- Challenges..

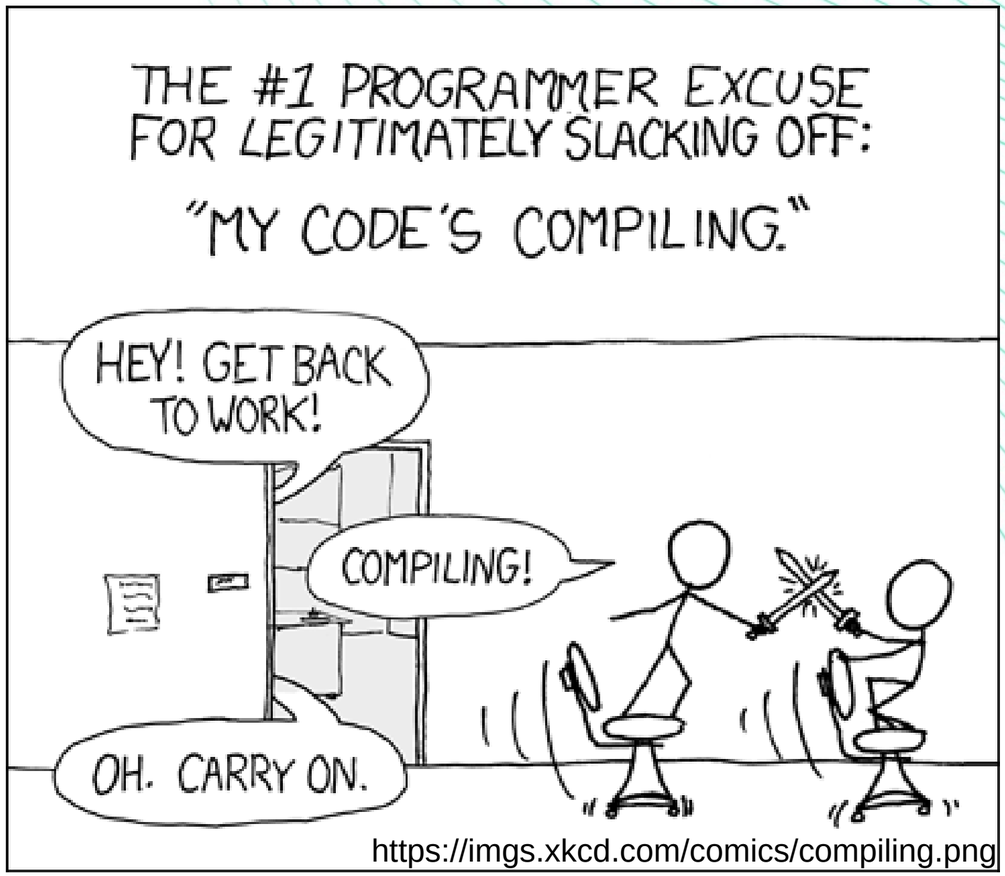
  Software integration in HPC,

  codes, workloads, optimization, skill gap, …



Software in HPC

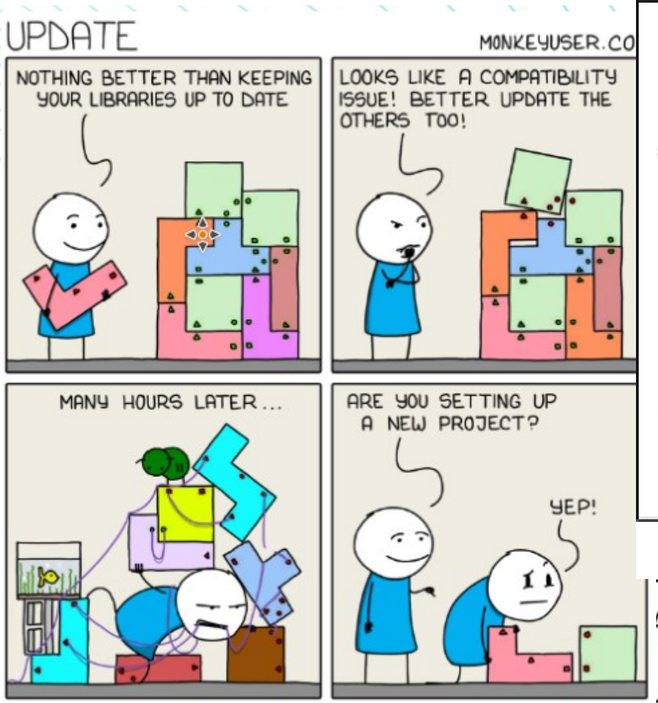
- Scientific software presents challenges,
- lacking of comments and examples,
- insufficient documentation,
- written by scientists forced to code,
- lack of support,
- non-standard installation,
- non-optimized,
- and beyond.

