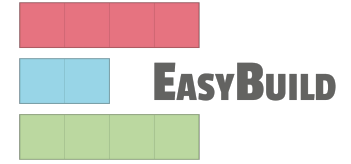


Prepared environment



- This information (and more!) is available at <https://hackmd.io/@multixscale/VSC>
- Access via ssh
 - Shell access: `ssh -t <username>@vmos.vsc.ac.at l55`
 - 2 times password + OTP
- Clone the repositories we will use during the tutorial
 - `git clone https://github.com/EESSI/eessi-demo.git`
 - `git clone https://github.com/EESSI/eessi-nextflow-example.git`
- A reservation will be used to do the hands-on parts
 - See shared document for the full command as it is a big one!



Introduction to E E S S I

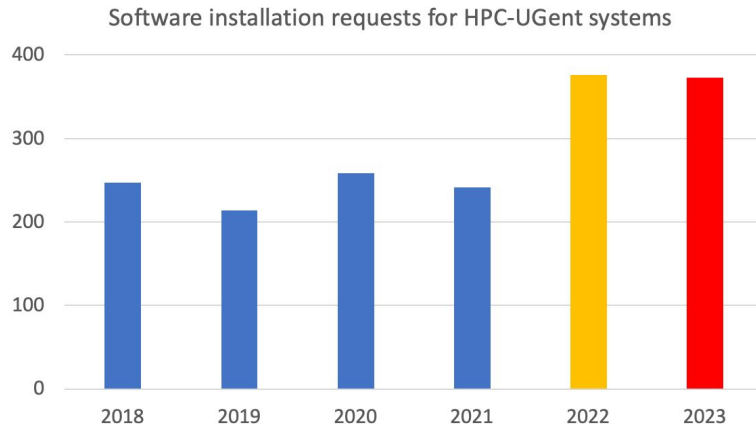
EUROPEAN ENVIRONMENT FOR
SCIENTIFIC SOFTWARE INSTALLATIONS

Alan O'Cais
University of Barcelona/CECAM
alan.ocais@cecam.org



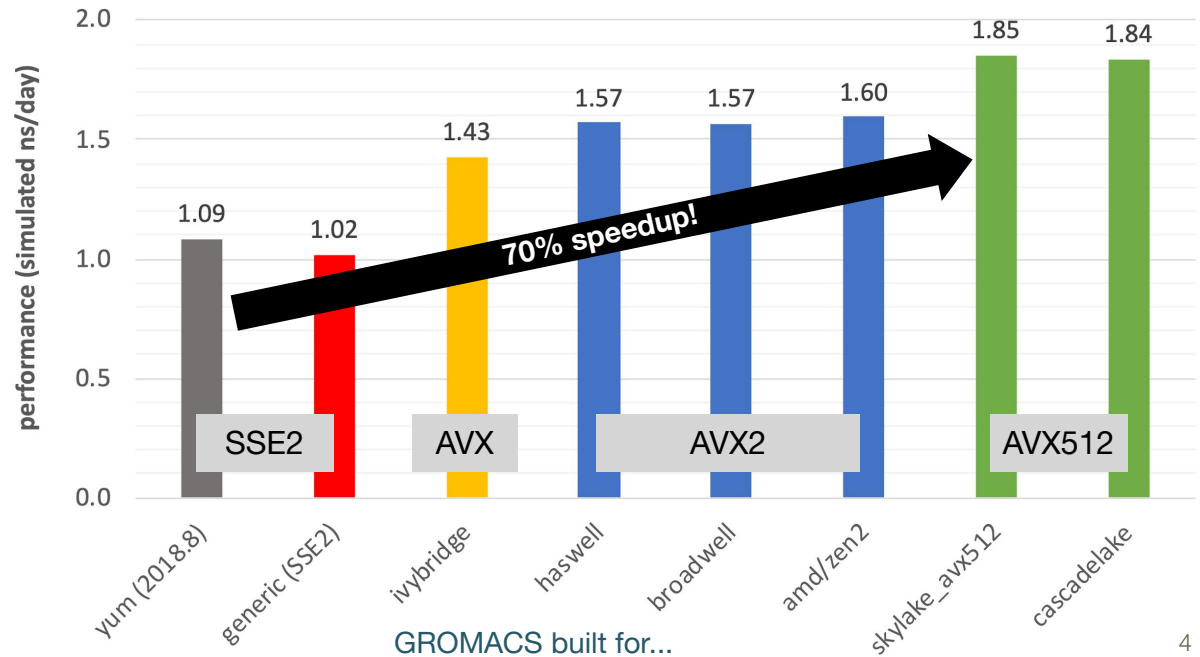
The changing landscape of scientific computing

- **Explosion of available scientific software** applications (bioinformatics, AI boom, ...)
- Increasing interest in **cloud** for scientific computing (flexibility!)
- **Increasing variety in processor (micro)architectures** beyond Intel & AMD: Arm is coming already here (see [Fugaku](#), [JUPITER](#), ...), RISC-V is coming (soon?)
- In strong contrast: available (wo)manpower **in HPC support teams is (still) limited...**



Optimized scientific software installations

- Software should be optimized for the system it will run on
- Impact on performance is often significant for scientific software
- Example: GROMACS 2020.1 (PRACE benchmark, Test Case B)
- Metric: (simulated) ns/day, higher is better
- Test system: dual-socket Intel Xeon Gold 6420 (Cascade Lake, 2x18 cores)
- Performance of different GROMACS binaries, on exact same hardware/OS



EESSI in a nutshell

- *European Environment for Scientific Software Installations (EESSI)*
- **Shared repository of (optimized!) scientific software installations**
- Avoid duplicate work across (HPC) sites by collaborating on a shared software stack
- Uniform way of providing software to users, regardless of the system they use!
- Should work on any Linux OS (+ WSL, macOS) and system architecture
 - From laptops and personal workstations to HPC clusters and cloud
 - Support for different CPUs, interconnects, GPUs, etc.
- **Focus on performance, automation, testing, collaboration**



