



# Getting started with EESSI

*Thomas Röblitz, University of Bergen, Norway*

7th EasyBuild User Meeting (virtual) - 26 Jan 2022

# What is EESSI?

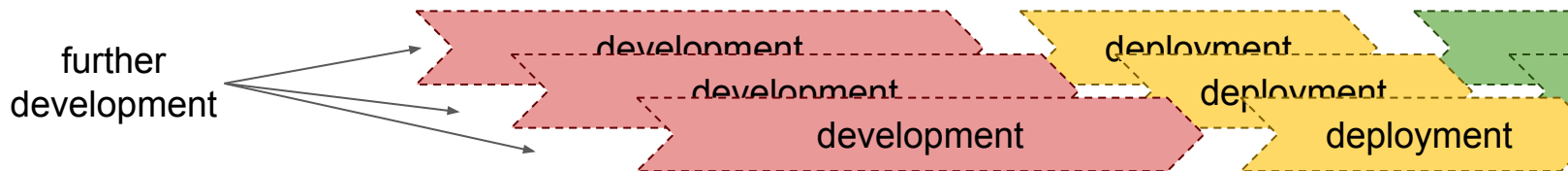
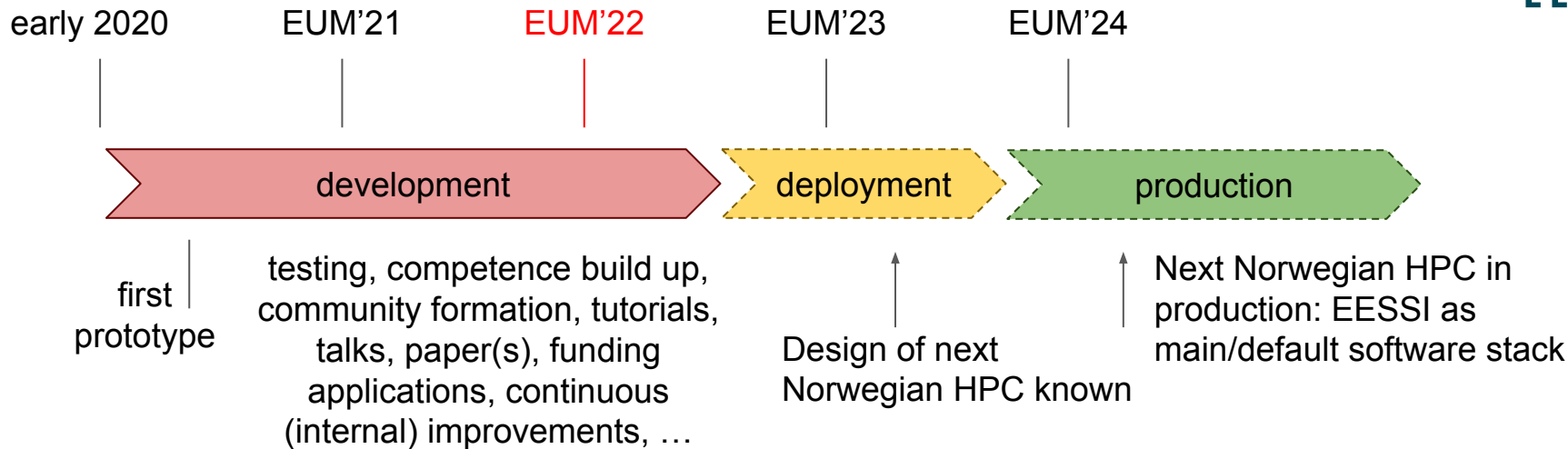
<https://www.eessi-hpc.org>

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



- The challenge:
  - Same software everywhere (HPC, Cloud, servers, laptops)
  - Optimized for specific CPUs, well tested, works on different OSs
  - Plug 'n play, limited setup
  
- The solution: EESSI - European Environment for Scientific Software Installations
  - “Streams” (scientific) software installations on-demand
  - Any machine, anywhere, nearly instantly available

