Co-funded by the European Union

EuroHPC Joint Undertaking

EuroHPC Centre of Excellence

EESSI status update @ EUM23

Caspar van Leeuwen

SURF

25-04-2023

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European High Performance Computing Joint Undertaking (JU) and countries participating in the project. Neither the European Union nor the granting authority can be held responsible for them.

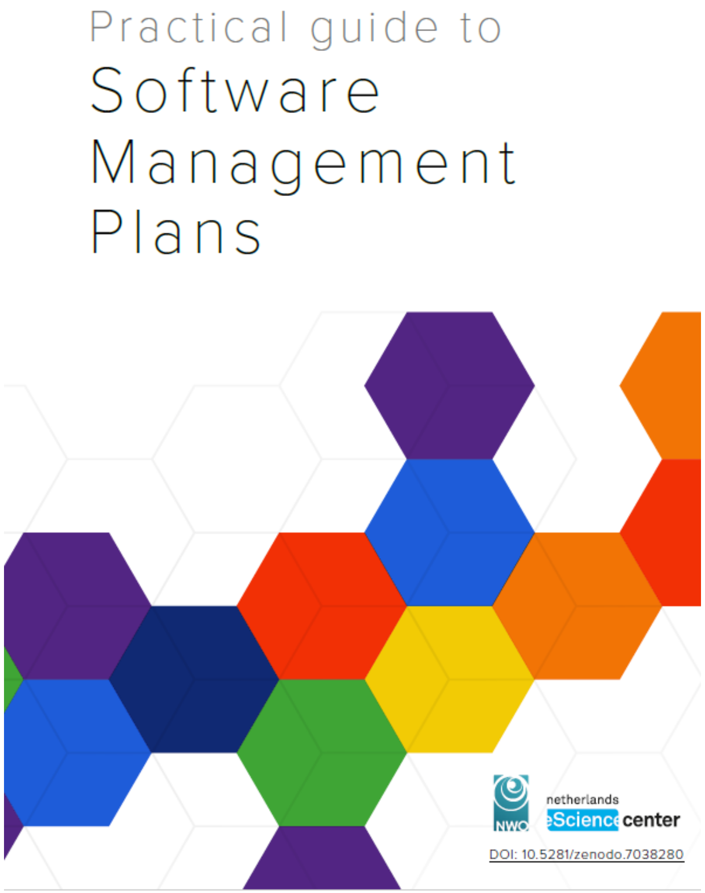
About me

- Caspar van Leeuwen
- High Performance Computing, Machine Learning
- Joined SURF 6 years ago
- EasyBuild Maintainer
- Contributions: EasyBuild RPATH support, easystack files, easyconfigs

Before I start…

- Let’s educate the scientific software developer!
- Martinez-Ortiz, Carlos et al 2023
- Disclaimer: I was on the Sounding Board for the development of this guide ☺

DOI: 10.5281/zenodo.7589725

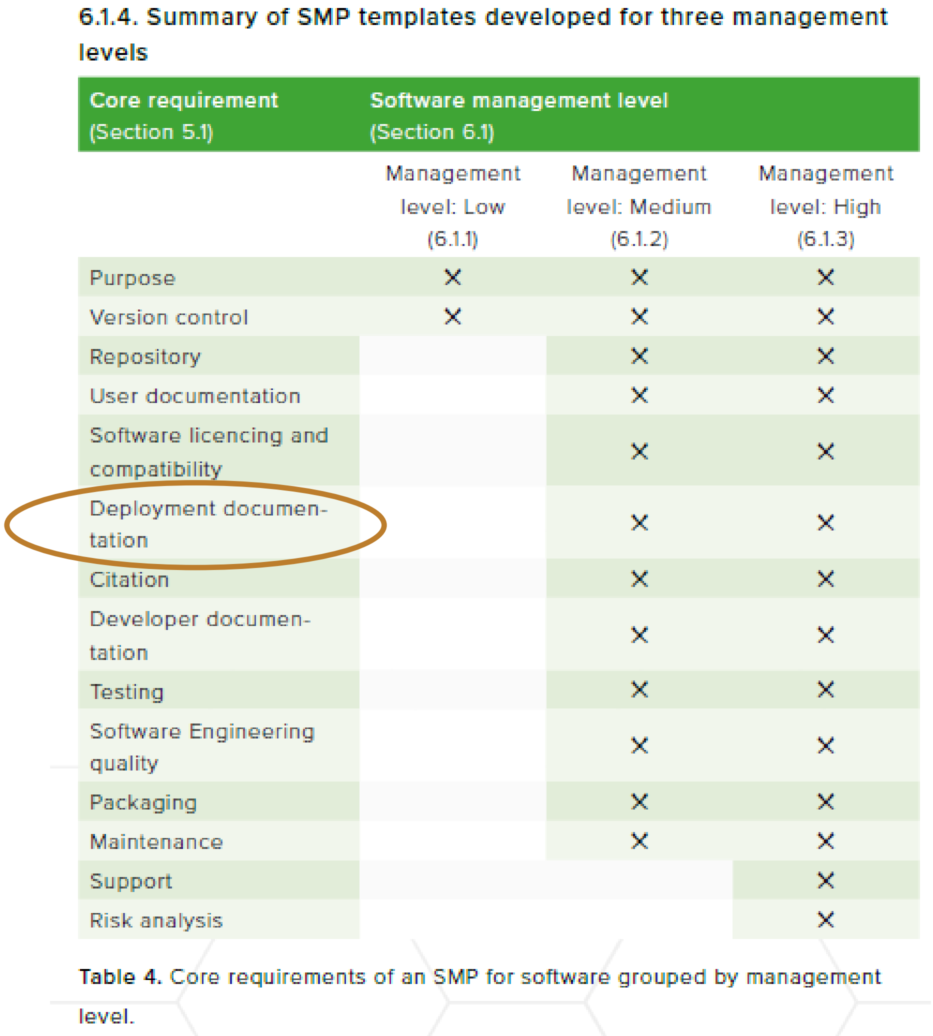


Practical guide to Software Management Plans

NWO

netherlands eScience center

DOI: 10.5281/zenodo.7038280



6.1.4. Summary of SMP templates developed for three management levels

Core requirement (Section 5.1)	Software management level (Section 6.1)		
	Management level: Low (6.1.1)	Management level: Medium (6.1.2)	Management level: High (6.1.3)
Purpose	×	×	×
Version control	×	×	×
Repository		×	×
User documentation		×	×
Software licencing and compatibility		×	×
Deployment documentation		×	×
Citation		×	×
Developer documentation		×	×
Testing		×	×
Software Engineering quality		×	×
Packaging		×	×
Maintenance		×	×
Support			×
Risk analysis			×

Table 4. Core requirements of an SMP for software grouped by management level.