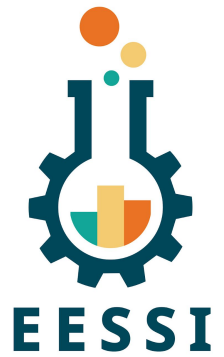
E E S S I

EUROPEAN ENVIRONMENT FOR
SCIENTIFIC SOFTWARE INSTALLATIONS

<https://www.eessi.io>

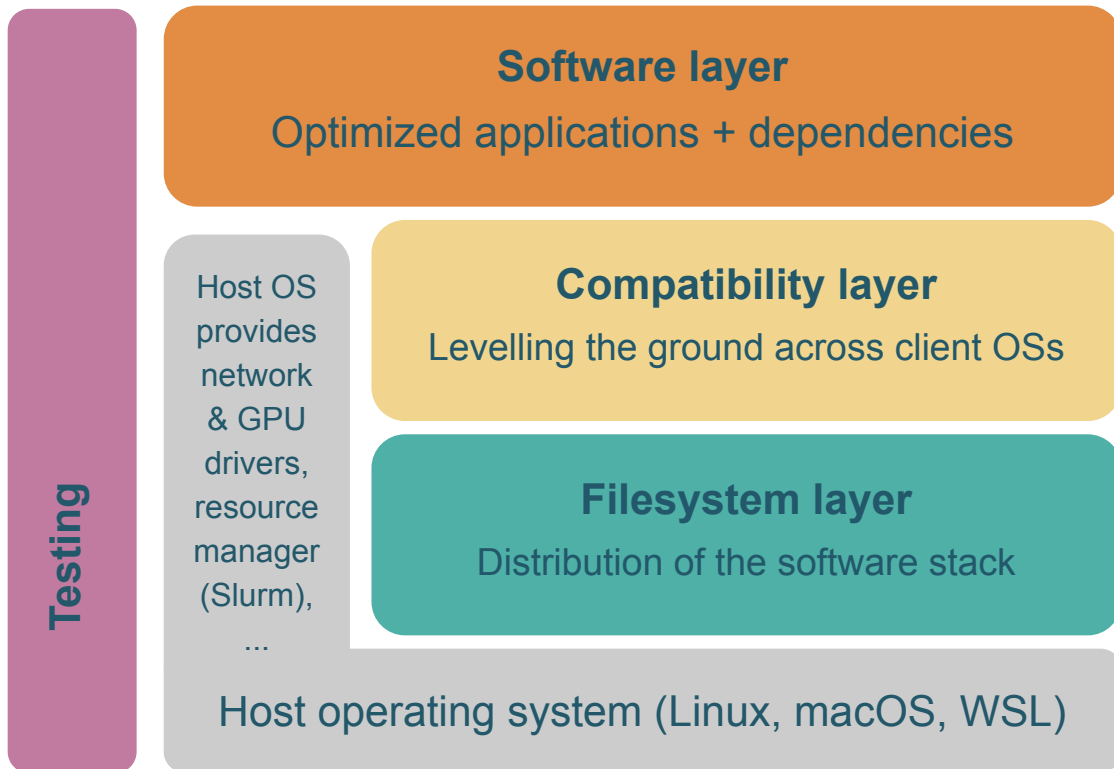
<https://www.eessi.io/docs/>

Major goals of EESSI

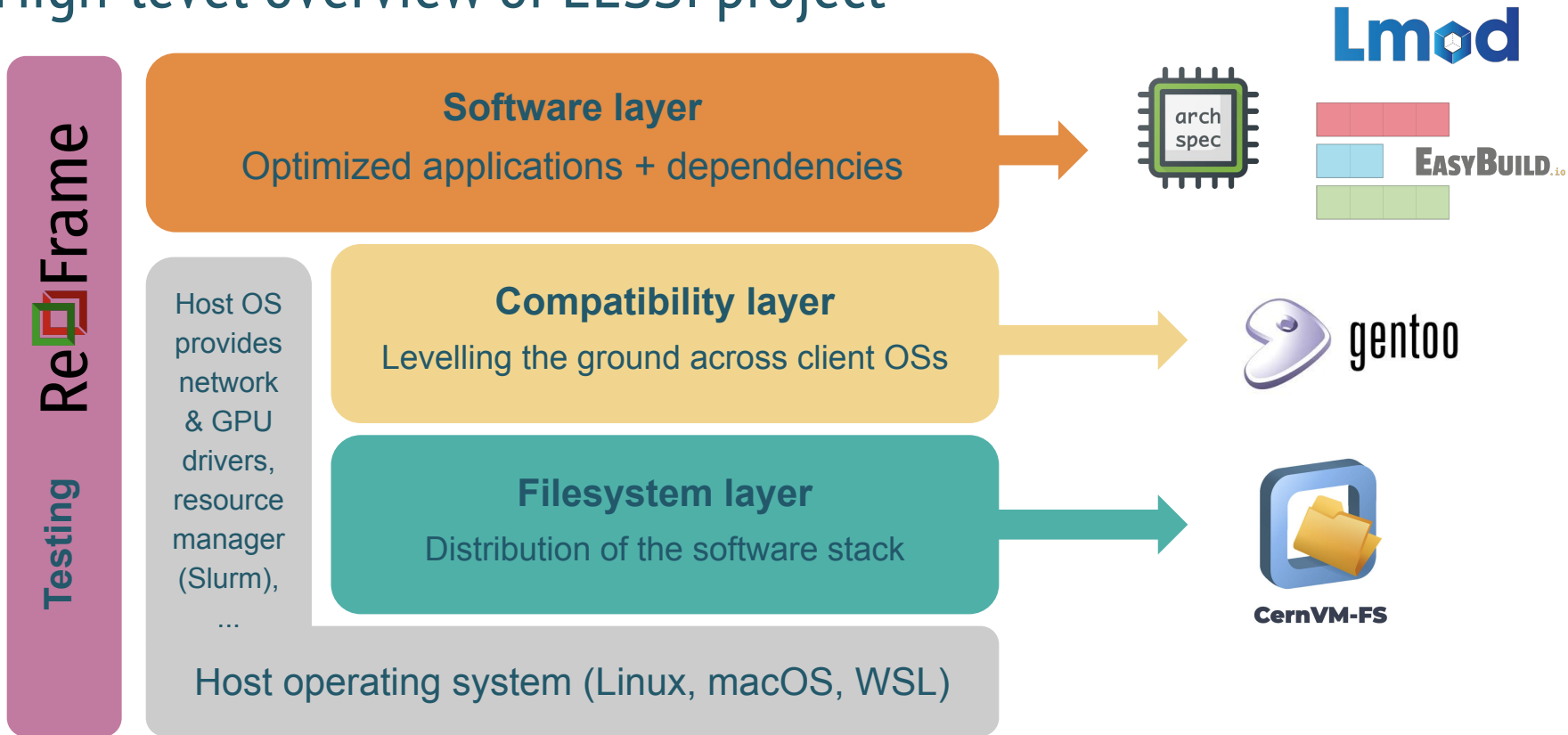


- **Avoid duplicate work** (for researchers, HPC support teams, ...)
 - Tools that automate software installation process (EasyBuild, Spack) are not sufficient
 - Go beyond sharing build recipes => work towards a shared software stack
- Providing a truly **uniform software stack**
 - Use the (exact) same software environment everywhere
 - Without sacrificing performance for “mobility of compute” (like is typically done with containers/conda)