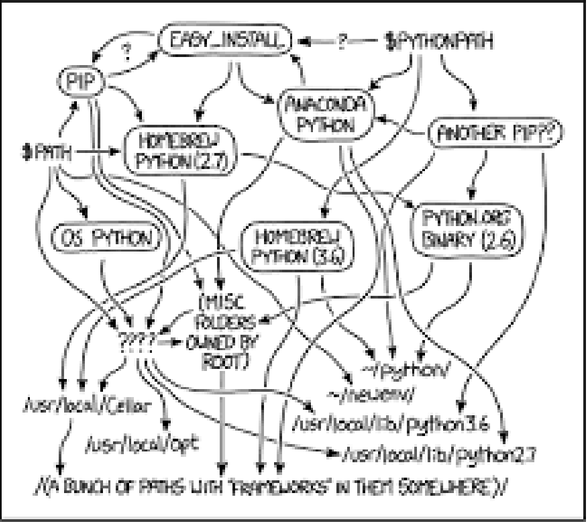


https://imgs.xkcd.com/comics/compiling.png

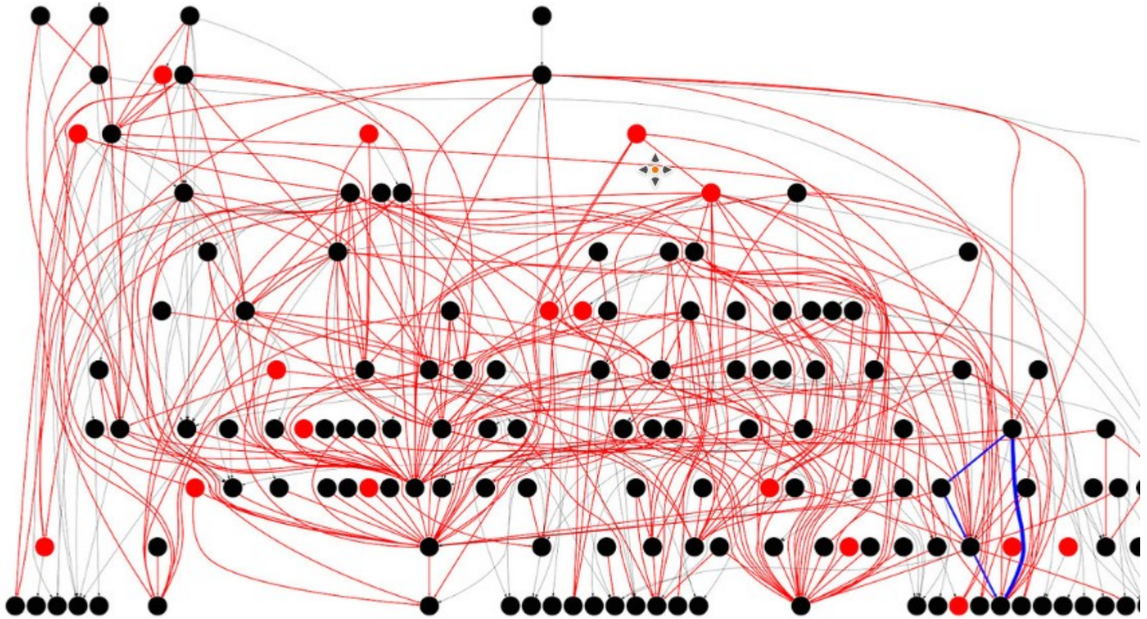
Dependency Hell

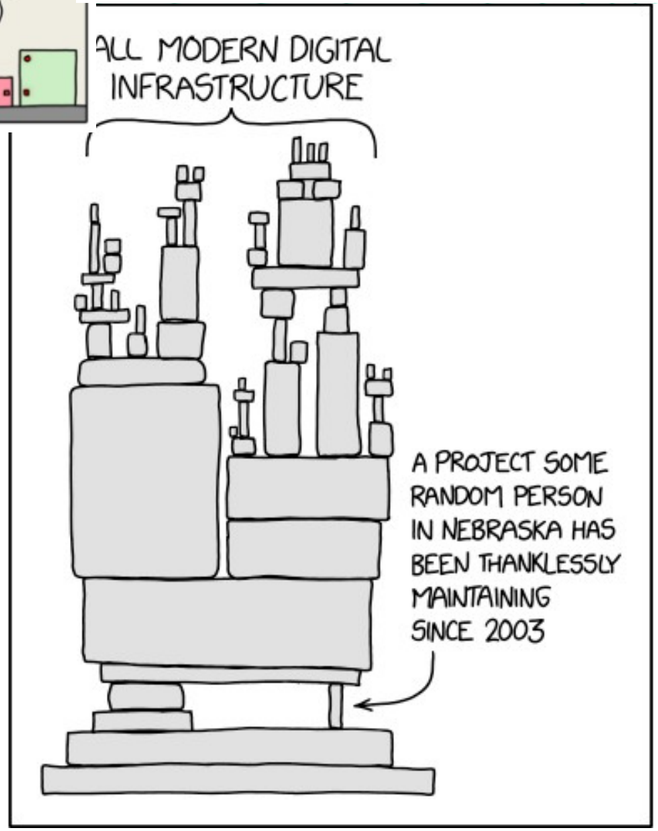
- Is a common problem due to complex, conflicting, or overlapping dependencies in the software stack.





MY PYTHON ENVIRONMENT HAS BECOME SO DEGRADED THAT MY LAPTOP HAS BEEN DECLARED A SUPERFUND SITE.

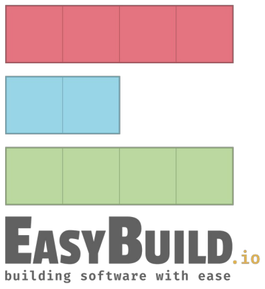




https://assets.browserlondon.com/wp-content/uploads/2020/08/XKCD-dependency.png

Software in HPC

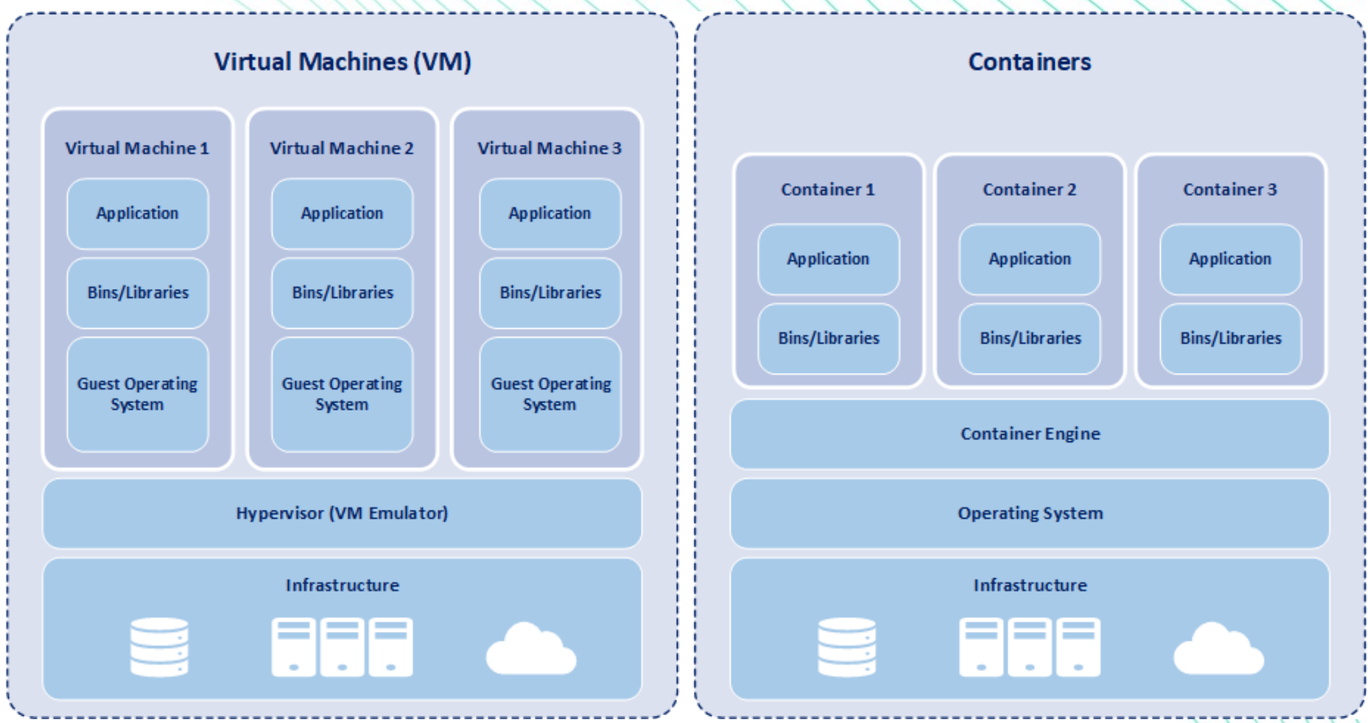
- Package managers,
  - EasyBuild, Spack, Conda, ..
- system packages,
- environment modules,
- custom builds,
- containers,
- and beyond.