My dream

Image: https://www.abbythepup.com/

I want scientists to be able to run their computation …

- on any compute infrastructure they want,
- with whatever software they need,
- on any data they want,
- ...
- *making the most efficient usage of that compute infrastructure*

↑

That’s my high-performance computing heart talking ☺

The EESSI dream

A …

- Cross-platform (laptop, cloud VM, HPC cluster)
- Ready-to-use (just mount-and-go)
- Optimized (CPU architecture, GPU architecture, interconnects)
- Software stack

Image: freesvg.org

Shared dreams…

I want scientists to be able to run their computation …

- on any compute infrastructure they want,
- with whatever software they need, ✓ EESSI
- on any data they want,
- …
- *making the most efficient usage of that compute infrastructure* ✓ EESSI

EESSI community

Founding partners:

TU Delft

VU University Amsterdam

university of groningen

TU/e Eindhoven University of Technology

SURF

UNIVERSITY OF TWENTE.

UNIVERSITY OF CAMBRIDGE

GHENT UNIVERSITY

DELL Technologies

Extensive interest from HPC *and* cloud community:

HPC Now!

Microsoft Azure

CSCS

CERN

aws

cecam Centre Européen de Calcul Atomique et Moléculaire

FRED HUTCH

JÜLICH Forschungszentrum

UiO : University of Oslo

HPC.NRW

The inception of EESSI

High performance computing (HPC) centers manage large software stacks for their users

- Focus on performance (big calculations, performance loss = more money spent)
- Increasingly complex world
  - more (research) software
  - more non-traditional (inexperienced) HPC users
  - more flavours of hardware
- Too much work for HPC staff …

Image: pexels.com

Avoid duplicate work

Current situation

- Use build tools (e.g. EasyBuild, Spack)
  - Build-from-source procedures shared through ‘recipes’
  - Each site still installs their own stack (and tests?)
  - Build procedures do not always work ‘out of the box’ … (different OS-es, etc)

EESSI

- All contribute to *one* shared software stack

Image: freeimageslive.co.uk

Benefit to the end-user

Current situation

- Moving from one system to another (e.g. laptop, cloud, HPC cluster) takes effort!
    - Moving data
    - Recreating software environment

Using EESSI

- Software environment is identical
- Only move data

Image: rawpixel.com

EESSI: Scope & goals

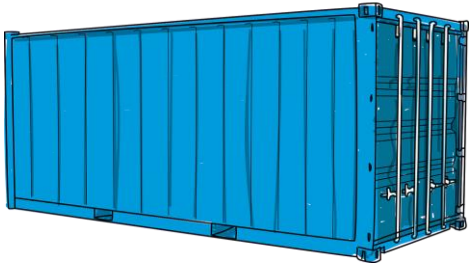
- *European Environment for Scientific Software Installations (EESSI)*
- **Shared repository of (optimized!) scientific software installations**
- Avoid duplicate work across IT support teams: collaborate on a shared software stack
- Uniform way of providing software to users, regardless of system they use!
- Should work on any Linux OS (+ WSL, and possibly 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**

EESSI
EUROPEAN ENVIRONMENT FOR
SCIENTIFIC SOFTWARE INSTALLATIONS

https://www.eessi-hpc.org

https://eessi.github.io/docs (try out the pilot setup!)

Why not just containers?



Containers are …

- designed for portability => typically built without hardware-specific optimizations
- often quite large/bulky
    - Download several GB just to use one small tool
- a static environment
    - Additional tool needed? => Rebuild container, or pull in another one
- lot’s of duplication => hard to test (N containers means testing N full software stacks)
- …

Image: pixabay.com

Learn from the things that work

- Containers are isolated from the host because they have their own OS
- The Alliance has a shared software stack between the systems they manage

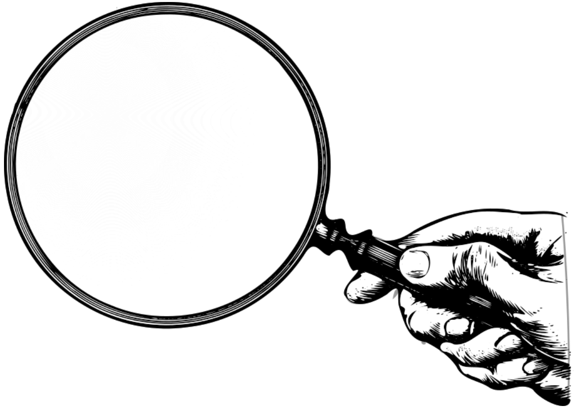


Image: openclipart.org

So, how does EESSI work…?

gentoo

Abstraction from the OS (like a container)

So, how does EESSI work…?

gentoo

Abstraction from the OS (like a container)

easybuild

Optimized builds for a large range of hardware architectures

So, how does EESSI work…?

CernVM-FS

A way to get the software distributed globally

gentoo

Abstraction from the OS (like a container)

easybuild

Optimized builds for a large range of hardware architectures

So, how does EESSI work…?

CernVM-FS

A way to get the software distributed globally

gentoo

Abstraction from the OS (like a container)