- Facilitate HPC training, development of (scientific) software, ...

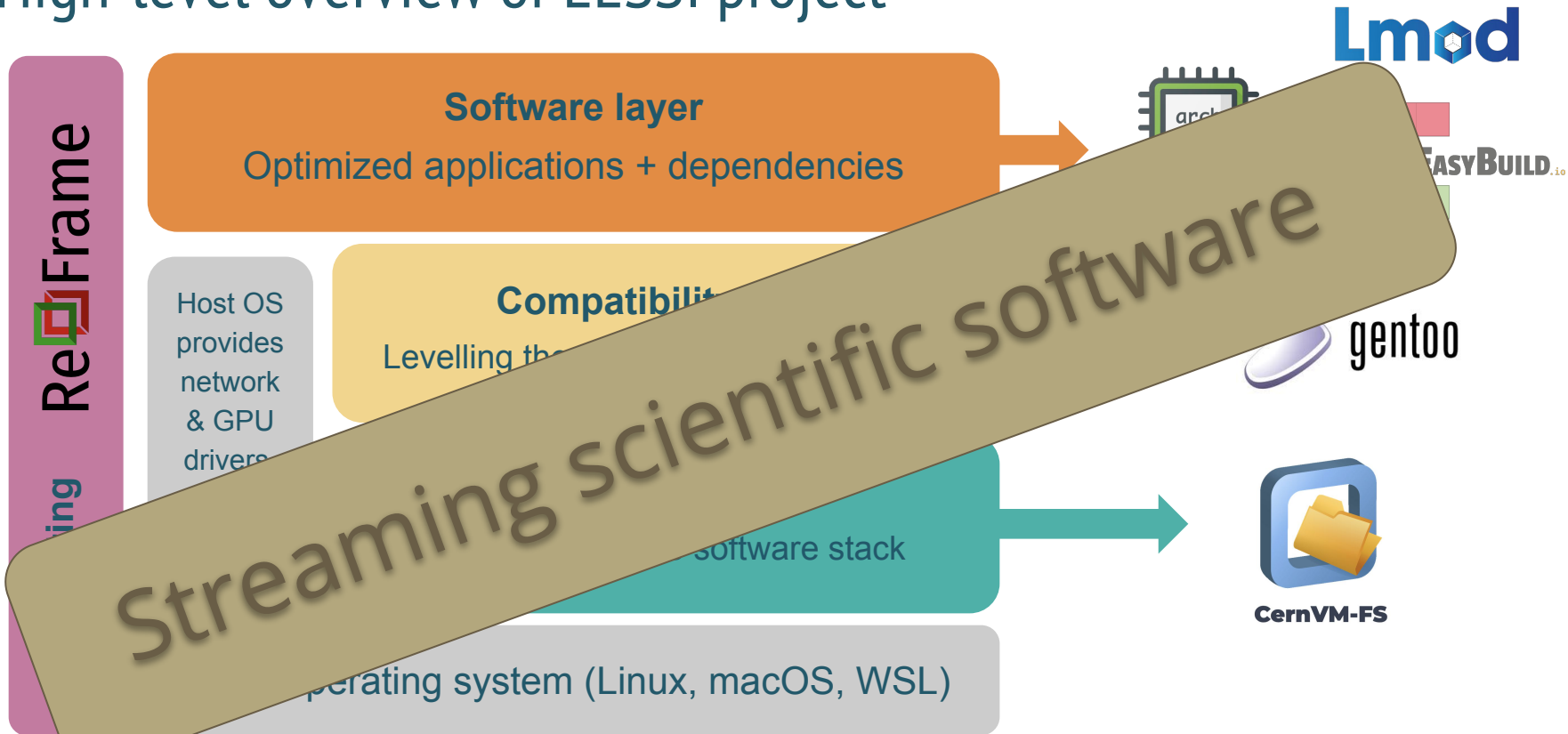
High-level overview of EESSI project



High-level overview of EESSI project



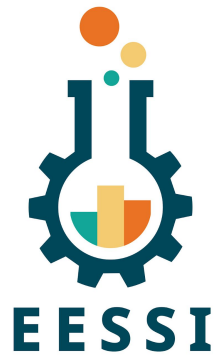
High-level overview of EESSI project





DEMO

Demo scenario



- Demo: Using an “empty” **Amazon Linux 2023** VM in AWS (**Arm Graviton2**)
 - No CernVM-FS installed, EESSI not available yet, but only takes 2 min.
 - Requires admin rights (`sudo` to install extra packages)
 - Set up EESSI environment by sourcing init script
 - Running EESSI demo script from

<https://github.com/EESSI/eessi-demo>

Demo: Amazon Linux 2023 Arm VM in AWS (1/2)