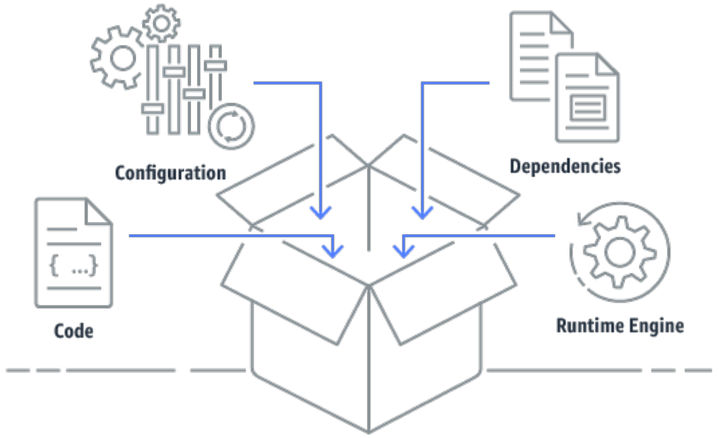
Container

- A container is a running process on the system controlled by the host kernel.
  - Isolated,
  - lightweight,
  - standalone,
  - portable,
  - reproducible,
  - secure,
  - and executable software package.

- Everything needed to run an software.



https://bi-insider.com/wp-content/uploads/2021/09/Containers-Virtual-Machines.png



https://miro.medium.com/v2/resize:fit:600/1*yTyYD5XYYAGMbhgLoCgeJw.png

Container: Keywords

Definition file (recipe)

Instructions for making the image.

Build

Compilation/converting the recipe into image.

Image

Compiled container, ready to be used.

Repository

Place to store and share the images.

Container

Running image.

Runtime

Software that manages containers.

Containarization in HPC

- Containers in HPC offer flexible and alternative software deployment.
- Avoid local installation on the system.
  - Multiple versions.
- Container shares the physical hardware. and OS with other containers.
  - Comparable to bare-metal.
  - Container != VM
- Embraced technology, slow adaptation.
- Convenient option for industrial users.

Isolation != Security

https://media.licdn.com/dms/image/v2/D5612AQECdlnTOjwf5g/article-cover_image-shrink_720_1280/article-cover_image-shrink_720_1280/0/1687160334655?e=2147483647&v=beta&t=wZebUrPXlo1sNgkrCHjuIAMwt0NZeWmfKQE0F08tc4I

Integrated into cloud-based, HPC, AI/ML workloads, and scientific computing frameworks.

Containarization in HPC

Portability, Deployment, and Scalability

Ensures that software runs consistently across different environments.

Reproducibility

Definition files means documentation and procedure could be repeated.

Dependency Management

Avoids conflicts by encapsulating software dependencies.

User Flexibility

Allows users to use their preferred software stacks, legacy code, ..

Performance Efficiency

Unlike VMs, containers introduce minimal overhead.

Security and Isolation

User privileges, namespaces, controlled environments, ..

Containers have come a long way!

- Early concepts of isolation from 1970’s.
- Convergence of HPC, AI & Cloud!
- Not suitable for all software!
- Containers are not just in HPC.
  - Cloud computing,
  - edge computing,
  - serverless computing
  - AI/ML Workloads
  - CI/CD, Devops, ..
- Charliecloud, Singularity/Apptainer, Enroot, Docker, Podman, Sarus, ...

Apptainer, formerly Singularity

- A container platform designed specifically for HPC, MPI and scientific workloads.
- It allows users to run containers without root privileges.
- Ideal for multi-user environments.
- Simplifies the workflow (end-to-end).
- Integrated in common workload manager Slurm.
- Secure. Portable. Encrypted.