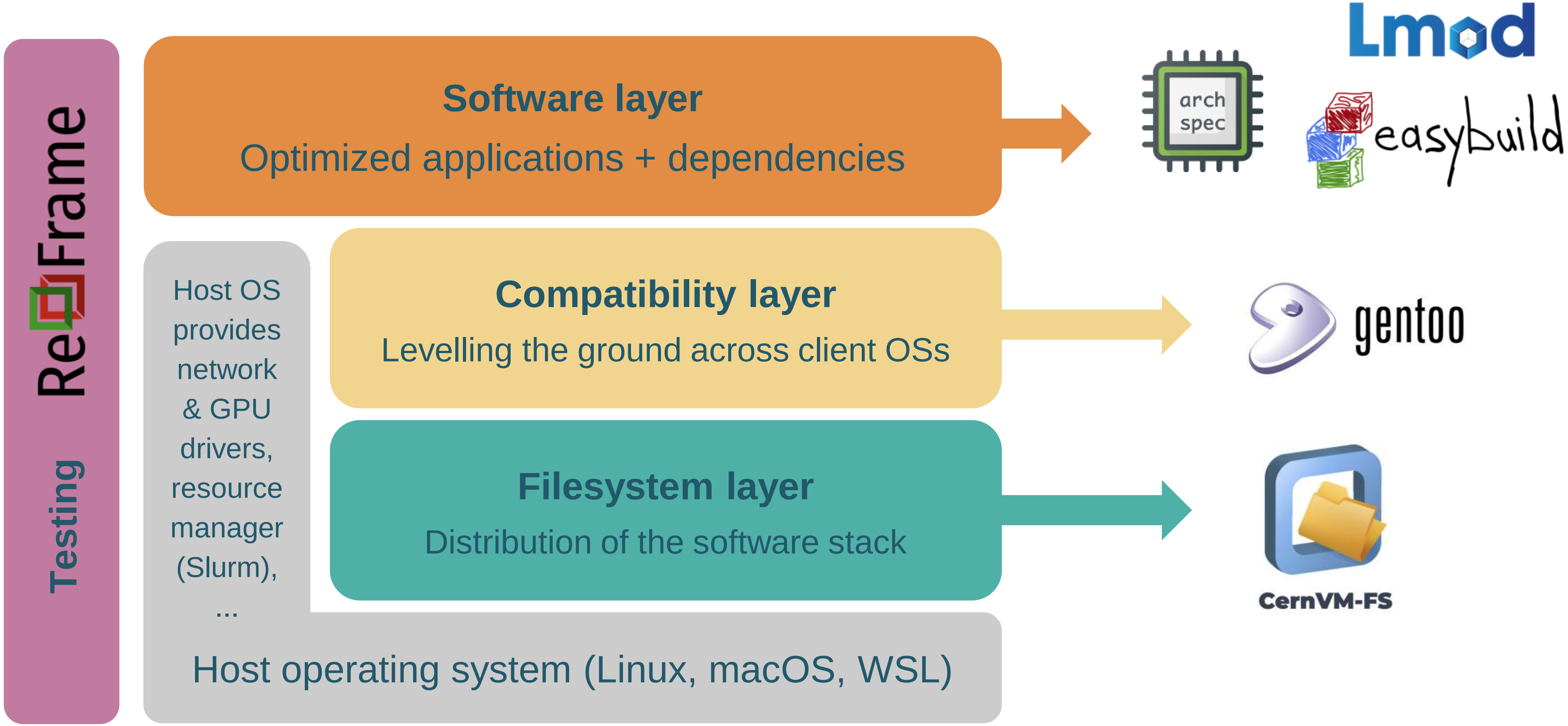
easybuild

Optimized builds for a large range of hardware architectures

arch spec

Automatic selection of the right optimization at runtime

High-level overview of EESSI project



Software layer

Optimized applications + dependencies

Compatibility layer

Levelling the ground across client OSs

Filesystem layer

Distribution of the software stack

Host OS provides network & GPU drivers, resource manager (Slurm), ...

Host operating system (Linux, macOS, WSL)

Testing

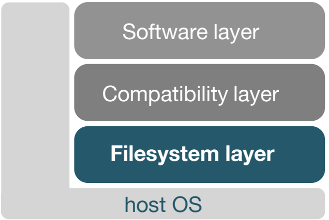
intel AMD ARM POWER9 RISC-V

CernVM-FS

https://cvmfs.readthedocs.io

CERN virtual machine filesystem

- Developed to support software deployment on the worldwide-distributing computing infrastructure used by CERN (the ‘Grid’)
- POSIX read-only file system in user space
- Files and directories are hosted on standard webservers and mounted in /cvmfs
- Strong focus on redundancy and I/O performance (mirrors & caching)
- Pulls in files “as needed” (more efficient compared to containers)



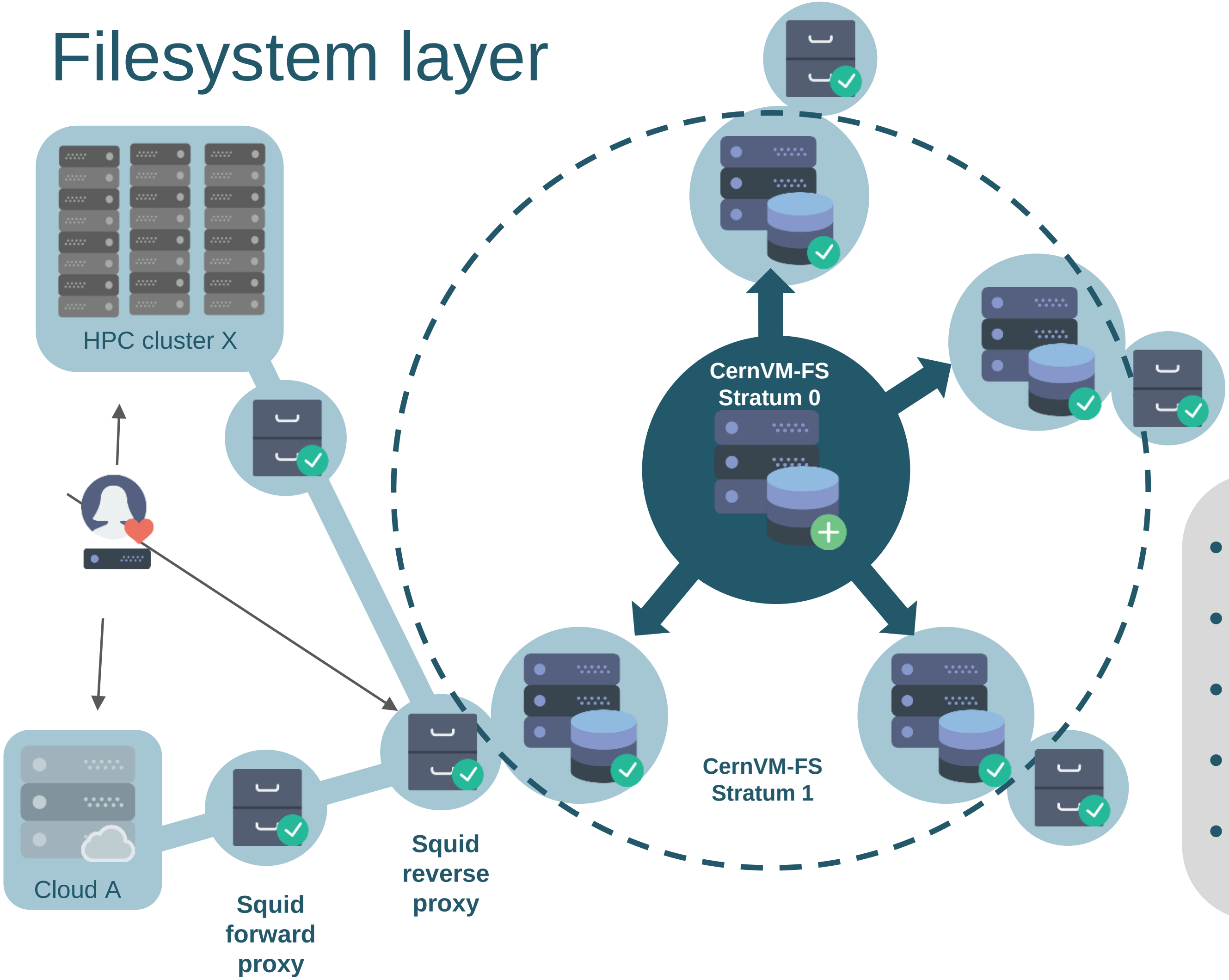
Software layer

Compatibility layer

Filesystem layer

host OS

Filesystem layer



CernVM-FS

https://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, …

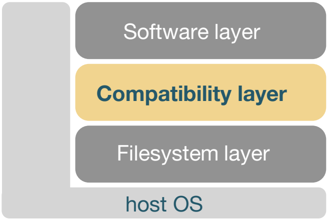
(icons via https://www.flaticon.com/authors/smashicons)