- We needed to: <https://github.com/EESSI/eessi-demo>
 - **Install CernVM-FS** packages
 - Install **EESSI CernVM-FS configuration** (`cvmfs-eessi-config*` package)
 - Set up **minimal CernVM-FS client configuration** in `/etc/cvmfs/default.local`
- For production usage (especially large-scale), you should also:
 - Use a **squid proxy**, next to a local client cache (better start-up performance)
 - Set up your own **Stratum-1 mirror server** (protection against network disconnects)
 - Also recommended to “be a good citizen” in the EESSI CernVM-FS network

Demo: Amazon Linux 2023 Arm VM in AWS (2/2)

- Once CernVM-FS + EESSI configuration is installed, you're good to go!
- Set up EESSI environment by sourcing the init script, load modules, run.



```
$ ls /cvmfs/software.eessi.io
host_injections  init  README.eessi  versions

$ source /cvmfs/software.eessi.io/versions/2023.06/init/lmod/bash
...
Environment set up to use EESSI (2023.06), have fun!

$ module avail GROMACS TensorFlow OpenFOAM Bioconductor

$ module avail GROMACS TensorFlow OpenFOAM Bioconductor

---- /cvmfs/software.eessi.io/versions/2023.06/software/linux/x86_64/amd/zen3/modules/all ----

GROMACS/2024.1-foss-2023b          R-bundle-Bioconductor/3.16-foss-2022b-R-4.2.2
OpenFOAM/v2312-foss-2023a        R-bundle-Bioconductor/3.18-foss-2023a-R-4.3.2 (D)
OpenFOAM/10-foss-2023a           TensorFlow/2.13.0-foss-2023a
OpenFOAM/11-foss-2023a          (D) ...
```

<https://github.com/EESSI/eessi-demo>



Hands-on 1: Running a demo (10 mins)

Hands-on scenario



- On provided infrastructure
 - EESSI CernVM-FS repository readily available and configured
 - First:

```
git clone https://github.com/EESSI/eessi-demo
```

You should run demos under the EESSI reservation (see <https://hackmd.io/@multixscale/VSC>)

Try out EESSI yourself using Apptainer!

- Only Apptainer (or Singularity) is required to run the EESSI client container
- **Should work on any Linux distribution**, on Intel/AMD/Arm CPUs
- Detailed instructions available at https://www.eessi.io/docs/getting_access/eessi_container



```
$ apptainer shell --fusemount ...
...
Apptainer> ls /cvmfs/software.eessi.io/
host_injections  init  README.eessi  versions

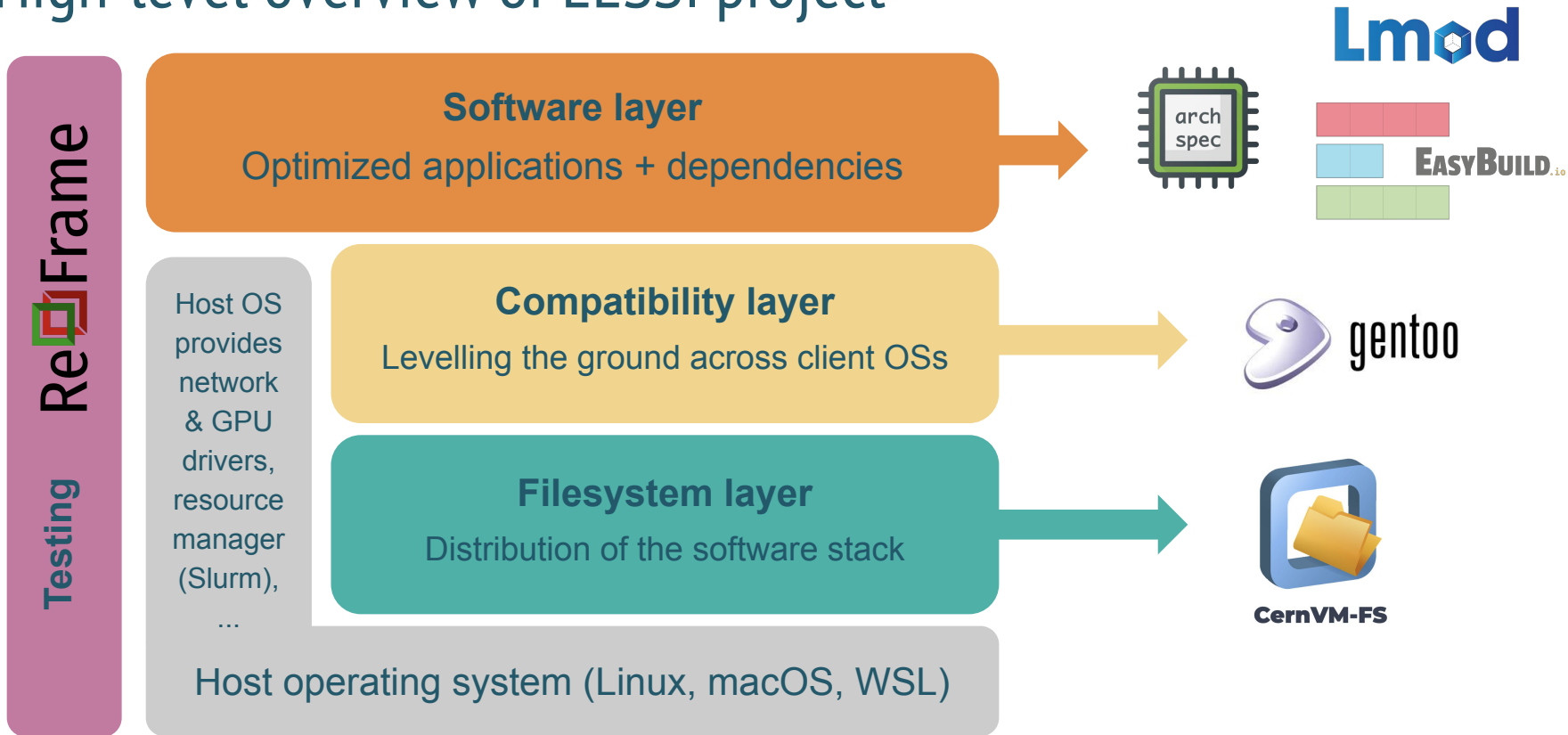
Apptainer> source /cvmfs/software.eessi.io/versions/2023.06/init/bash

Found EESSI repo @ /cvmfs/software.eessi.io/versions/2023.06!
archdetect says x86_64/generic
Using x86_64/generic as software subdirectory.
Using /cvmfs/software.eessi.io/versions/2023.06/software/linux/x86_64/generic/modules/all as the directory to be added
to MODULEPATH.
Found Lmod configuration file at
/cvmfs/software.eessi.io/versions/2023.06/software/linux/x86_64/generic/.lmod/lmodrc.lua
Initializing Lmod...
Prepending /cvmfs/software.eessi.io/versions/2023.06/software/linux/x86_64/generic/modules/all to $MODULEPATH...
Environment set up to use EESSI (2023.06), have fun!
```