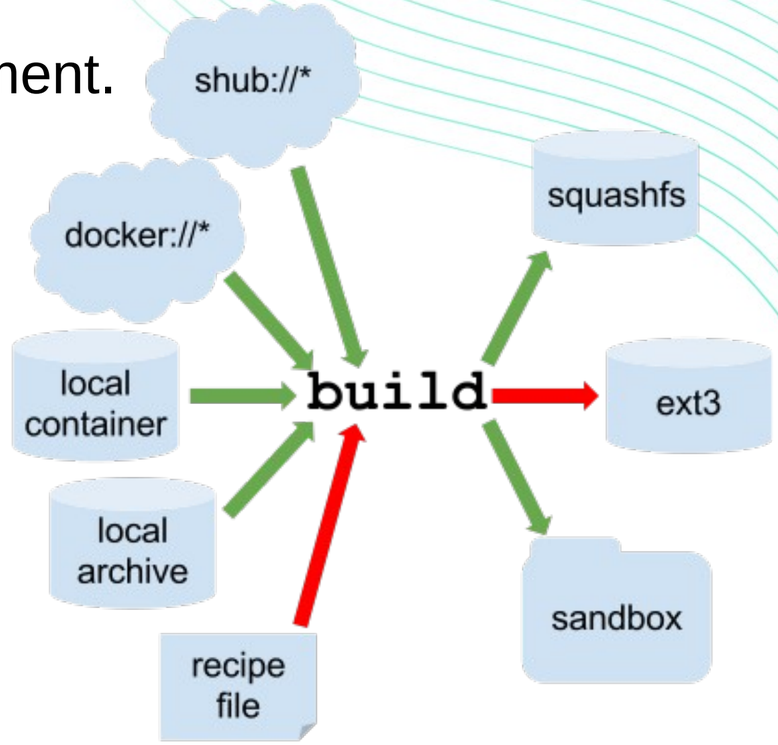
slurm
workload manager

https://sylabs.io/singularity-pro/

https://apptainer.org

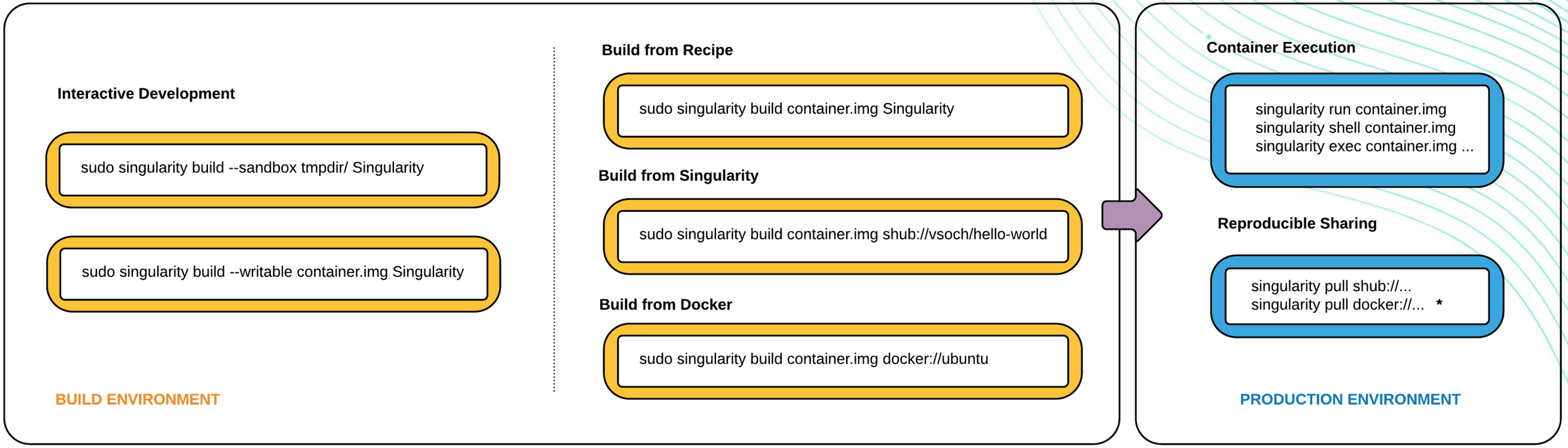
Common Workflow

- Preparation, build
  - Use a container definition, specify software environment.
- Registry
  - Push the container in a repository.
- Deploy
  - Run the container via job schedulers.
- Execute, scale
  - The container runs across multiple nodes, leveraging HPC resources.



https://docs.sylabs.io/guides/2.6/user-guide/_images/build_input_output.png

Common Workflow



https://singularityware.github.io/

* Docker construction from layers not guaranteed to replicate between pulls

Pull, Run, and Exec a Container

- Get image, run, and execute within existing container from repository.
- Pull

```
[teop@vglogin0008 ~]$ singularity pull ubuntu.sif docker://ubuntu:latest
INFO:    Converting OCI blobs to SIF format
INFO:    Starting build...
INFO:    Fetching OCI image...
28.4MiB / 28.4MiB [=============================================================================================================] 100 % 592.2 KiB/s 0s
INFO:    Extracting OCI image...
INFO:    Inserting Singularity configuration...
INFO:    Creating SIF file...
```

- Run

```
[teop@vglogin0008 ~]$ singularity run ubuntu.sif cat /etc/os-release
PRETTY_NAME="Ubuntu 24.04.1 LTS"
NAME="Ubuntu"
VERSION_ID="24.04"
VERSION="24.04.1 LTS (Noble Numbat)"
VERSION_CODENAME=noble
ID=ubuntu
ID_LIKE=debian
HOME_URL="https://www.ubuntu.com/"
SUPPORT_URL="https://help.ubuntu.com/"
BUG_REPORT_URL="https://bugs.launchpad.net/ubuntu/"
PRIVACY_POLICY_URL="https://www.ubuntu.com/legal/terms-and-policies/privacy-policy"
UBUNTU_CODENAME=noble
LOGO=ubuntu-logo
```

- Exec

```
[teop@vglogin0008 ~]$ singularity exec ubuntu.sif cat /etc/os-release | grep PRETTY_NAME
PRETTY_NAME="Ubuntu 24.04.1 LTS"
```

Building a Singularity Container

- Singularity Definition File.

```
Bootstrap: docker
From: ubuntu:24.04

%post
    apt update && apt install -y python3 pip
    pip install torch

%runscript
    echo "Hello EPICURE!"
```

- Execute “%runscript” through run command.

```
[teop@vglogin0008 epicure-webinar]$ singularity run ubuntu.sif
Hello EPICURE!
```

- Building a container requires root (or fakeroot) privileges, but running it does not.

- Building the Singularity Container from Definition File.

```
[teop@vglogin0008 epicure-webinar]$ singularity build --fakeroot ubuntu.sif ubuntu.def
INFO:    Starting build...
INFO:    Fetching OCI image...
28.4MiB / 28.4MiB [=============================================================================================
INFO:    Extracting OCI image...
INFO:    Inserting Singularity configuration...
INFO:    Running post scriptlet
+ apt update
```

Running Commands in a Container

- Interactive shell inside the writable container.

```
[teop@vglogin0008 epicure-webinar]$ singularity shell --fakeroot --writable  ubuntu
WARNING: nv files may not be bound with --writable
WARNING: --writable applies to temporary sandbox only, changes will not be written to the original image.
INFO:    Converting SIF file to temporary sandbox...
```

- Interactive shell inside the container.

```
[teop@vglogin0008 epicure-webinar]$ singularity shell ubuntu.sif
Singularity> whoami
teop
```

- Inspect the container (metadata, def,..).

```
[teop@vglogin0008 epicure-webinar]$ singularity inspect --deffile ubuntu.sif
Bootstrap: docker
From: ubuntu:24.04

%post
    apt update && apt install -y python3 pip
    #pip install torch

%runscript
    echo "Hello EPICURE!"
```

- Execute commands inside a container.

```
[teop@vglogin0008 epicure-webinar]$ singularity exec --nv ubuntu.sif nvidia-smi
Sat Mar 29 21:30:09 2025
+-----------------------------------------------------------------------------------------+
| NVIDIA-SMI 565.57.01              Driver Version: 565.57.01      CUDA Version: 12.7     |
|-----------------------------------------+------------------------+----------------------+
| GPU  Name                 Persistence-M | Bus-Id          Disp.A | Volatile Uncorr. ECC |
| Fan  Temp   Perf          Pwr:Usage/Cap |           Memory-Usage | GPU-Util  Compute M. |
|                                         |                        |               MIG M. |
|=========================================+========================+======================|
|   0  NVIDIA A100-PCIE-40GB          On  |   00000000:27:00.0 Off |                    0 |
| N/A   26C    P0             33W /  250W |       4MiB /  40960MiB |      0%      Default |
|                                         |                        |             Disabled |
+-----------------------------------------+------------------------+----------------------+

+-----------------------------------------------------------------------------------------+
| Processes:                                                                              |
|  GPU   GI   CI        PID   Type   Process name                              GPU Memory |
|        ID   ID                                                               Usage      |
|=========================================================================================|
|  No running processes found                                                             |
+-----------------------------------------------------------------------------------------+
```

Convert it

- Docker container can be “converted” to Singularity container

```
[teop@vglogin0008 ~]$ singularity pull ubuntu.sif docker://ubuntu:latest
INFO:    Converting OCI blobs to SIF format
INFO:    Starting build...
INFO:    Fetching OCI image...
28.4MiB / 28.4MiB [==========================================================================================] 100 % 592.2 KiB/s 0s
INFO:    Extracting OCI image...
INFO:    Inserting Singularity configuration...
INFO:    Creating SIF file...
```

- Convert Singularity image (.sif) to writable image (sandbox), and vice versa.

```
[teop@vglogin0006 ~]$ singularity build --fakeroot ubuntu/ ubuntu.sif
INFO:    Starting build...
INFO:    Verifying bootstrap image ubuntu.sif
WARNING: integrity: signature not found for object group 1
WARNING: Bootstrap image could not be verified, but build will continue.
INFO:    Extracting local image...
INFO:    Inserting Singularity configuration...
INFO:    Creating SIF file...
INFO:    Build complete: ubuntu/
```

spython

- Singularity Python API,
- working with Singularity containers,
- easy installation,
- managing converts from “Dockerfile”,
- and beyond.