# Motivation – EESSI history & future



# For whom is this talk? (and for whom it is not)

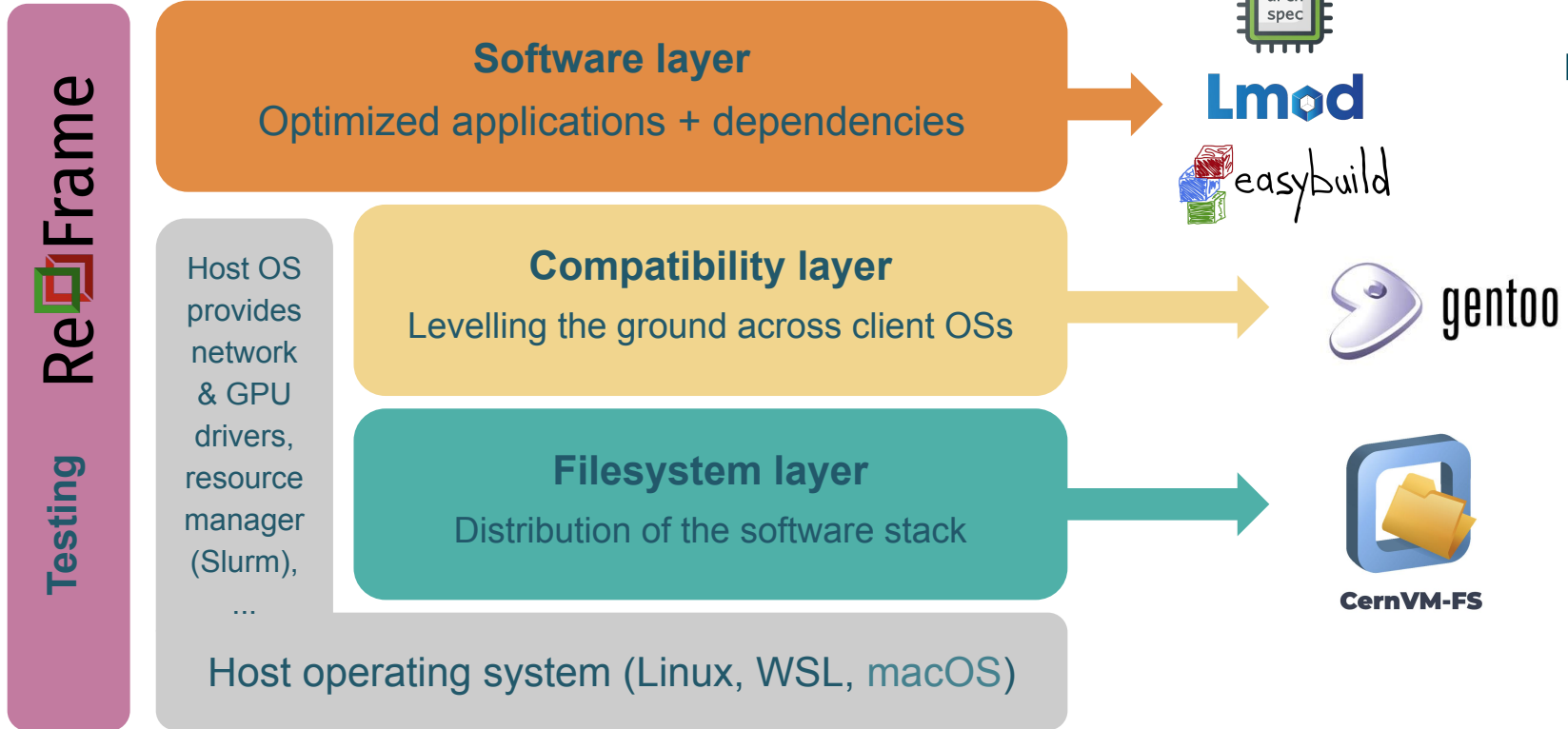


- Anyone who wants to try out EESSI (anywhere)
- System administrators who want to make EESSI available on their systems
- Developers (of scientific codes) who want to build on top of EESSI

*Any attempt to use EESSI and any feedback is highly encouraged!*

- Will not cover many EESSI internals or developing EESSI or extending EESSI
- **Disclaimer: EESSI pilot is not ready for production!**