**How exactly does it
work?**

High-level overview of EESSI project



Filesystem layer

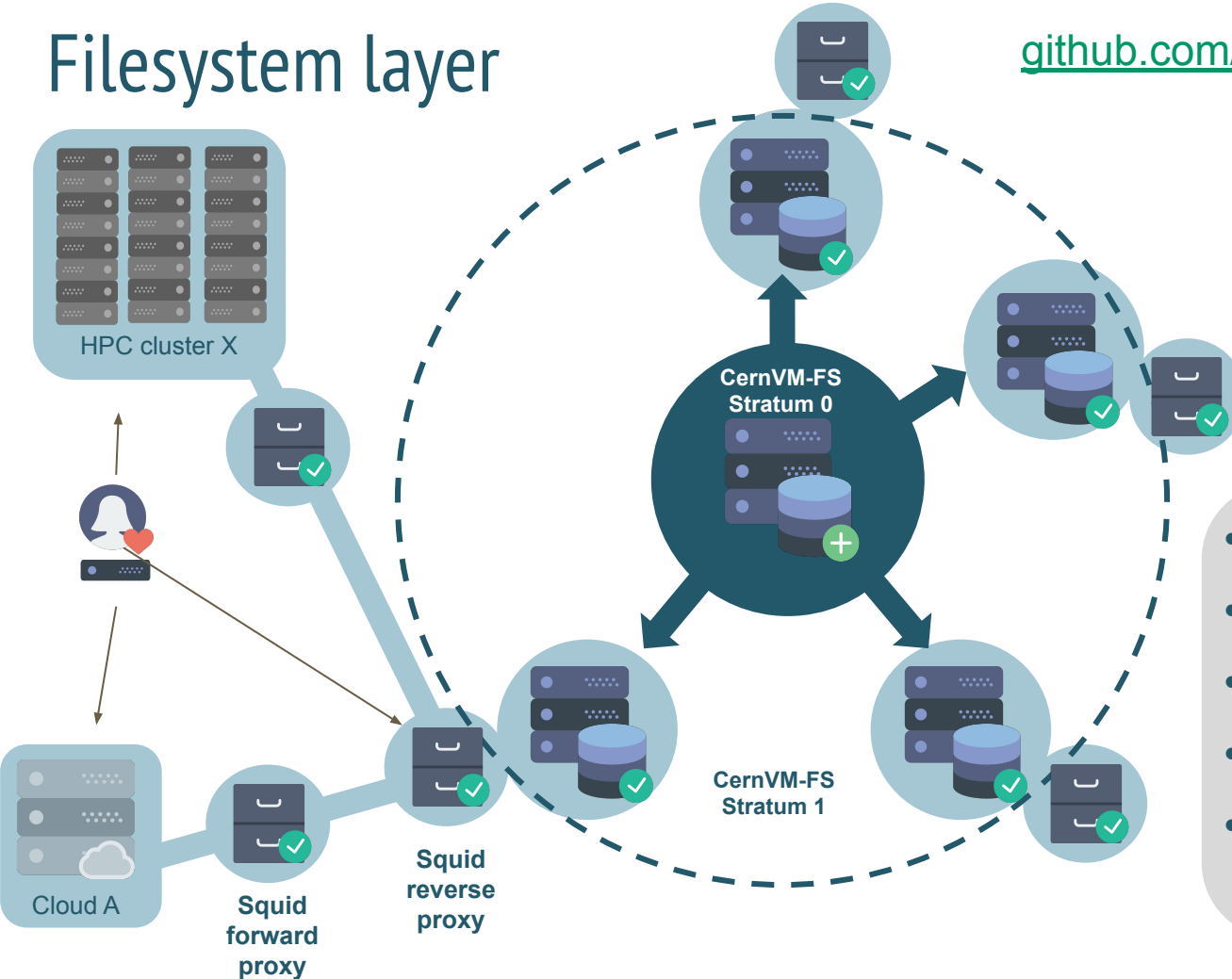
github.com/EESSI/filesystem-layer

(icons via <https://www.flaticon.com/authors/smashicons>)



CernVM-FS

cvmfs.readthedocs.io



- Global distribution of software installations
- Centrally managed software stack
- Redundant network of “mirrors”
- Multiple levels of caching
- **Same software stack everywhere:**
laptops, HPC clusters, cloud VMs, ...