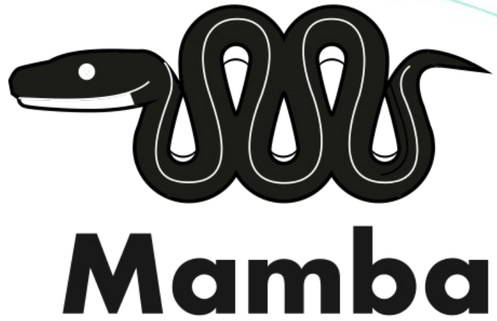
https://singularityhub.github.io/singularity-cli/

Managing Files and Data

- In HPC environments where file I/O speed is critical.
  - Many small files may stress parallel system,
  - single container file can improve performance,
  - optimize your code to minimize file system accesses,
  - advised to use environments.
- Use bind mounts to access large directories on the host system.
  - Multiple binds per container,
  - automatic bind of home directory, and local files,
  - mount an additional layer (OverlayFS) over the existing Singularity container.
- Large-scale storage in HPC environments,
  - Leverage scratch directories,
  - local storage (/tmp) for high-performance I/O.

Package managers in Containers

- Setting up Python environments inside a Singularity container.
- Avoid re-downloading package. Cache persist between runs.



- Each has its advantages and trade-offs depending on your HPC workload!

Pip Installs Packages (pip)

- Pip is a lightweight way to install Python packages in container.

- Suitable for smaller environments.
  - Install packages from requirements file.

- Example of installing pip within Singularity Definition file:

```
%post
    python3 -m pip install --upgrade pip
    pip install numpy pandas scipy
```

https://pypi.org/project/pip/

Conda

- Conda & miniconda are powerful package manager that can handle Python and system dependencies.

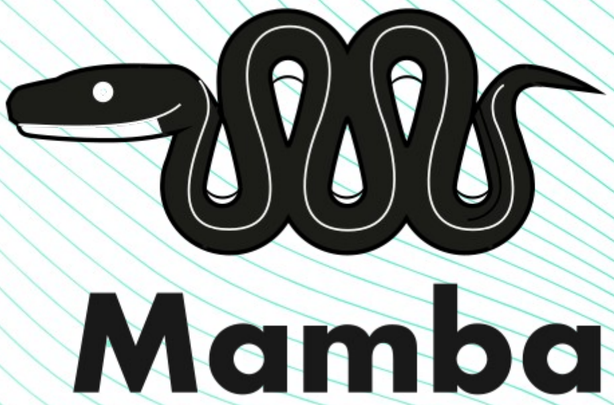
- Suitable for complex environments.
  - Create, activate, deactivate, list environments.
  - Create environment from yaml files.

- Example of installing Conda within Singularity Definition file:

```
%post
    apt update && apt install -y wget bzip2
    wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh -O /miniconda.sh
    bash /miniconda.sh -b -p /opt/miniconda && rm /miniconda.sh
    /opt/miniconda/bin/conda init bash
    /opt/miniconda/bin/conda create -y -n myenv python=3.9 numpy pandas scipy
%environment
    export PATH="/opt/miniconda/bin:$PATH"
    source activate myenv
```

https://docs.conda.io/en/latest/

Mamba



- Mamba is a reimplementation and drop-in replacement for Conda that is much faster.

- Suitable for complex environments.
  - Avoid updating existing packages --freeze-installed

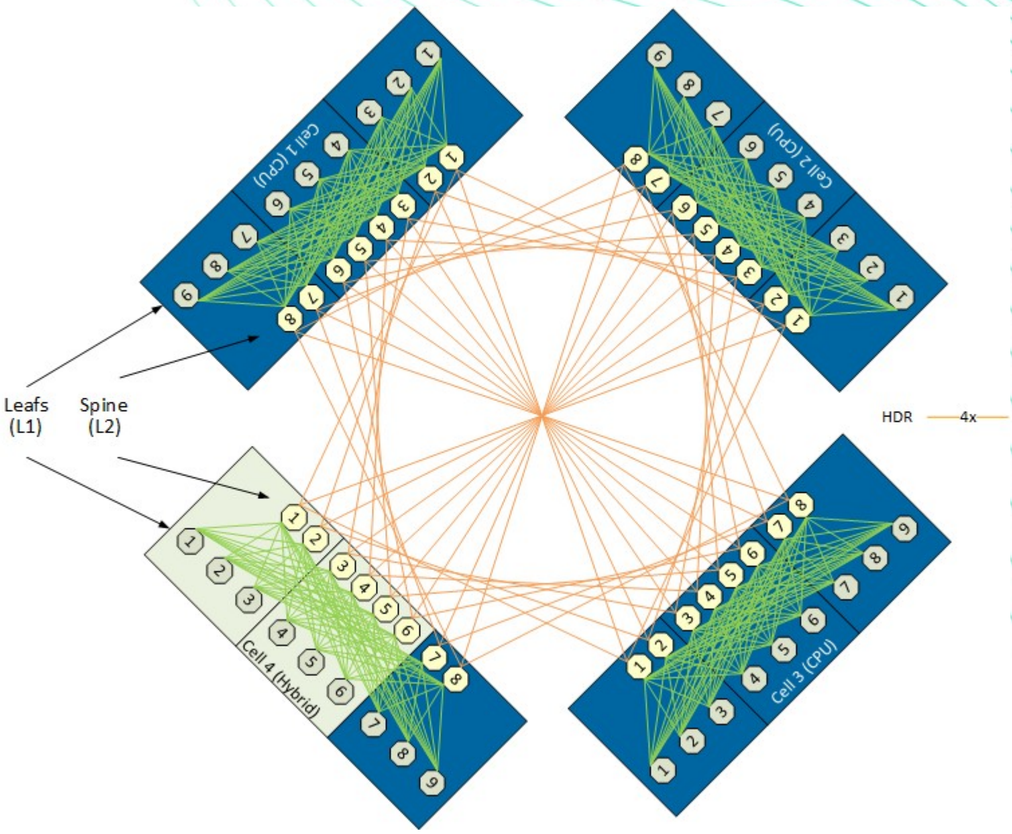
- Example of installing Mamba within Singularity Definition file:

```
%post
    apt update && apt install -y wget bzip2
    wget https://github.com/mamba-org/micromamba-releases/releases/latest/download/micromamba-linux-64 -O /usr/local/bin/micromamba
    chmod +x /usr/local/bin/micromamba
    export MAMBA_ROOT_PREFIX=/opt/micromamba
    micromamba shell init --shell=bash
    micromamba create -y -n epicureenv python=3.X numpy pandas scipy
%environment
    export MAMBA_ROOT_PREFIX=/opt/micromamba
    export PATH="/opt/micromamba/bin:$PATH"
    micromamba activate epicureenv
```