# High-level overview of the EESSI project



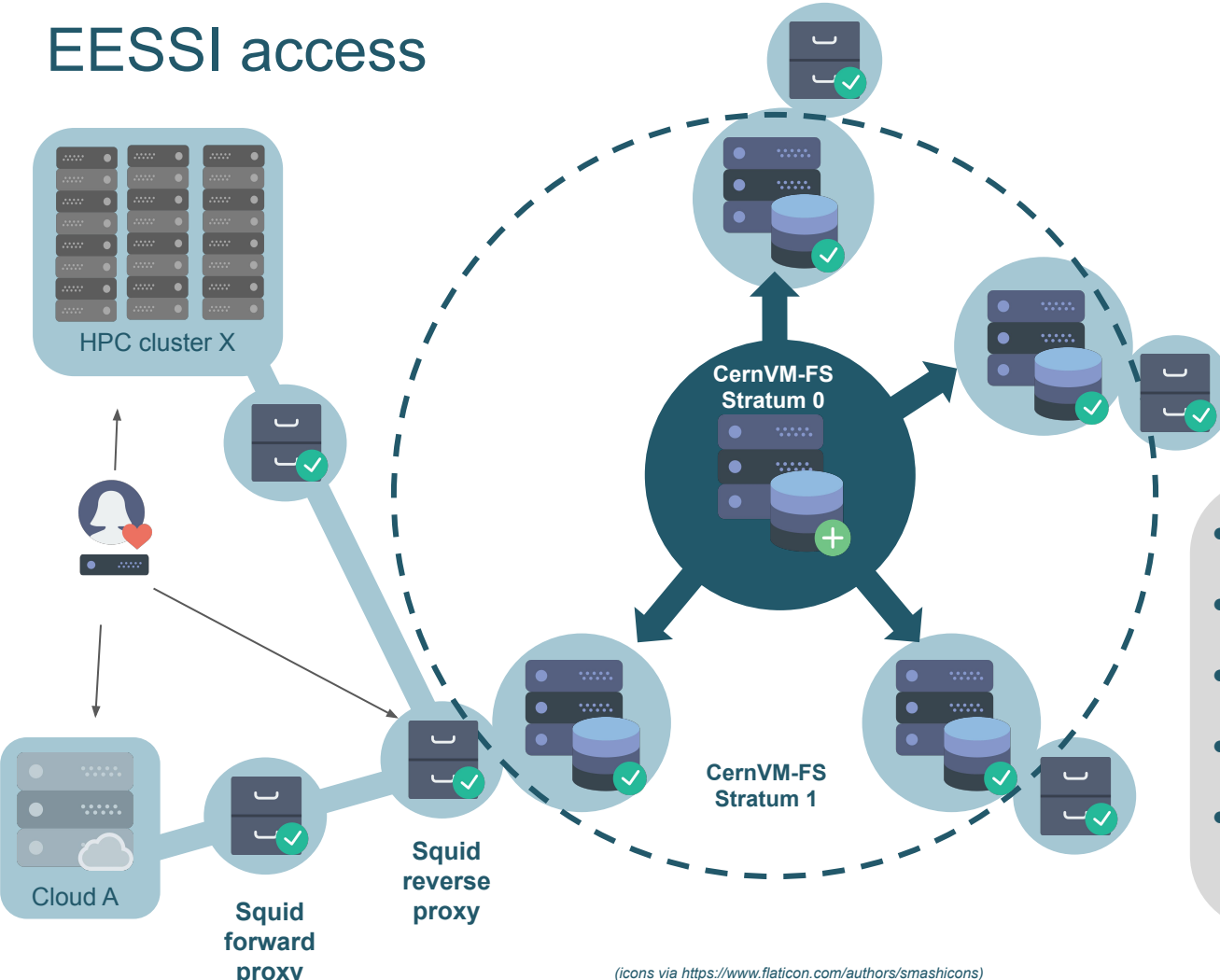
# EESSI access



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>)

# How is the EESSI software stack built? (simplified view)



1. Build **compatibility layer** for a CPU family (x86\_64, aarch64, ...)
    - Ingest it into repository on CernVM-FS central server
  2. Build **application software stack** for target CPU architecture(s) (Intel Skylake, AMD Rome, ...)
    - Ingest it into repository on CernVM-FS central server
- Traditionally building a software stack with EasyBuild ⇒ just step 2 on shared FS
    - Step 2 in EESSI: ensure compatibility layer is used + distribute installations via CernVM-FS
  - Containers vs EESSI ⇒ step 1 provides container-like abstraction for software included in EESSI



# Scenario 1: EESSI is available on your system (1/2)

Step 0: Is EESSI actually available?

```
$ ls /cvmfs/pilot.eessi-hpc.org  
host_injections  latest  versions
```

Step 1: Setup shell environment

```
source /cvmfs/pilot.eessi-hpc.org/latest/init/bash
```

Step 2: Use EESSI like you would use a “normal” EasyBuild/Lmod stack

```
ml av GROMACS  
ml GROMACS/2020.4-foss-2020a-Python-3.8.2  
which gmx  
gmx --version
```