Compatibility layer

github.com/EESSI/compatibility-layer



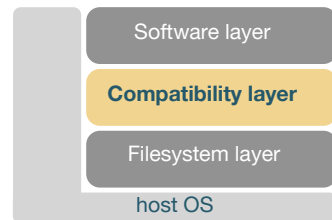
powered by



- **Gentoo Prefix** installation (in `/cvmfs/.../compat/<os>/<arch>/`)
- Set of tools & libraries installed in non-standard location
- Limited to low-level stuff, incl. glibc (no Linux kernel or drivers)
 - Similar to the OS layer in container images
- Only targets a supported processor **family** (`aarch64`, `x86_64`, `riscv64`)
- **Levels the ground for different client operating systems** (Linux distros, also WSL/macOS)
- Currently in production repository:

`/cvmfs/software.eessi.io/versions/2023.06/compat/linux/aarch64`

`/cvmfs/software.eessi.io/versions/2023.06/compat/linux/x86_64`



Software layer

github.com/EESSI/software-layer

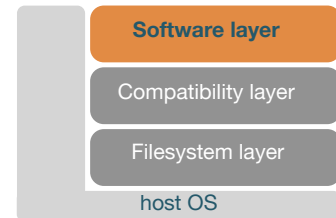
- Provides scientific software applications, libraries, and dependencies
- **Optimized for specific CPU microarchitectures** (AMD Zen3, ...)
 - Separate subdirectory/tree for each (in `/cvmfs/.../software/...`)
- **Leverages libraries** (like glibc) **from compatibility layer** (not from host OS)
- Installed with EasyBuild, incl. environment module files
- Lmod environment modules tool is used to access installations
- **Best subdirectory for host is selected automatically** via `archdetect`



powered by

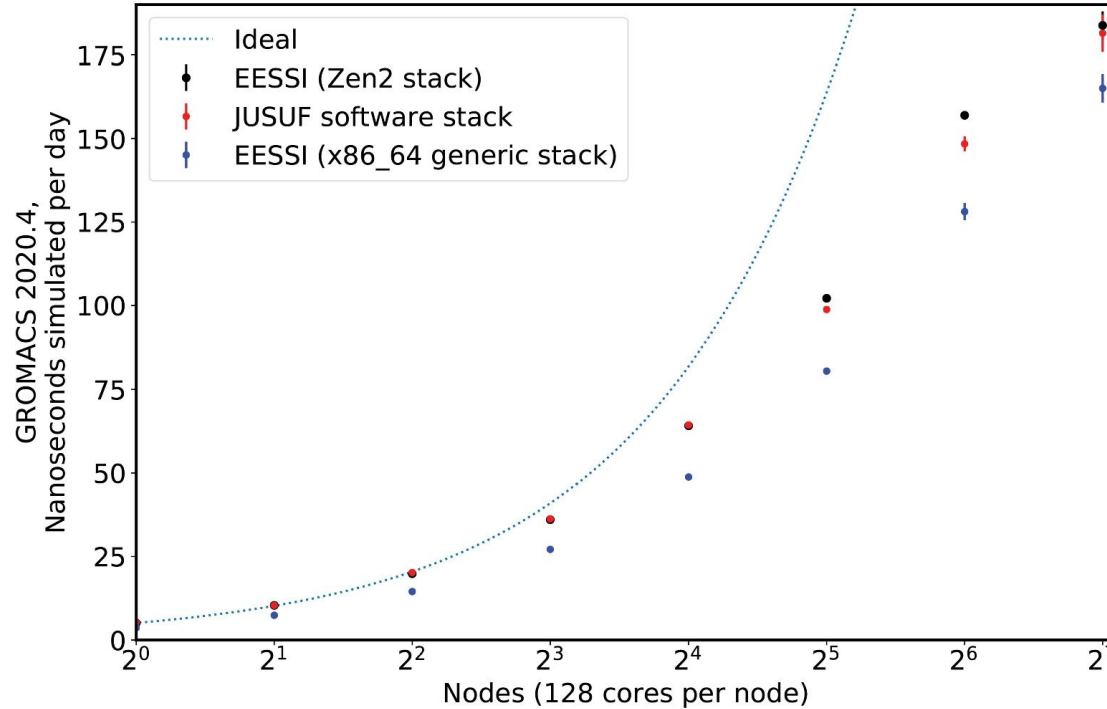
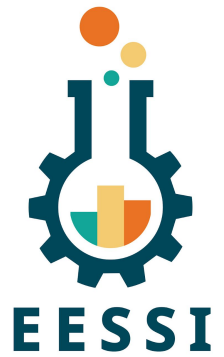


Lmod



EESSI paper (open access)

doi.org/10.1002/spe.3075



Paper includes proof-of-concept performance evaluation compared to system software stack, performed at JUSUF @ JSC using GROMACS 2020.4, up to 16,384 cores (CPU-only)

