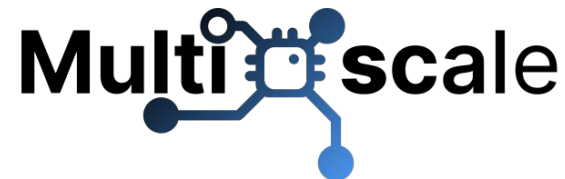
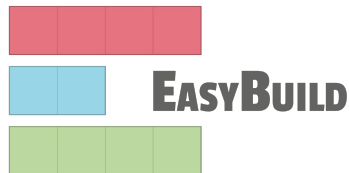


Prepared environment for hands-on

- Small Rocky 9 cluster (in the cloud)
- **You need to create an account!**
 - Signup: <https://mokey.tutorial.eessi.science/auth/signup>
 - Accounts will only be approved for access on the day of the tutorial, so **please record your username/password !**
 - “Reset password” link does **not** work
- Access via ssh or web browser (**pick one and stick to it**)
 - Shell access: `ssh tutorial.eessi.science`
 - Use login node for hands-on, it has 16 cores so should be fine to share for small group
 - Via browser: <https://tutorial.eessi.science/>
 - Make sure to change default “Time” to 4 hours
 - Take 4 cores
- System will be up for the week



hackmd.io/@multixscale/EESSI



EESSI CI/CD

20 May 2026 - Alan O'Cais

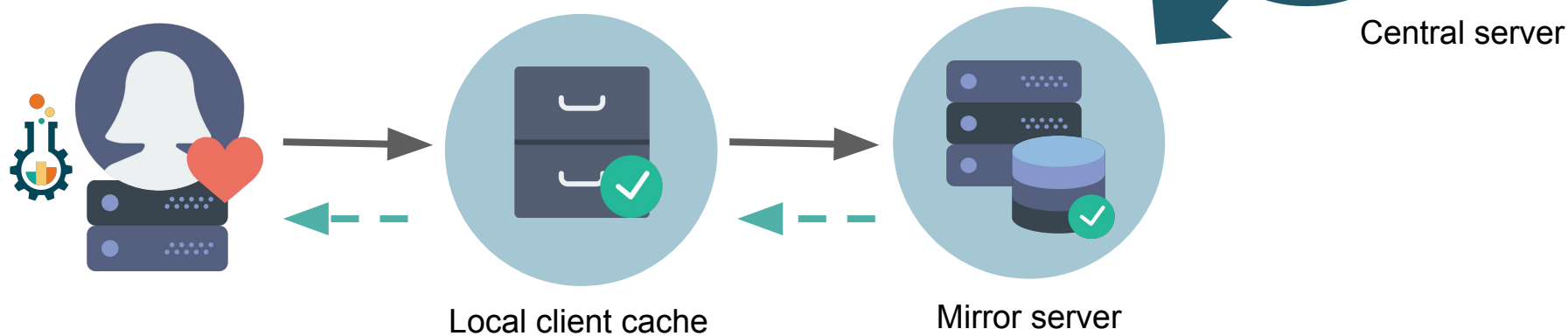
<https://www.eessi.io/docs/training-events/2026/webinar-series-2026Q2>

*What if you no longer have to install
a **broad range of scientific software**
from scratch on every laptop, HPC cluster,
or cloud instance you use or maintain,
without compromising on performance?*



The EESSI User Experience

```
$ source /cvmfs/software.eessi.io/versions/2023.06/init/lmod/bash
Module for EESSI/2023.06 loaded successfully
$ module load GROMACS/2024.1-foss-2023b
$ gmx mdrun ...
```



EESSI provides **on-demand streaming**
of (scientific) software (like music, TV-series, ...)

EESSI for CI/CD



- Introducing Continuous Integration (CI)
- Navigating EESSI to build my project
- Building my project with the EESSI GitHub Action
- Building with **EESSI-extend** and the EESSI GitHub Action
- Continuous Deployment and what EESSI can offer there

What is Continuous Integration (CI)?

- Development practice of frequently merging code into a shared repository
- Automated build and test run on each code change
- Helps detect bugs early and improve code quality
- Provides immediate feedback to developers
- Ensures the application is always in a deployable state

Key Components of CI

-  Version Control System (e.g., Git)
-  Automated Build System
-  Automated Testing Suite
-  Notification/Feedback Mechanism

Benefits of Continuous Integration

-  Early detection of integration issues
-  Faster and safer release cycles
-  Encourages small, incremental changes
-  Improved code quality and team collaboration
-  Easier refactoring and code maintenance

Specific Challenges of CI in HPC context

- Need HPC-suitable toolchains and dependencies
- Need a way to deal with MPI
- Typically want to test defined architectures
- Want to have accelerator support
- Performance and scalability (also) matter