# Scenario 1: EESSI is available on your system (2/2)



## Step 3: Running a job in Slurm

```
salloc --nodes=1 --exclusive ...

source /cvmfs/pilot.eessi-hpc.org/latest/init/bash

ml GROMACS/2020.4-foss-2020a-Python-3.8.2

URL=https://repository.prace-ri.eu/ueabs/GROMACS/1.2/GROMACS_TestCaseA.tar.gz
wget -c $URL -O - | tar -xz

OMP_NUM_THREADS=4 srun --mpi=pmix -n 4 gmx_mpi mdrun -s ion_channel.tpr -maxh
0.50 -reseedway -noconfout -nsteps 1000 -g logfile -dlb yes
```



# Demo!

Scenario 1: EESSI is available on your system

## Scenario 2: EESSI **not** available yet on your system (1/3)



Step 0: Is EESSI available?

```
$ ls /cvmfs/pilot.eessi-hpc.org
ls: cannot access '/cvmfs/pilot.eessi-hpc.org': No such file or
directory
```

Step 1: Do you have Singularity with support for `--fusemount`?

```
$ singularity --version
singularity version 3.7.4-1
# >= 3.6.0, better >= 3.7.4 (security fixes)
```

Step 2: Create temporary directories (not on a shared filesystem!)

```
export TMPDIR=$(mktemp -d) # add '-p /DIR' if /tmp too small
mkdir -p $TMPDIR/{var-lib-cvmfs,var-run-cvmfs,home}
```

## Scenario 2: EESSI **not** available yet on your system (2/3)



### Step 3: Prepare container environment

```
export SINGULARITY_BIND="$TMPDIR/var-lib-cvmfs:/var/lib/cvmfs, \  
                        $TMPDIR/var-run-cvmfs:/var/run/cvmfs" \  
export SINGULARITY_HOME="$TMPDIR/home:/home/$USER" \  
export EESSI_PILOT="container:cvmfs2 pilot.eessi-hpc.org \  
                  /cvmfs/pilot.eessi-hpc.org"
```

### Step 4: Launch EESSI client container

```
singularity shell --fusemount "$EESSI_PILOT" \  
docker://ghcr.io/eessi/client-pilot:centos7
```

## Scenario 2: EESSI **not** available yet on your system (3/3)



### Step 5: Setup shell environment

```
Singularity > source /cvmfs/pilot.eessi-hpc.org/latest/init/bash
```

### Step 6: Use EESSI as you would use a “normal” EasyBuild/Lmod stack

```
Singularity > ml av GROMACS
Singularity > ml GROMACS/2020.4-foss-2020a-Python-3.8.2
Singularity > which gmx
Singularity > gmx --version
...
Singularity > OMP_NUM_THREADS=16 gmx mdrun -s ion_channel.tpr -maxh
0.50 -reseedway -noconfout -nsteps 1000 -g logfile -dlb yes -ntmpi 1
```