Compatibility layer

- **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`, `ppc64le`, `x86_64`)
- Creates ‘level playing field’ to build software layer, so that it works on large range of host OS-es
- Currently in pilot repository:

```
/cvmfs/pilot.eessi-hpc.org/versions/2021.12/compat/linux/aarch64
/cvmfs/pilot.eessi-hpc.org/versions/2021.12/compat/linux/ppc64le
/cvmfs/pilot.eessi-hpc.org/versions/2021.12/compat/linux/x86_64
```

powered by



Software layer

- Provides scientific software applications, libraries, and dependencies
- **Optimized for specific CPU microarchitectures** (Intel Skylake, 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
- **Best subdirectory for host is selected automatically** via archspec
  - Little end-user knowledge needed
  - Useful when you don't *know* which hardware your task will land on
- Lmod environment modules tool is used to access installations

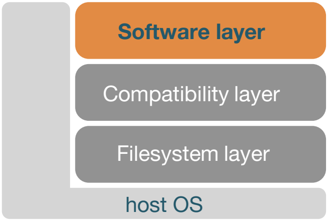
EESSI

powered by

easybuild

Lmod

arch spec



Current status: pilot repository 2021.12

- Working **proof of concept**
- Ansible playbooks, scripts, docs at `https://github.com/eessi`
- CernVM-FS: Stratum 0 @ Univ. of Groningen + four Stratum 1 servers
- Software (CPU-only): Bioconductor, GROMACS, OpenFOAM, R, TensorFlow, Spark, IPython, Horovod, QuantumESPRESSO, ReFrame, …
- Hardware targets:
  - `{aarch64,ppc64le,x86_64}/generic`
  - `intel/{haswell,skylake_avx512}, amd/{zen2,zen3}, aarch64/{graviton2,graviton3), ppc64le/power9le`

https://eessi.github.io/docs/pilot

What is MultiXscale?

MultiXscale:

- Horizon EuroHPC Center of Excellence, focused on multiscale modelling
- 6M euro budget, across 13 sites, 2023 - 2027
- Collaboration between the CECAM[1] network and several partners in EESSI
- Three scientific WPs: develop scientific code for multiscale modelling
- Two technical WPs: develop and support EESSI (facilitating the scientific work packages)

[1] Center for Atomic and Molecular simulations

What is MultiXscale?

Technical Work Packages

- WP1: Developing a Central Platform for Scientific Software on Emerging Exascale Technologies
  - Stability, testing, support for new architectures
- WP5: Building, Supporting and Maintaining a Central Shared Stack of Optimized Scientific Software Installations
  - Support, monitoring, community contributions

Image: vectorportal.com

What is MultiXscale?

Key benefits to EESSI

- MultiXscale has dedicated funding to work on EESSI
- The project plan for MultiXscale essentially gives EESSI a roadmap
- Scientific workpackages provide feedback
- Will stimulate making EESSI available on more clusters
- Will provide training to admins & end users

EESSI: current activities

Improve security of CVMFS stratum 0 with yubikeys

- Acquisition of new (physical) stratum 0 server
- Prerequisite for EESSI config being shipped with CVMFS by default
- Will increase availability of EESSI to *any* system that has CVMFS installed

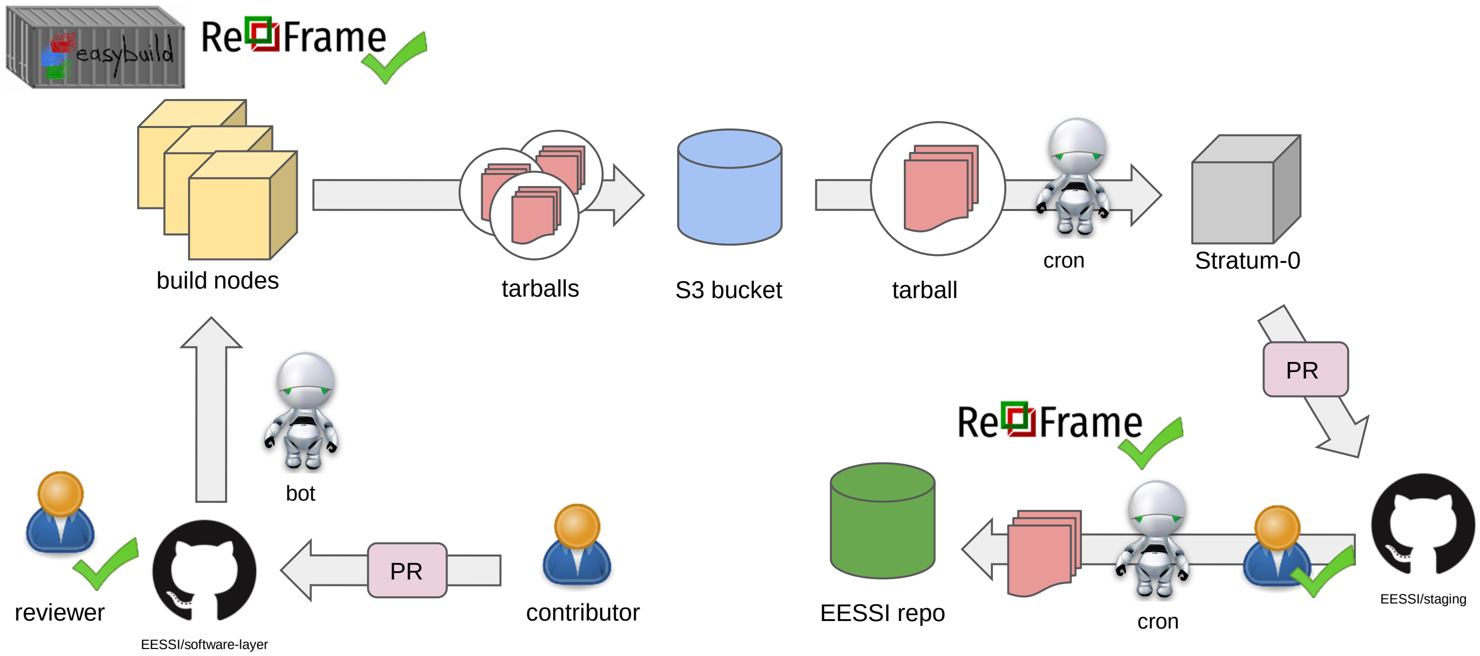
EESSI: current activities

Build new compatibility layer (2023.04)

- Various issues with building (bootstrapping) Gentoo Prefix
- X86_64 & aarch64 now work
  - RISCV64 broken, but less priority
  - PPC64le will only be included if it builds out of the box
- Synergy between The Alliance & EESSI solving these issues