Current status of EESSI

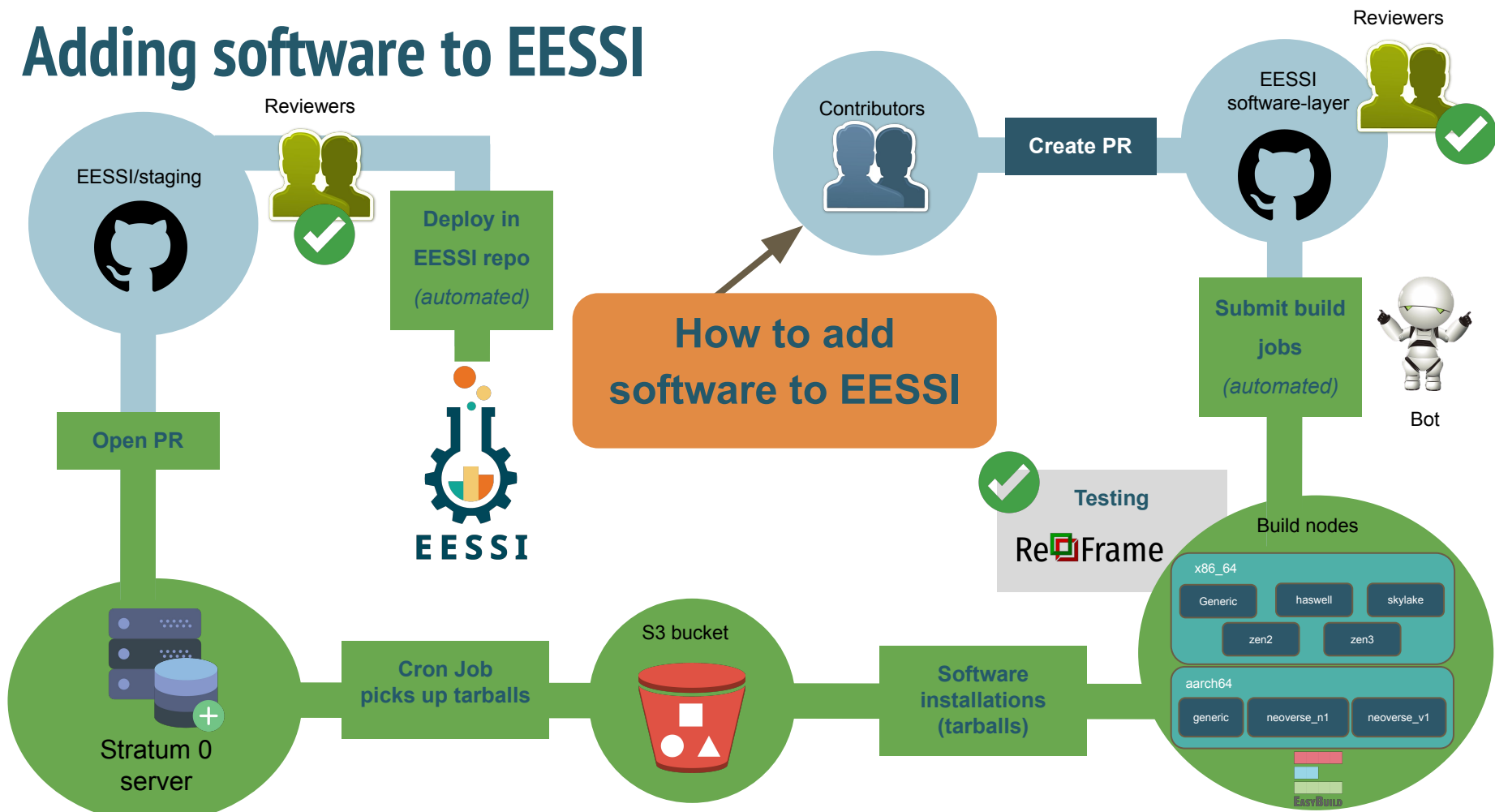
- **Production repository software .eessi.io has been released!**
- Ansible playbooks, scripts, docs at <https://github.com/eessi>
- CernVM-FS: Stratum 0 @ Univ. of Groningen + two Stratum 1 servers
- Software (CPU-only): Bioconductor, GROMACS, OpenFOAM, R, TensorFlow, ...
- Hardware targets:
 - {aarch64,x86_64}/generic
 - intel/{haswell,skylake_avx512}, amd/{zen2,zen3,zen4}, aarch64/{neoverse_n1,neoverse_v1}
- NVIDIA GPU support being brought to production standard
- Supported by Azure and AWS: sponsored credits to develop necessary infrastructure





What about adding the software I care about?

Adding software to EESSI

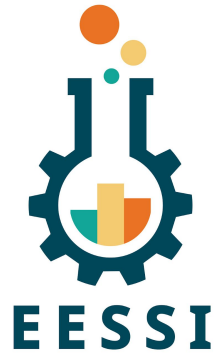


eessi.io/docs/software_layer/adding_software



EESSI use cases

Overview of use cases enabled by EESSI



- A uniform software stack across HPC clusters, clouds, servers, and laptops
- Enable portable workflows
- Can be leveraged in continuous integration (CI) environments
- Significantly facilitates setting up infrastructure for HPC training
- Enhanced collaboration with software developers and application experts

Also discussed in our open-access paper, available via doi.org/10.1002/spe.3075

EESSI provides a uniform software stack



- Main goal: **same software everywhere**: laptop, server, HPC, cloud, ...
- Wide variety of systems supported
 - CPUs: x86_64 (Intel, AMD), aarch64 (Arm), riscv64 (soon...)
 - OS: any Linux distribution, on Windows via WSL, on macOS via Lima
 - High-speed interconnects (Infiniband), GPUs, etc.
- **Without compromising on software performance**
 - Optimized software installations for specific CPU microarchitectures + auto-detection
 - Large contrast with generic binaries often used in containers
- Facilitates migrating from laptop to HPC, cloud bursting, ...

EESSI enables portable workflows



- Portable workflows are significantly easier when relying on EESSI
- They often involve running a broad set of tools, which all need to be available
- Workflows definitions (Snakemake, Nextflow,...) can leverage (or be included in) EESSI
- You can ship your execution environment *inside* your git repository using [direnv](#)
 - If your users have EESSI and `direnv`, then can start running your workflow after cloning!



Hands-on 2: Shipping a workflow (10 mins)

Hands-on scenario



- On provided infrastructure
 - EESSI is available, and `direnv` is shipped via EESSI
- Clone a workflow that leverages EESSI
 - `git clone https://github.com/EESSI/eessi-nextflow-example`
- **After initialising EESSI**, workflow is trivial so can be run via shell
 - Enter repository: `cd eessi-nextflow-example`
 - Setup `direnv`: `eval "$(direnv hook bash)"`
 - Allow `direnv` in dir: `direnv allow`
 - Run the workflow: `./hello_plus_version.nf`

Leveraging EESSI in CI environments



- EESSI can be used in CI environments like:
 - GitHub: github.com/marketplace/actions/eessi
 - GitLab: gitlab.com/explore/catalog/eessi/gitlab-eessi
- We can provide:
 - Different compilers to test your software with
 - Required dependencies for your software
 - Additional tools like ReFrame, performance analysis tools,...
- Other than CernVM-FS, no software installations required
 - Everything that is actually needed is pulled in on-demand by CernVM-FS
- Significantly facilitates also running CI tests in other contexts (laptop, HPC, ...)