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

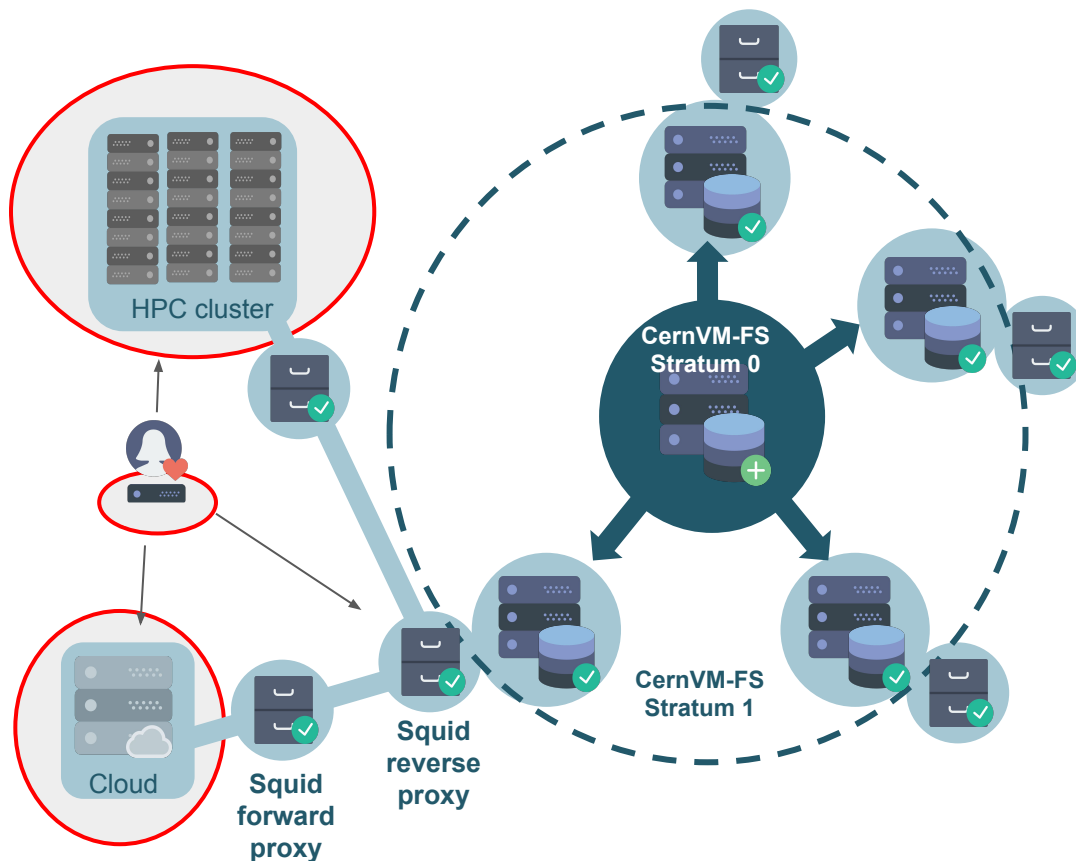
Alternative: use cvmfsexec on a recent OS, see <https://github.com/cvmfs/cvmfsexec>



# Demo!

Scenario 2: EESSI **not** available yet on your system

# Scenario 3: System administrator enables EESSI (1/3)



# Scenario 3: System administrator enables EESSI (2/3)



## Step 1: Install CernVM-FS client

```
RPM=https://ecsft.cern.ch/dist/cvmfs/cvmfs-release/cvmfs-release-latest.noarch.rpm
sudo yum install -y $RPM
sudo yum install -y cvmfs
```

## Step 2: Install EESSI configuration package + minimal CernVM-FS configuration

```
RPM=https://github.com/EESSI/filesystem-layer/releases/download/latest/
cvmfs-config-eesi-latest.noarch.rpm

sudo yum install -y $RPM
sudo bash -c "echo 'CVMFS_CLIENT_PROFILE="single"' > /etc/cvmfs/default.local"
sudo bash -c "echo 'CVMFS_QUOTA_LIMIT=10000' >> /etc/cvmfs # 10GB local cache"
```

## Step 3: Setup autofs

```
sudo cvmfs_config setup
```



# Scenario 3: System administrator enables EESSI (3/3)



Done!

EESSI can now be used as in Scenario 1

```
$ ls /cvmfs/pilot.eessi-hpc.org
host_injections latest versions
$ source /cvmfs/pilot.eessi-hpc.org/latest/init/bash
$ ml av GROMACS
...
```

A few noteworthy considerations for the simplest scenario:

- Machine (client) requires access to Internet
- CernVM-FS caches data by default under `/var/lib/cvmfs`
  - Adjust via configuration setting: `CVMFS_CACHE_BASE` ([see CernVM-FS docs](#))
- See more settings with

```
$ cvmfs_config showconfig pilot.eessi-hpc.org
```



# Demo!

Scenario 3: System administrator enables EESSI

# Scenario 3: System administrator enables EESSI



On larger systems (HPC, Clouds, larger number of desktops/laptops):

- **Should also set up local proxies + Stratum 1** (ensure availability, better performance, ...)
- Very detailed documentation at <https://cvmfs.readthedocs.io> and [https://eessi.github.io/docs/filesystem\\_layer/stratum1](https://eessi.github.io/docs/filesystem_layer/stratum1)
- See also CernVM-FS tutorial at EUM'21: <https://cvmfs-contrib.github.io/cvmfs-tutorial-2021>

On compute nodes with no Internet connection

- [CernVM-FS alien cache](#) on shared file system can be used
- Also useful for running MPI jobs when using EESSI via a container (no CernVM-FS on system)
- See also <https://github.com/EESSI/filesystem-layer/issues/37>

# Scenario 4: EESSI for developers (1/3)



Idea: Simplify CI for developers, for example using GitHub Actions

- Can be cumbersome to set up dependencies for running CI workflow
- Troubleshooting to get the CI workflow right can be time consuming

⇒ If EESSI is available on target system, would be great to use EESSI in CI workflow...