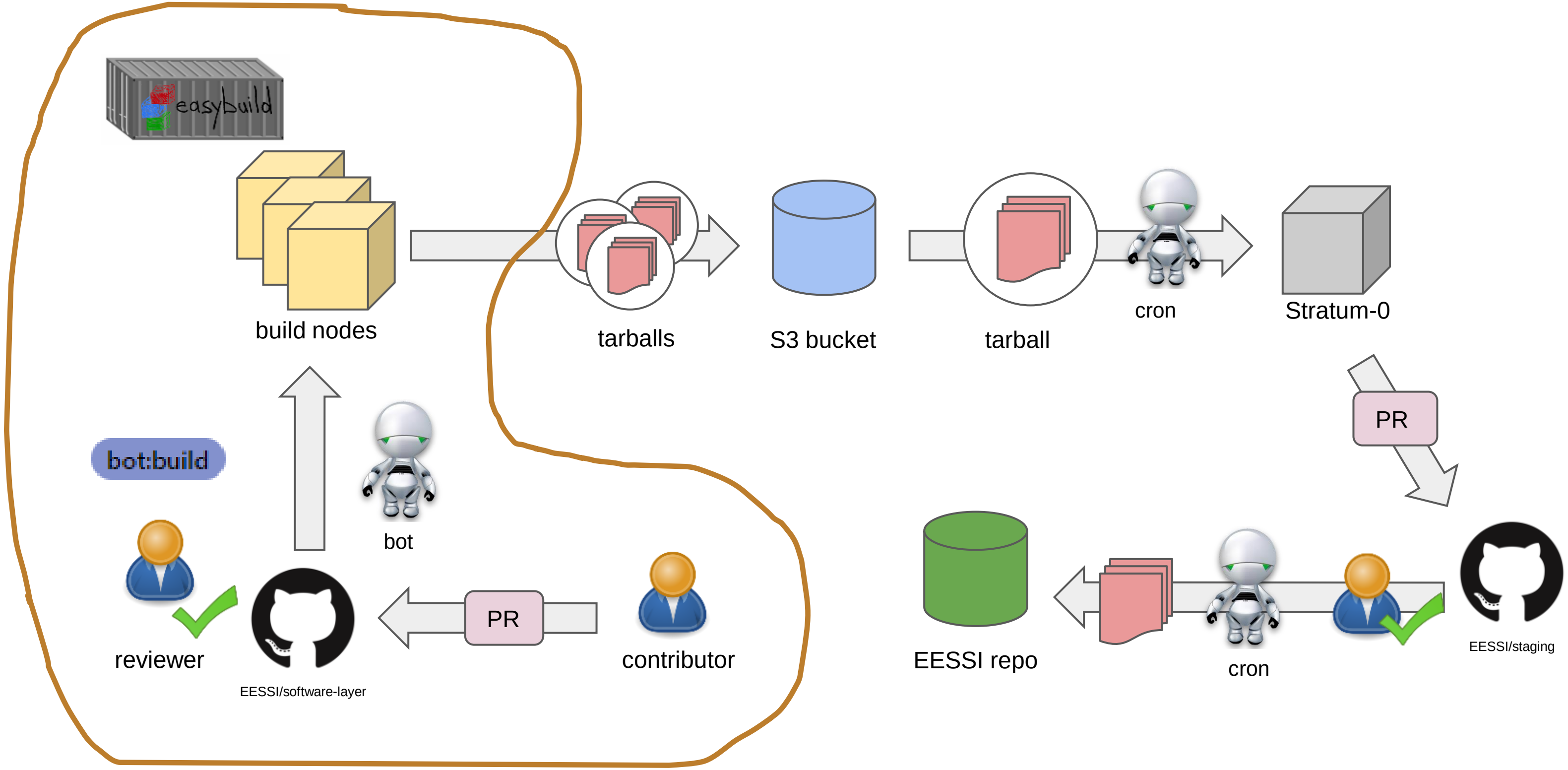
EESSI: current activities

Processing community contributions: automation with human supervision



EESSI: current activities

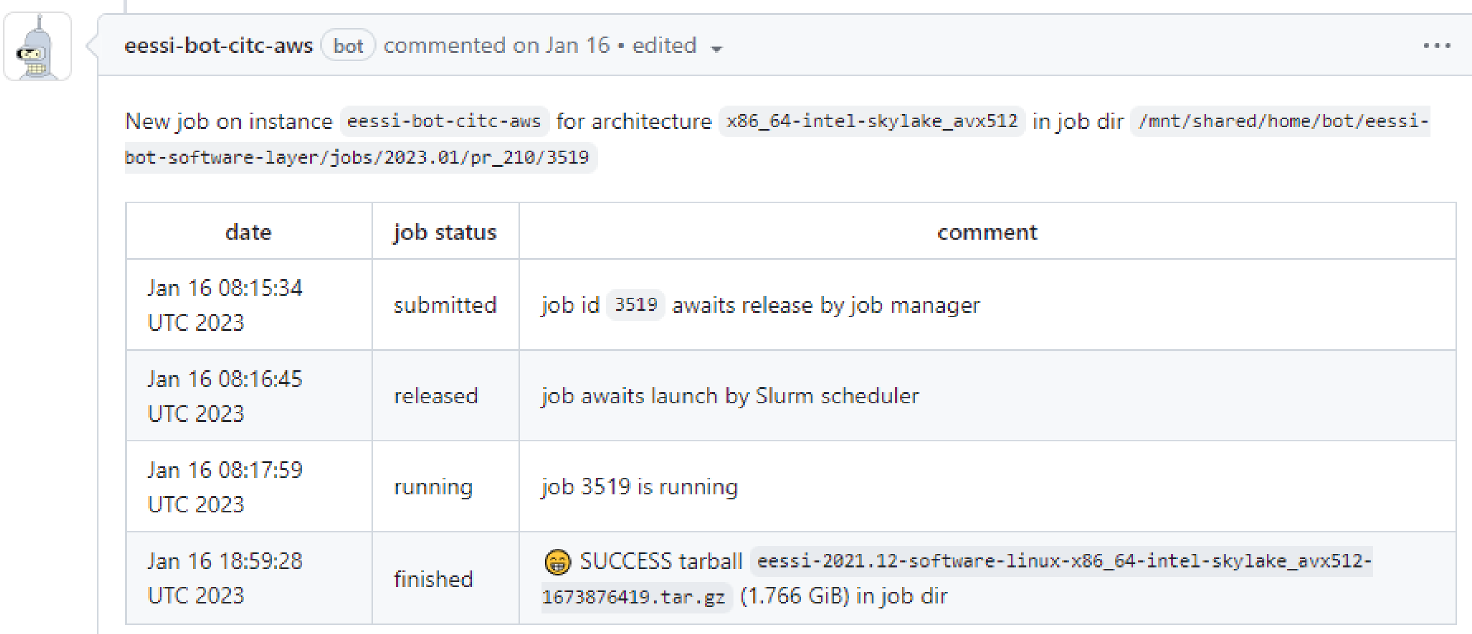
PR to EESSI/SoftwareLayer



EESSI: current activities

Processing community contributions

- Bot automatically builds (with EasyBuild) for EESSI/software-layer PR labeled bot:build (by a reviewer)