Leveraging EESSI in CI environment



We have an EESSI GitHub Action that provides EESSI+di renv:

See it in action in the `github-eessi-action` repository:

github.com/EESSI/github-action-eessi

github.com/EESSI/github-action-eessi/blob/main/.github/workflows/gromacs-usage.yml

```
name: ubuntu_tensorflow
on: [push, pull_request]
jobs:
```

```
  build:
```

```
    runs-on: ubuntu-latest
```

```
    steps:
```

```
      - uses: actions/checkout@v3
```

```
      - uses: eessi/github-action-eessi@v3
```

```
      with:
```

```
        eessi_stack_version: '2023.06'
```

```
      - name: Test EESSI
```

```
        shell: bash
```

```
        run: |
```

```
          module load TensorFlow
```

```
          python -c 'import tensorflow; print(tensorflow.__version__)'
```



Leveraging EESSI GitHub Action



```
build
succeeded 2 minutes ago in 1m 1s
Search logs

> ✓ Set up job 2s
> ✓ Run actions/checkout@v2 0s
> ✓ Run eessi/github-action-eessi@main 52s
▼ ✓ Test EESSI 5s
  1 ▼ Run module load GROMACS
  2   module load GROMACS
  3   gmx --version
  4   shell: /usr/bin/bash --noprofile --norc -e -o pipefail {0}
  5   env:
  6     EESSI_SILENT: 1
  7     BASH_ENV: /cvmfs/pilot.eessi-hpc.org/versions/2021.06/init/bash
  8
  9     :-) GROMACS - gmx, 2020.4-MODIFIED (-:
 10
 11           GROMACS is written by:
 12   Emile Apol      Rossen Apostolov    Paul Bauer    Herman J.C. Berendsen
 13   Par Bjelkmar    Christian Blau    Viacheslav Boilykh    Kevin Boyd
 14   Aldert van Buuren    Rudl van Drunen    Anton Feenstra    Alan Gray
 15   Gerrit Groenhof    Anca Hamuraru    Vincent Hindriksen    M. Eric Irngang
 16   Aleksei Iupinov    Christoph Junghans    Joe Jordan    Dimitrios Karkoulis
 17   Peter Kasson      Jiri Kraus    Carsten Kutzner    Per Larsson
 18   Justin A. Lemkul    Viveca Lindahl    Magnus Lundborg    Erik Marklund
 19   Pascal Merz      Pieter Meulenhoff    Teemu Murtola    Szilard Pall
 20   Sander Pronk      Roland Schulz    Michael Shirts    Alexey Shvetsov
 21   Alfons Sijbers    Peter Tieleman    Jon Vincent    Teemu Virolainen
 22   Christian Wennberg    Maarten Wolf    Artem Zhmurov
 23   and the project leaders:
```

<https://github.com/EESSI/github-action-eessi/actions/runs/3044539257/jobs/4905040409>

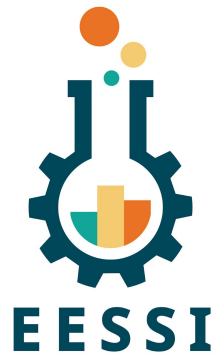
Facilitate HPC training

- EESSI can significantly reduce effort required to set up infrastructure for HPC training sessions (introductory, software-specific, ...)
- Setting up a throwaway Slurm cluster in the cloud is easy via Magic Castle
 - Simple process once you have cloud credits:
 - Create cluster by editing a single file
 - Automatically configured within 20 minutes
 - Includes support for GPU and fast interconnects (infiniband, EFA)
 - EESSI project uses Magic Castle for the build-and-deploy “bot”
 - There are also commercial alternatives that can/will support EESSI (Azure/AWS)
- EESSI can provide (scientific) software that is required for the training
- Attendees can easily set up the *same* software environment later on their own system(s) by leveraging EESSI



Collaboration with software developers + experts

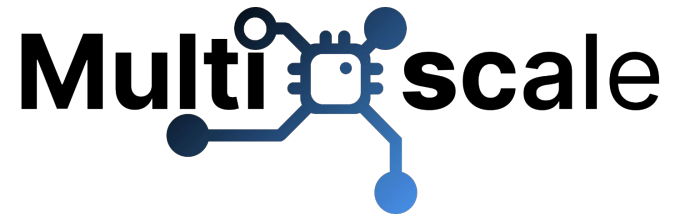
- A central software stack by/for the community opens new doors...
- We can work with software developers/experts to verify the installation
 - Check how installation is configured and built
 - Help to verify whether software is functional for different use cases
 - Show us how to do extensive testing of their software
 - Evaluate performance of the software, enable performance monitoring
 - **“Approved by developers” stamp for major applications included in EESSI**
- Relieve software developers from burden of getting their software installed
 - Remove need to provide pre-built binary packages?
- Developers can also leverage EESSI themselves: dependencies, CI, ...





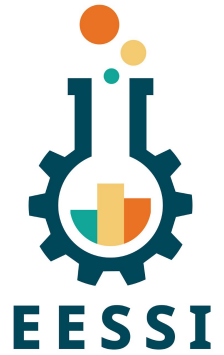
Who pays for this?

The MultiXscale EuroHPC Project



- EuroHPC Centre of Excellence
 - 4 year project, started Q1 2023
- Budget of ~6M EUR (50% EU funding, 50% national funding)
 - Roughly 50% of funding for EESSI-related activities
- Collaboration between EESSI and CECAM (total of 16 partners)
 - EESSI primarily addresses technical aspects
 - CECAM network provides scientific expertise
- Scientific target are multiscale simulations with 3 key use cases
 - Helicopter design and certification for civil transport
 - Battery applications to support the sustainable energy transition
 - Ultrasound for non-invasive diagnostics and biomedical applications
- <https://www.multixscale.eu>

EESSI is nominated for an award!



- EESSI has been nominated for the **HPCwire Readers' Choice Awards 2024**, in the *"Best HPC Programming Tool or Technology"* category.
- **You can help us win the award by joining the vote**
- <https://www.hpcwire.com/2024-hpcwire-readers-choice-awards-voting-is-open>
- <https://www.eessi.io/docs/blog/2024/09/20/hpcwire-readers-choice-awards-2024>



E E S S I

EUROPEAN ENVIRONMENT FOR
SCIENTIFIC SOFTWARE INSTALLATIONS

Website: eessi.io

GitHub: github.com/eessi

Documentation: eessi.io/docs

Blog: eessi.io/docs/blog

YouTube channel: youtube.com/@eessi_community

Paper (open access): doi.org/10.1002/spe.3075

EESSI support portal: gitlab.com/eessi/support

[Monthly online meetings](#) (first Thursday, 2pm CEST)

Join our mailing list & Slack channel