**Why spend time on dependencies if EESSI provides everything you need?**

Trivial to integrate in any project!

# Scenario 4: EESSI for developers (2/3)



Accessing EESSI in a GitHub Actions workflow is very... easy:

```
jobs:
  eessi:
    runs-on: ubuntu-20.04
    steps:
      - name: Check out repository
        uses: actions/checkout@v2
      - name: Mount EESSI CernVM-FS pilot repository
        uses: cvmfs-contrib/github-action-cvmfs@main
        with:
          # name of EESSI pilot repository
          cvmfs_repositories: pilot.eessi-hpc.org
          # EESSI configuration package (long download URL)
          cvmfs_config_package: https://github.com/EESSI/filesystem-layer/releases/download/latest/
            cvmfs-config-eessi_latest_all.deb
          # direct access to CernVM-FS network, no proxy
          cvmfs_http_proxy: DIRECT

      - name: Set up EESSI environment and run tests
        run: |
          source /cvmfs/pilot.eessi-hpc.org/versions/2021.12/init/bash
          ./run_tests.sh # what the developer really cares about, just load modules for dependencies!
```

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

[https://github.com/EESSI/eessi-demo/actions/workflows/pilot\\_repo\\_native.yml](https://github.com/EESSI/eessi-demo/actions/workflows/pilot_repo_native.yml)

[https://github.com/EESSI/eessi-demo/blob/main/.github/workflows/pilot\\_repo\\_native.yml](https://github.com/EESSI/eessi-demo/blob/main/.github/workflows/pilot_repo_native.yml)



# Scenario 4: EESSI for developers (3/3)



Summary

Jobs

- ✓ pilot\_repo\_native (Bioconduc...)
- ✓ pilot\_repo\_native (Bioconduc...)
- ✓ pilot\_repo\_native (GROMACS...)
- ✓ pilot\_repo\_native (GROMACS...)
- ✓ pilot\_repo\_native (OpenFOA...)
- ✓ pilot\_repo\_native (OpenFO...)
- ✓ pilot\_repo\_native (TensorFlo...)
- ✓ pilot\_repo\_native (TensorFlo...)

**pilot\_repo\_native (OpenFOAM, 2021.12)** succeeded 2 hours ago in 15m 10s

Search logs

- > ✓ Set up job 2s
- > ✓ Check out software-layer repository 1s
- > ✓ Mount EESSI CernVM-FS pilot repository 47s
- ✓ Run demo 14m 19s

```
1 ▶ Run source /cvmfs/pilot.eessi-hpc.org/versions/2021.12/init/bash
7 Found EESSI pilot repo @ /cvmfs/pilot.eessi-hpc.org/versions/2021.12!
8 Using x86_64/intel/haswell as software subdirectory.
9 Using /cvmfs/pilot.eessi-hpc.org/versions/2021.12/software/linux/x86_64/intel/haswell
  /modules/all as the directory to be added to MODULEPATH.
10 Found Lmod configuration file at /cvmfs/pilot.eessi-hpc.org/versions/2021.12/software
  /linux/x86_64/intel/haswell/.lmod/lmodrc.lua
11 Initializing Lmod...
12 Prepending /cvmfs/pilot.eessi-hpc.org/versions/2021.12/software/linux/x86_64/intel/haswell
  /modules/all to $MODULEPATH...
13 Environment set up to use EESSI pilot software stack, have fun!
14 /home/runner/work/eessi-demo/eessi-demo/OpenFOAM
15 WORKDIR: /tmp/runner/5019
16 /tmp/runner/5019
17 /tmp/runner/5019/motorBike
18 generating mesh...
19 New entry maxGlobalCells 200000000;
```

<https://github.com/EESSI/eessi-demo/runs/4950453207>

# Questions?

- EESSI *not ready for production yet*, but can be used for testing and feedback!
- Website: <https://www.eessi-hpc.org>
- Documentation: <https://eessi.github.io/docs>
- Introduction to EESSI (EUM'21): [https://www.youtube.com/watch?v=1CXwzIW\\_MsU](https://www.youtube.com/watch?v=1CXwzIW_MsU)
- **Join the EESSI mailing list and Slack:** <https://www.eessi-hpc.org/join>
- Monthly update meetings, open to join for anyone interested  
<https://github.com/EESSI/meetings/wiki>
- EESSI hackathons (Dec'21 + Jan'22, plans for more):  
<https://github.com/EESSI/hackathons>