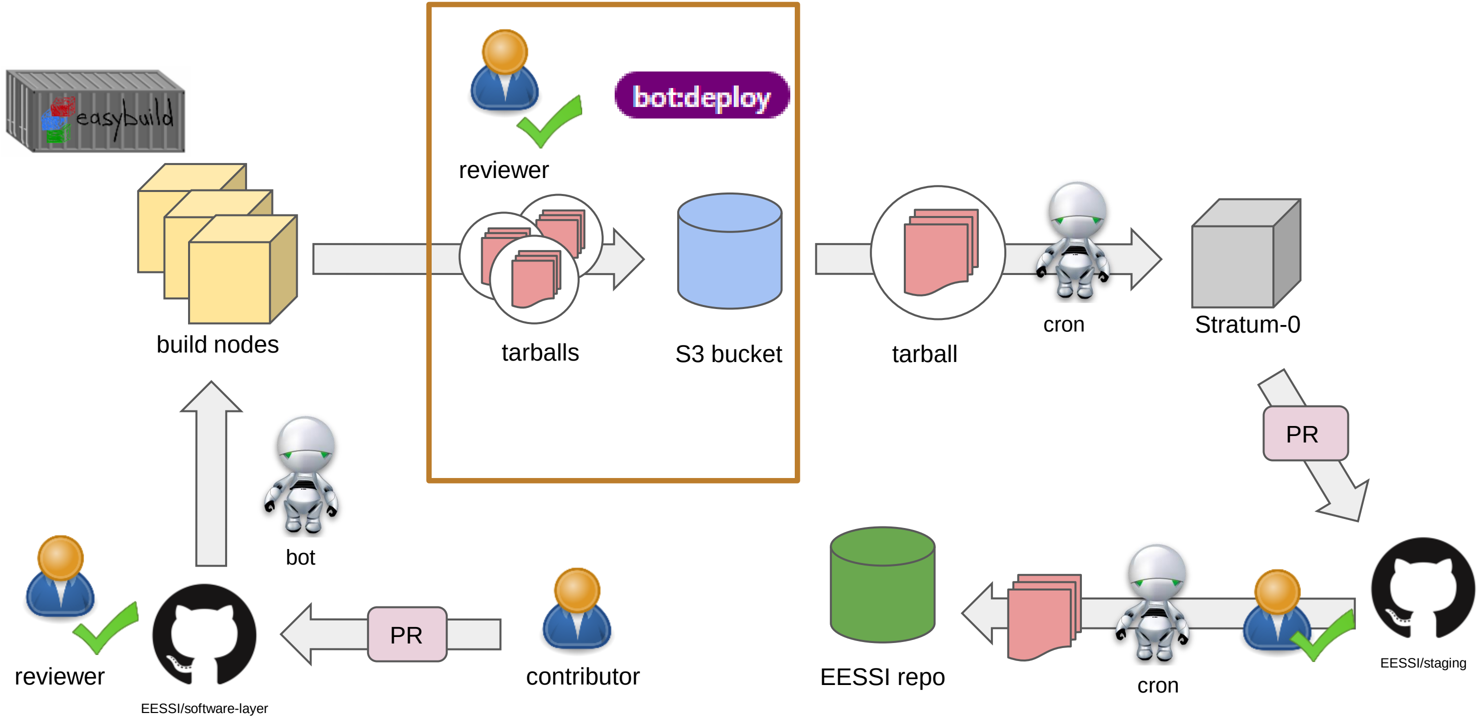
eessi-bot-citc-aws bot commented on Jan 16 • edited

New job on instance `eessi-bot-citc-aws` for architecture `x86_64-intel-skylake_avx512` in job dir `/mnt/shared/home/bot/eessi-bot-software-layer/jobs/2023.01/pr_210/3519`

date	job status	comment
Jan 16 08:15:34 UTC 2023	submitted	job id `3519` awaits release by job manager
Jan 16 08:16:45 UTC 2023	released	job awaits launch by Slurm scheduler
Jan 16 08:17:59 UTC 2023	running	job 3519 is running
Jan 16 18:59:28 UTC 2023	finished	😁 SUCCESS tarball `eessi-2021.12-software-linux-x86_64-intel-skylake_avx512-1673876419.tar.gz` (1.766 GiB) in job dir

EESSI: current activities

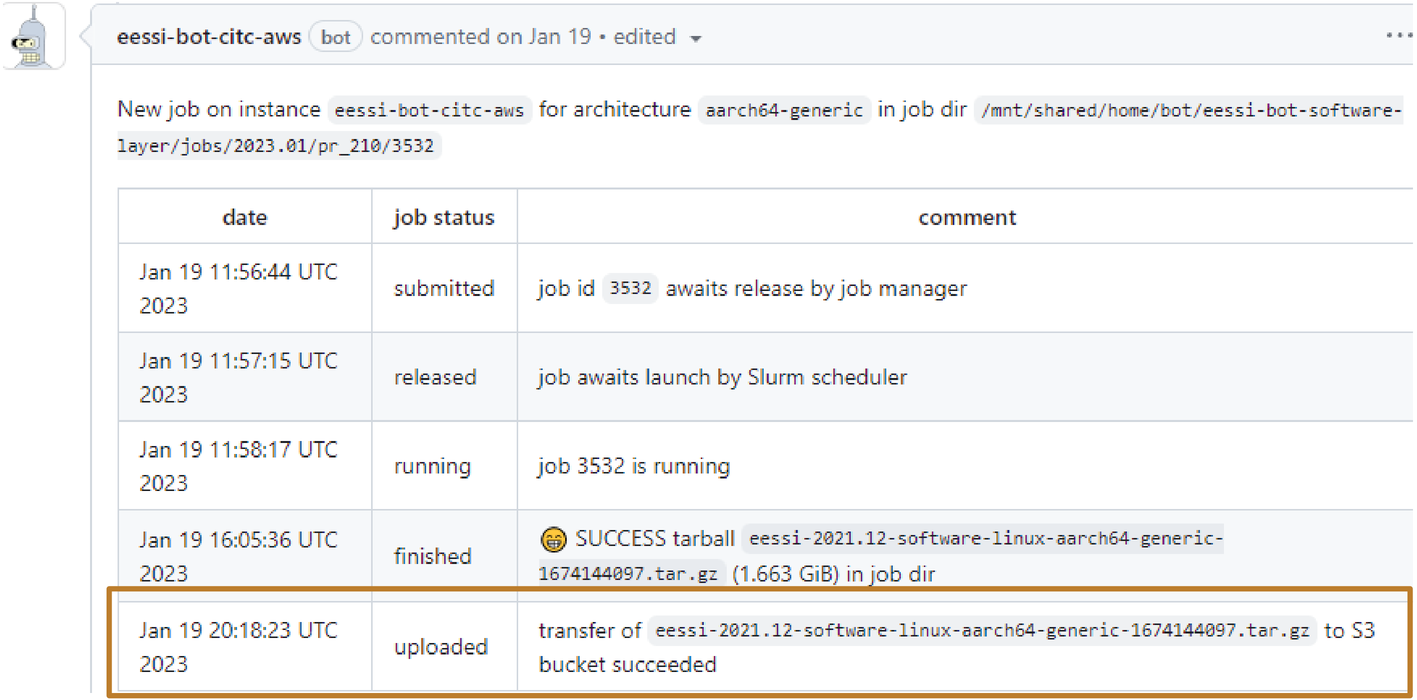
Processing community contributions: automation with human supervision