https://github.com/mamba-org/mamba

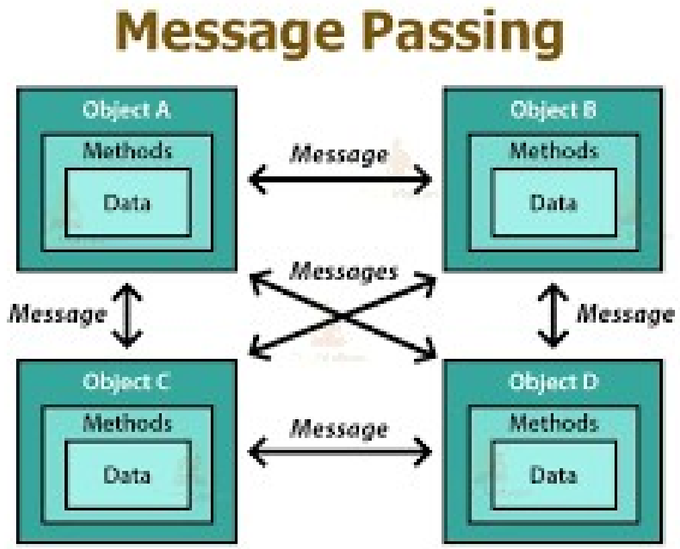
Interconnectivity

- Running Singularity containers with InfiniBand, Slingshot,..
- Network is not isolated, its shared!
  - resources, files, memory, …
  - If InfiniBand is detected, it will show active ports.
  - Use OSU MPI micro-benchmarks to verify it.
  - Build Slurm with PMIX & UCX for better performance!
    - srun --mpi=list
- Requires special configurations to ensure
  - Low Latency & High Bandwidth
  - GPUDirect RDMA Support
  - Allows direct GPU-to-GPU communication
    - Use --nv for GPU support (if applicable).
  - Scalability (Enables multi-node, multi-GPU training.



Message Passing Interface (MPI)

- MPI is used for parallelisation in HPC environments.
  - Slurm integration (UCX & PMIx).
  - MPI requires direct host communication.
  - CUDA-Aware MPI (MPI builded with CUDA).
  - Alternatives i.e. MPICH, Intel MPI,..
- Running MPI applications inside containers can be tricky.
  - Networking, process management, and GPU compatibility.
  - MPI versions inside the container should match the host version!



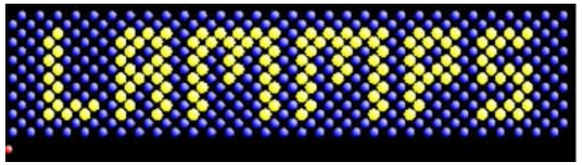
https://shorturl.at/EVAlw

Workloads

- Gromacs (GPU + MPI)

```
singularity run --nv -B /ceph/hpc/software/bench/water-cut1.0_GMX50_bare/0012:/host                \
                     --pwd /host /ceph/hpc/software/containers/singularity/images/gromacs-gpu.sif \
                     mpirun -np 4 gmx mdrun -nb gpu -pin on -v -noconfout -s topol.tpr
```

- LAMMPS (CPU + MPI)



```
mpirun -np 128 --oversubscribe --use-hwthread-cpus --map-by hwthread --bind-to core \
       singularity exec /ceph/hpc/software/containers/singularity/images/lammps.sif \
       lmp_mpi -var x 10 -var y 40 -var z 40 -in in.lj
```

- QE (GPU + MPI)

```
export OMP_NUM_THREADS=$SLURM_CPUS_PER_TASK
srun --mpi=pmi2 singularity exec --nv /ceph/hpc/software/containers/singularity/images/qe-gpu.sif pw.x -inp ausurf.in
```

- mpiBench (Hybrid MPI + OMP)

```
singularity exec mpiBench.sif mpirun -x OMP_NUM_THREADS=8 -x OMP_PLACES=cores -n 16 --map-by node:PE=8 --bind-to core mpiBench
```

  - Run it via srun and specify MPI interface, mpirun outside container, or within container!
  - Debug it via logs, verbose mode, report-bindings, gdb, valgrind,..
  - Profiling tools LIKWID, Score-P, TotalView, Intel VTune, Nsight Systems (nways),..

Workloads

- TensorFlow Benchmarks

```
git clone https://github.com/tensorflow/benchmarks.git && cd benchmarks/scripts/tf_cnn_benchmarks/

SINGULARITYENV_CUDA_VISIBLE_DEVICES=0,1 singularity run --nv -B /root/bench/benchmarks/scripts/tf_cnn_benchmarks/:/host \
--pwd /host /ceph/hpc/software/containers/singularity/images/tensorflow.sif python3 all_reduce_benchmark.py --num_gpus=2 \
--batch_size=128 --variable_update=replicated --nouse_fp16 --weight_decay=1e-4 --gradient_repacking=1
```

- PyTorch, transformers, and dependencies with CUDA support.

```
Bootstrap: docker
From: nvcr.io/nvidia/pytorch:25.03-py3
%post
    apt update -y
    pip install --upgrade pip
    pip install transformers datasets torch torchvision jupyter
%environment
    export PATH="/opt/conda/bin:$PATH"
%runscript
    python3 -c "import torch; print(torch.cuda.is_available())"
```

Workloads

- Building the Container (Requires Root Privileges)

```
[teop@vglogin0008 test]$ singularity build --fakeroot pytorch.sif pytorch.def
INFO:    Starting build...
INFO:    Fetching OCI image...
```

- Check CUDA

```
[teop@vglogin0008 epicure-webinar]$ singularity exec --nv pytorch.sif python3 -c "import torch; print(torch.cuda.is_available())"
True
```