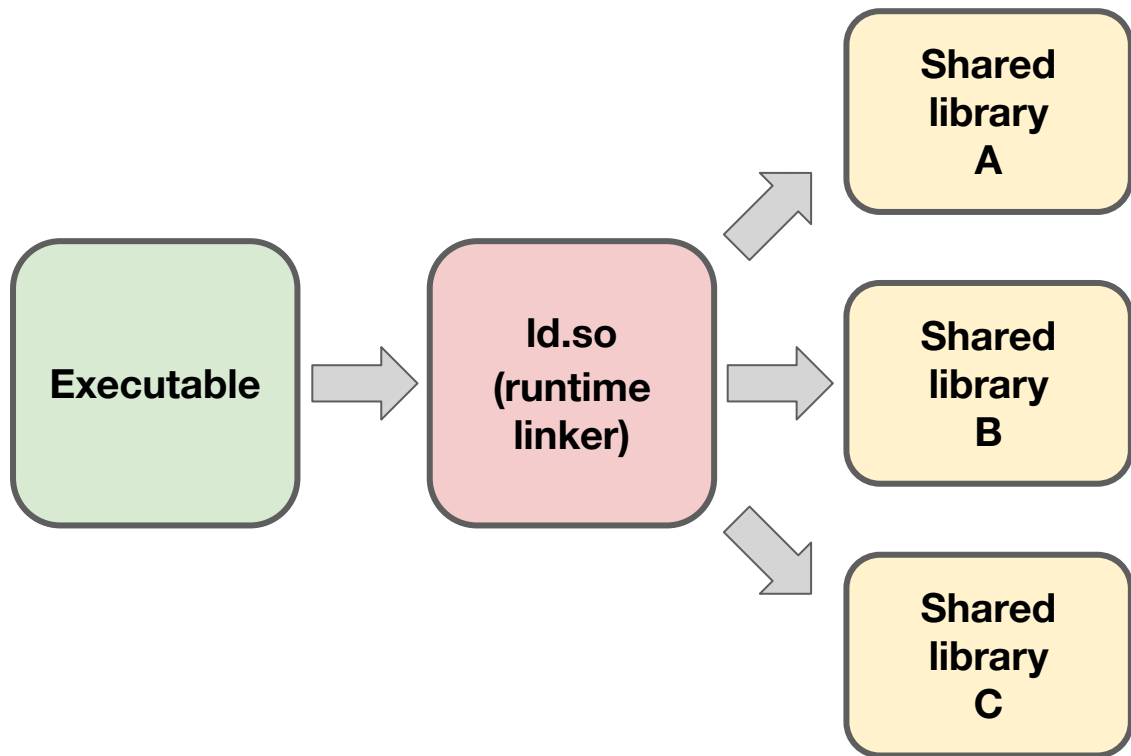
Leveraging EESSI in CI environments

- EESSI can be used in CI environments like:
 - GitHub: github.com/marketplace/actions/eessi
 - GitLab: gitlab.com/explore/catalog/eessi/gitlab-eessi
- EESSI 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 to get access to EESSI, no installations required!
 - Everything is pulled in on-demand by CernVM-FS
 - ...and someone else maintains the environment
- Significantly facilitates also running CI tests in other contexts

Building on top of EESSI

- Load the necessary modules that you need for compilation
 - Compiler, MPI, libraries, dependencies,...
 - Best to try to keep dependencies in the same toolchain
- Load a **builddenv** module for your toolchain
 - Configures typical environment variables for a build (**CC**, **CFLAGS**, **LIBS**, ...)
 - Uses *compiler wrappers* to simplify building with EESSI

Why do we need **builddenv**?



Where does the loader find the libraries?

- Clues in environment variables
- Information in the binary (RPATH)
- It's own defaults

An example software package

- github.com/EESSI/cicd-demo
- Dependencies: HDF5 and MPI
- Builds with CMake
- Tests via **ctest**

Cheat sheet: <https://hackmd.io/myPkGyj-Rz6pQNMOZW1IA?view>

EESSI plus Spack can also work in CI

- EESSI integration with Spack is still (very much) a work in progress
- github.com/lorisercole/spood is where things are currently happening
 - Demonstrates that the integration can work
- Proof of principle of EESSI + Spack multi-arch CI on GitHub
 - github.com/ocaisa/spood/pull/1

Cheat sheet: <https://hackmd.io/myPkGyj-Rz6pQNMOWi1IA?view>

Continuous Delivery (CD)

- Consider case of “release candidates” and production releases
- Developers commit code → triggers CI
- If tests pass, code can be included in a Release Candidate (RC)
- RC is deployed to a staging environment
- Manual/automated approval takes place
- RC is promoted to production manually or with a controlled deployment

What can EESSI do for CD?

- Production quality software deployed to **software.eessi.io**
- Development service **dev.eessi.io**, mostly for release candidates
 - More control for developers
 - Access/deployment to specific architectures
- EESSI is working on getting EuroHPC site support for **EESSI test suite**, could be leveraged for **dev.eessi.io** to remove need for direct access to resources for development

Repeat demo for yourself

- Good exercise to get the concepts straight
 - Fork github.com/EESSI/cicd-demo
 - Build locally with EESSI first
 - Transfer build to GitHub (or GitLab)
 - Port to EasyBuild (this is an investment for EESSI CD)
 - Transfer port to GitHub (or GitLab)
- Cheat sheet: <https://hackmd.io/myPkGyj-Rz6pQNMOZW1IA?view>



Website: <https://eessi.io>

Join our Slack channel (see join link on website)

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

Blog: <https://eessi.io/docs/blog>

GitHub: <https://github.com/eessi>

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

[EESSI YouTube channel](#)

[Bi-monthly online meetings](#)

(first Thursday of every other month, 14:00 CEST)

[EESSI Happy Hour](#)

(every Monday, 14:00 CEST)

MultiXscale

Web page: multixscale.eu

Facebook: [MultiXscale](https://www.facebook.com/MultiXscale)

Twitter: [@MultiXscale](https://twitter.com/MultiXscale)

LinkedIn: [MultiXscale](https://www.linkedin.com/company/multixscale)

BlueSky: [MultiXscale](https://bsky.app/profile/multixscale)



Co-funded by
the European Union



EuroHPC
Joint Undertaking



UNIVERSITAT DE
BARCELONA



Universität
Stuttgart



SORBONNE
UNIVERSITÉ



Université
de Toulouse



Consiglio Nazionale
delle Ricerche



MAX-PLANCK-GESELLSCHAFT