EESSI: current activities

Processing community contributions

- Bot uploads tarball to S3 bucket when is PR labeled bot:deploy



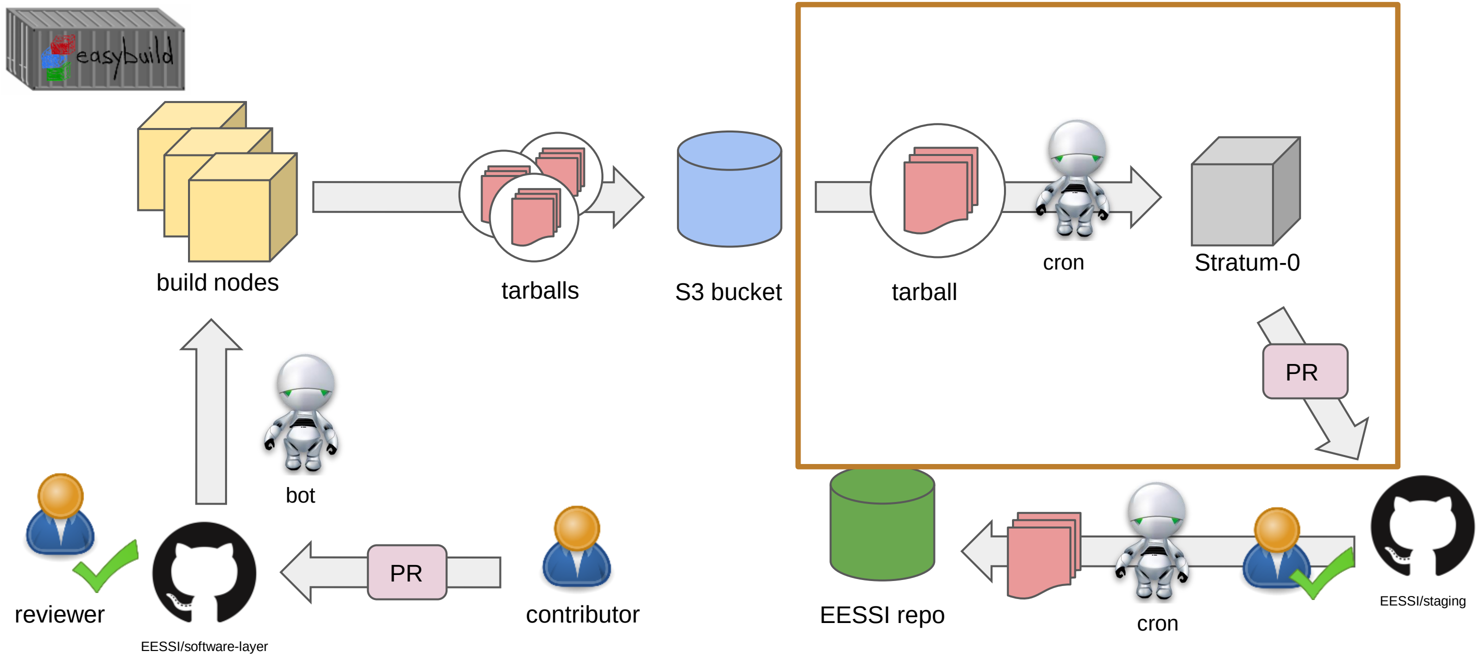
eessi-bot-citc-aws bot commented on Jan 19 • edited

New job on instance `eessi-bot-citc-aws` for architecture `aarch64-generic` in job dir `/mnt/shared/home/bot/eessi-bot-software-layer/jobs/2023.01/pr_210/3532`

date	job status	comment
Jan 19 11:56:44 UTC 2023	submitted	job id `3532` awaits release by job manager
Jan 19 11:57:15 UTC 2023	released	job awaits launch by Slurm scheduler
Jan 19 11:58:17 UTC 2023	running	job 3532 is running
Jan 19 16:05:36 UTC 2023	finished	😁 SUCCESS tarball `eessi-2021.12-software-linux-aarch64-generic-1674144097.tar.gz` (1.663 GiB) in job dir
Jan 19 20:18:23 UTC 2023	uploaded	transfer of `eessi-2021.12-software-linux-aarch64-generic-1674144097.tar.gz` to S3 bucket succeeded