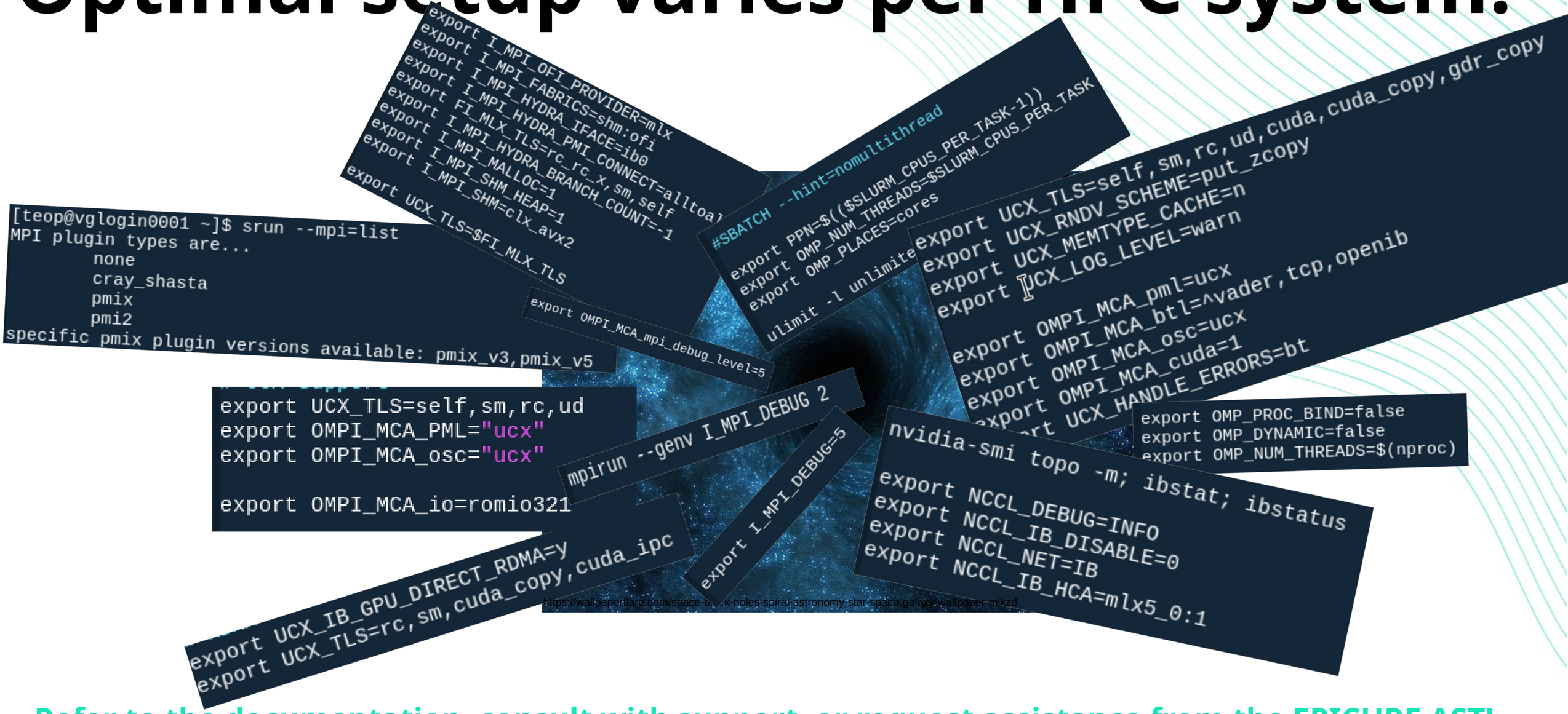
- Interactive work with Jupyter Notebook with GPUs

```
[teop@vglogin0008 epicure-webinar]$ singularity exec --nv pytorch.sif jupyter notebook --ip=0.0.0.0 --port=8888 --no-browser
```

```
[I 2025-03-27 14:07:07.621 ServerApp] notebook | extension was successfully loaded.
[I 2025-03-27 14:07:07.622 ServerApp] Serving notebooks from local directory: /ceph/hpc/home/teop/epicure-webinar
[I 2025-03-27 14:07:07.622 ServerApp] Jupyter Server 2.15.0 is running at:
[I 2025-03-27 14:07:07.622 ServerApp] http://hostname:8888/tree?token=36ec46368b7ed8df56a7e66d0d7dadddb7df908459669f85
[I 2025-03-27 14:07:07.622 ServerApp]     http://127.0.0.1:8888/tree?token=36ec46368b7ed8df56a7e66d0d7dadddb7df908459669f85
[I 2025-03-27 14:07:07.622 ServerApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
```

- Run it through job scheduler!

Optimal setup varies per HPC system!



Refer to the documentation, consult with support, or request assistance from the EPICURE AST!

Workloads and Optimizations

- Use optimized pre-build Images, then build on top of them.
- Use “--bind” efficiently to mount directories without redundancy.
  - Use host-pinned MPI & IB libraries for performance.
  - Bind mount necessary directories for InfiniBand support.
- Leverage Slurm, PMIx, and UCX for better MPI scaling.
- Running GPU-accelerated workloads using container (Multi GPUs & nodes)
  - CUDA, OpenCL, ROCm (--rocm), and NVIDIA (--nv)
  - --with ‘MPI, MPICH, CUDA, NCCL, CUDA-aware MPI’
- Use SquashFS & SIF images for reduced I/O overhead.
- Enable CPU/memory affinity for HPC workloads.
- Set “--containall” or “--writable-tmpfs” to isolate workloads.
  - prevent conflicts!
- Pre-cache libraries to reduce runtime loading delays.
  - LD_LIBRARY_PATH

.

Security: Best Practices

- Security must be embedded at all levels of the software,
- proper vulnerability and patch management,
- scan images and dependencies for vulnerabilities,
  - CVE vulnerabilities, misconfigurations, secrets in container images.
- use Apparmor, SeLinux, Seccomp,
- keep images updated, minimize the number of installed packages,
- use images from trusted sources, sign images,
  - NVIDIA NGC, Apptainer Library, DockerHub,..

- verify the integrity of the container image, verify signatures,
- immutable Singularity Image Format (sif) prevents unauthorized modifications.
- implement access controls,
  - integration with CI/CD pipelines and tools for automated vulnerability scanning.
- monitor activity on the host, auditing,
- use unprivileged user namespaces.

Trivy

A lightweight container scanner for Docker Singularity containers for vulnerabilities and security risks.


```
teop@hpcteopnb:~/Downloads/epicure-webinar$ trivy fs ubuntu-latest
2025-03-28T23:28:56.686+0100    INFO    Detected OS: ubuntu
2025-03-28T23:28:56.686+0100    WARN    This OS version is not on the EOL list: ubuntu 24.04
2025-03-28T23:28:56.686+0100    INFO    Detecting Ubuntu vulnerabilities...
2025-03-28T23:28:56.686+0100    INFO    Number of language-specific files: 0
2025-03-28T23:28:56.686+0100    WARN    This OS version is no longer supported by the distribution: ubuntu 24.04
2025-03-28T23:28:56.686+0100    WARN    The vulnerability detection may be insufficient because security updates are not provided

localhost.localdomain (ubuntu 24.04)
====================================
Total: 0 (UNKNOWN: 0, LOW: 0, MEDIUM: 0, HIGH: 0, CRITICAL: 0)
```

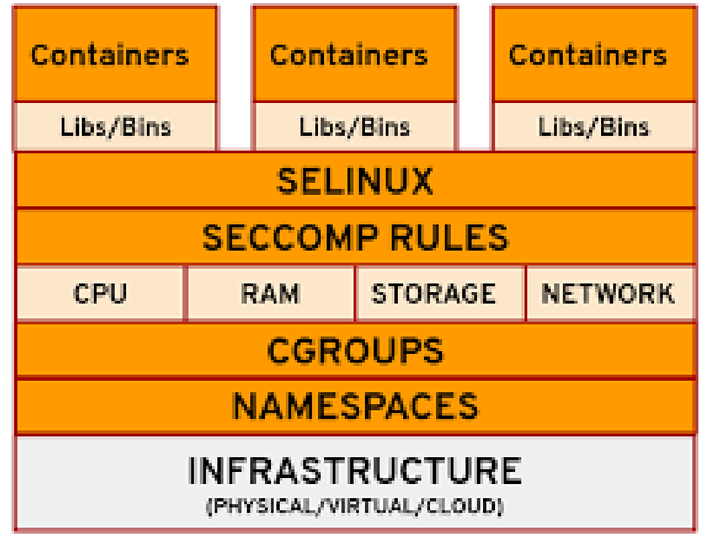
```
teop@hpcteopnb:~/Downloads/epicure-webinar$ trivy fs ubuntu-20.04
2025-03-28T23:26:24.587+0100    INFO    Detected OS: ubuntu
2025-03-28T23:26:24.587+0100    INFO    Detecting Ubuntu vulnerabilities...
2025-03-28T23:26:24.590+0100    INFO    Number of language-specific files: 0

localhost.localdomain (ubuntu 20.04)
====================================
Total: 19 (UNKNOWN: 0, LOW: 17, MEDIUM: 2, HIGH: 0, CRITICAL: 0)

+--------------+------------------+----------+---------------------------+---------------+---------------------------------------+
|   LIBRARY    | VULNERABILITY ID | SEVERITY |    INSTALLED VERSION      | FIXED VERSION |                 TITLE                 |
+--------------+------------------+----------+---------------------------+---------------+---------------------------------------+
| coreutils    | CVE-2016-2781    | LOW      | 8.30-3ubuntu2             |               | coreutils: Non-privileged             |
|              |                  |          |                           |               | session can escape to the             |
|              |                  |          |                           |               | parent session in chroot              |
|              |                  |          |                           |               | -->avd.aquasec.com/nvd/cve-2016-2781  |
+--------------+------------------+          +---------------------------+---------------+---------------------------------------+
| gpgv         | CVE-2022-3219    |          | 2.2.19-3ubuntu2.2         |               | gnupg: denial of service              |
|              |                  |          |                           |               | issue (resource consumption)          |
|              |                  |          |                           |               | using compressed packets              |
|              |                  |          |                           |               | -->avd.aquasec.com/nvd/cve-2022-3219  |
+--------------+------------------+          +---------------------------+---------------+---------------------------------------+
| libc-bin     | CVE-2016-20013   |          | 2.31-0ubuntu9.16          |               | sha256crypt and sha512crypt           |
|              |                  |          |                           |               | through 0.6 allow attackers           |
|              |                  |          |                           |               | to cause a denial of...               |
|              |                  |          |                           |               | -->avd.aquasec.com/nvd/cve-2016-20013 |
+--------------+                  +          +                           +---------------+                                       +
| libc6        |                  |          |                           |               |                                       |
|              |                  |          |                           |               |                                       |
|              |                  |          |                           |               |                                       |
|              |                  |          |                           |               |                                       |
+--------------+------------------+          +---------------------------+---------------+---------------------------------------+
| libncurses6  | CVE-2021-39537   |          | 6.2-0ubuntu2.1            |               | ncurses: heap-based buffer overflow   |
|              |                  |          |                           |               | in _nc_captoinfo() in captoinfo.c     |
|              |                  |          |                           |               | -->avd.aquasec.com/nvd/cve-2021-39537 |
+              +------------------+          +                           +---------------+---------------------------------------+
|              | CVE-2022-29458   |          |                           |               | ncurses: segfaulting OOB read         |
```

Namespaces and Cgroups

- Namespace is a Linux kernel feature.
- Allow per each namespace mapping of UIDs and GIDs.
- Enable setting up a container without privileged operations.
- User in container can perform task without being root.
- Provide a layer and resource isolation.
- Namespaces: PID, Network, IPC, Mount, UTS, User, ..
- Cgroup limit and allocate resources.



https://opensource.com/article/21/8/container-linux-technology

”cgroups are used to manage resources, namespaces are used for isolation of processes!”

Containers at VEGA

User namespaces Network namespaces

- Cgroups (v1) enabled in Slurm (ensure the resource limitations)
- SingularityPRO is installed on the compute and login nodes
  - Login nodes are suitable for container preparation as they are equipped with GPUs (4x A100).
- Rootless containers
  - fakeroot feature
  - assigned by request, not by default.
- Definition files of our containers are available to users.
- Embedded containers within ARC-CE RTEs.
- Template for OFED container image is available for users to build on top of it.
- Work in progress..
  - Automatic container (image) deployment, security checks CI/CD,..
  - images will be available in national (SLING) image registry based on CernVM-FS

VEGA
HPC

Containers have come a long way!

- Highly accepted in HPC environments, and beyond!
- Keep minimal, lightweight, and maintainable image.
- Keep security checks in place!
  - Keep sensitive data out of container.
- Handling many small files, use environments to reduce system stress!
- Bind-mount
  - Bind data into container, rather that copied it.
- Latest publications reject the common misconceptions!

EPICURE

VEGA HPC

IZUM
Institute of Information Science, Maribor, Slovenia

Jožef Stefan Institute

Alja Prah

Sebastien Strban

Dejan Lesjak

Žiga Zebec

Samo Miklavc

Co-funded by the European Union

EuroHPC Joint Undertaking

This project has received funding from the European High Performance Computing Joint Undertaking under grant agreementNo.101139786. Views and opinions expressed are, however, those of the author(s) only and do not necessarily reflect those of the European Union orEuroHPC Joint Undertaking. Neither the European Union nor the granting authority can be held responsible for them.

References

- [1] EuroHPC Vega Documentation. https://doc.vega.izum.si/
- [2] Krasovec, B., Prica, T.: Secure usage of containers in the hpc environment. In:
- Nordic e-Infrastructure Tomorrow, pp. 96–112. Springer, Cham (2025)
- [3] Lesjak, D., Prah, A., Krasovec, B.: From isolation to integration: A decade of con-
- tainer technology in slovenian HPC. In: Nordic e-Infrastructure Tomorrow, pp. 142–151. Springer, Cham (2025)

[4] Prica, T.: Development and supporting activities on EuroHPC Vega. ASHPC24 p. 14 (2024)

[5] Prah, A., Krasovec B.: Overcoming challenges in building scientific software for HPC. ASHPC24 p. 16 (2024)

EPICURE
Unlocking European-level HPC Support

Thank you!

Follow us

pmo-epicure@postit.csc.fi

teo.prica@izum.si

Co-funded by the European Union

EuroHPC Joint Undertaking

This project has received funding from the European High Performance Computing Joint Undertaking under grant agreementNo.101139786. Views and opinions expressed are, however, those of the author(s) only and do not necessarily reflect those of the European Union orEuroHPC Joint Undertaking. Neither the European Union nor the granting authority can be held responsible for them.