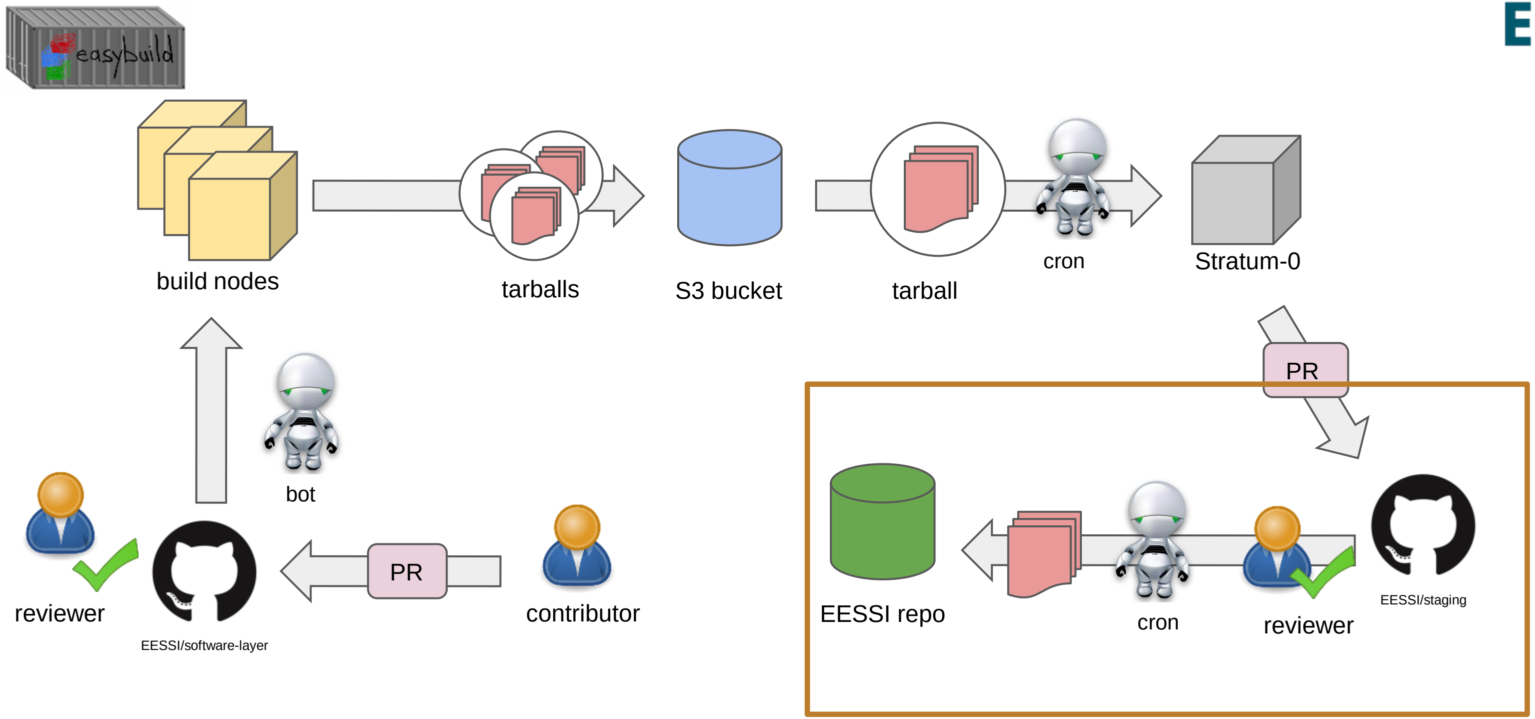
EESSI: current activities

CRON job downloads tarball to Stratum-0 and creates PR to EESSI/staging



EESSI: current activities

Bot triggers CVMFS ingest command when PR is merged



EESSI: current activities

Nvidia GPU support

- Challenge 1: we are not allowed to redistribute all CUDA SDK components (CUDA UELA)
- In build pipeline, EESSI script replaces non-redistributable CUDA code with symlinks
- Symlinks point to a *host_injections* dir (on FS of the host)
- Sysadmins can easily install a full CUDA SDK in the *host_injections* dir, which can then be used by other software from the EESSI stack

EESSI: current activities

Nvidia GPU support

- Challenge 2: new CUDA libraries don’t work with old driver versions, but on the EESSI side, we don’t have control over the driver version
- EESSI provides a script to install CUDA compatibility libraries in the *host_injections* dir. This increases the compatibility range.

EESSI: current activities

Nvidia GPU support

- Challenge 3: EESSI uses a build container to build additional software. However, Apptainer/Singularity mounts CUDA drivers in a non-standard location, causing installation issues
- Work in progress

EESSI: current activities

Test suite (based on ReFrame)

- Challenge: should be extremely portable, and run on any host (laptop, VM, cluster). How?
- Solution: all system-specific info in ReFrame config file. Test should do ‘reasonable’ things based on that info. E.g.
  - Only generate GPU tests if there is a partition with GPUs
  - Run one MPI rank per core / GPU for pure CPU/GPU MPI applications
- Created a ‘blueprint’ for portable testing: GROMACS

EESSI: future activities

Bot refinements

- Retrigger failed builds / builds for specific architectures
- Better debugging (provide downloadable container for failed builds)
- Integration of test step in community contribution workflow

Test suite

- Low level tests
- More application tests
- Portable performance testing

EESSI: future activities

Expand hardware support

- AMD GPUs
- RISC-V

Training

- For end-users (first training @ HPCKP May 2023)
- For sysadmins

EESSI: future activities

Support extending EESSI with a local stack or site-specific CVMFS stack, e.g.

- proprietary software
- fast deployment (good QA on community contributions to EESSI takes time)
- software in development

Explore use case of EESSI in CI with scientific workpackages

- EESSI allows very quick deployment of all dependencies in a CI environment

From zero to science in 3 steps

- Step 1: Install and configure CernVM-FS
  - System-wide CernVM-FS installation (requires admin privileges)
  - Use container with CernVM-FS + EESSI configuration pre-installed

    (see https://eessi.github.io/docs/pilot/#accessing-the-eessi-pilot-repository-through-singularity)

https://eessi.github.io/docs/pilot

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

```
# Now:
$ sudo yum install -y cvmfs
$ sudo yum install -y https://github.com/EESSI/filesystem-layer/releases/download/latest/cvmfs-config-eessi-latest.noarch.rpm

# Later:
$ sudo yum install -y cvmfs
```

From zero to science in 3 steps

- Step 1: Install and configure CernVM-FS
  - System-wide CernVM-FS installation (requires admin privileges)
  - Use container with CernVM-FS + EESSI configuration pre-installed
- Step 2: Set up environment: source EESSI init script
- Step 3: Load module(s) and run!

https://eessi.github.io/docs/pilot

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

```
# Step 2: set up environment
$ source /cvmfs/pilot.eessi-hpc.org/latest/init/bash

# Step 3: load module(s) to activate software (check with ‘module avail’), and run!
[EESSI pilot 2021.12] $ module load GROMACS
[EESSI pilot 2021.12] $ gmx mdrun ...
```

Demo: seeing is believing

```
# Initialize EESSI environment
source /cvmfs/pilot.eessi-hpc.org/latest/init/bash
# Load module
module load GROMACS/2020.4-foss-2020a-Python-3.8.2
# Download gromacs test case
curl -LJO https://github.com/victorusu/GROMACS_Benchmark_Suite/raw/1.0.0/HECBioSim/hEGFRDimer/benchmark.tpr
# Run test case
mpirun -np 128 --bind-to core gmx_mpi mdrun -dlb yes -ntomp 1 -npme -1 -nb cpu -s benchmark.tpr
```

EESSI

```
# Load module
module load 2022
module load GROMACS/2021.6-foss-2022a
# Download gromacs test case
curl -LJO https://github.com/victorusu/GROMACS_Benchmark_Suite/raw/1.0.0/HECBioSim/hEGFRDimer/benchmark.tpr
# Run test case
mpirun -np 128 --bind-to core gmx_mpi mdrun -dlb yes -ntomp 1 -npme -1 -nb cpu -s benchmark.tpr
```

Local modules

How can you collaborate with EESSI

EESSI is fully open source and community driven

- Contribute new software
- Get involved in the development of EESSI
    - Join our Monthly online meetings (first Thursday, 2pm CEST)
    - Join our mailing list / Slack: https://www.eessi-hpc.org/join/
    - Join the discussion on Github: https://github.com/eessi
    - Docs: https://eessi.github.io/docs/
    - Twitter: @eessi_hpc
    - YouTube: https://www.youtube.com/@eessi_community

EESSI

EUROPEAN ENVIRONMENT FOR
SCIENTIFIC SOFTWARE INSTALLATIONS

Co-funded by the European Union

Web page: multixscale.eu

Twitter: @MultiXscale

LinkedIn: MultiXscale

Facebook: MultiXscale

Youtube channel: MultiXscale

Funded by the European Union. This work has received funding from the European High Performance Computing Joint Undertaking (JU) and countries participating in the project under grant agreement No 101093169.