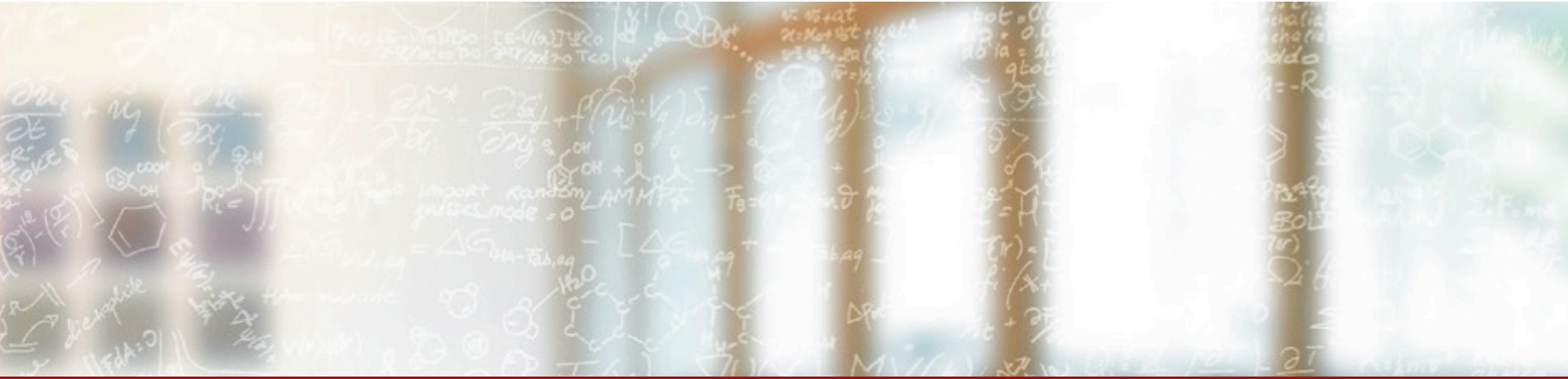




CSCS

Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

ETH zürich



EasyBuild @ CSCS: Site Update

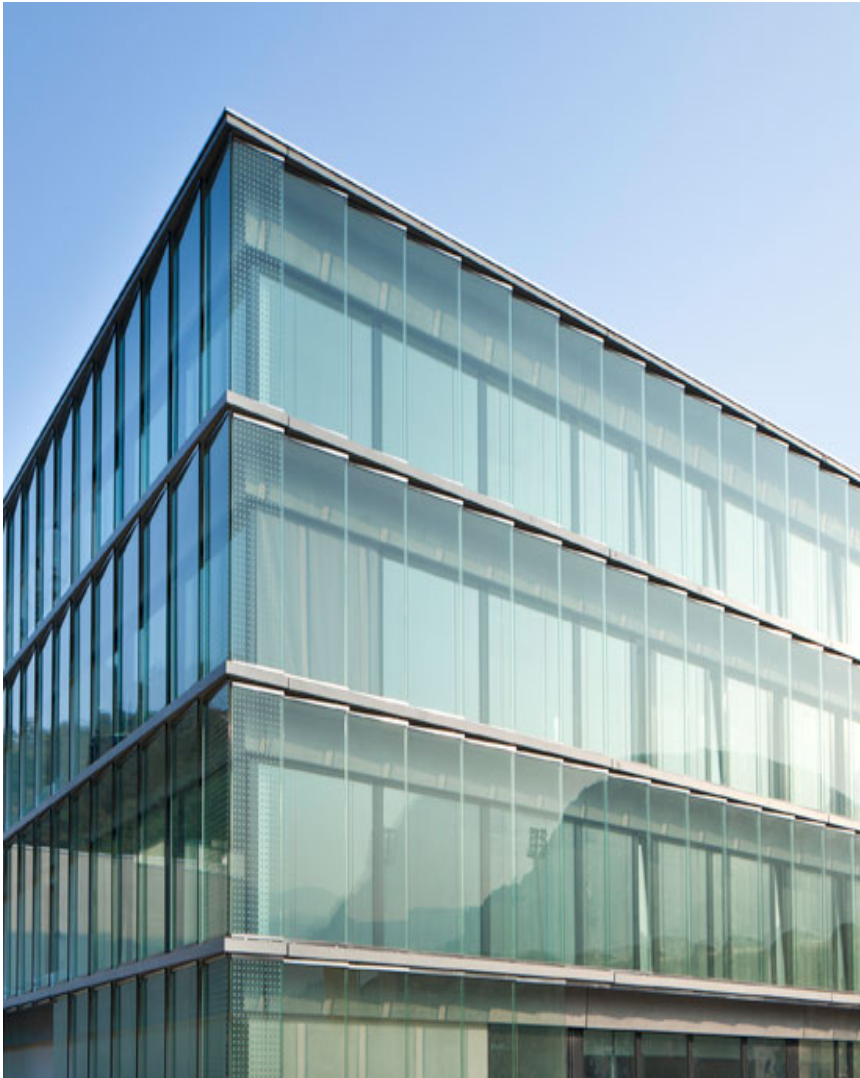
5th EasyBuild User Meeting

Jan 29th – 31st 2020, Barcelona

Luca Marsella

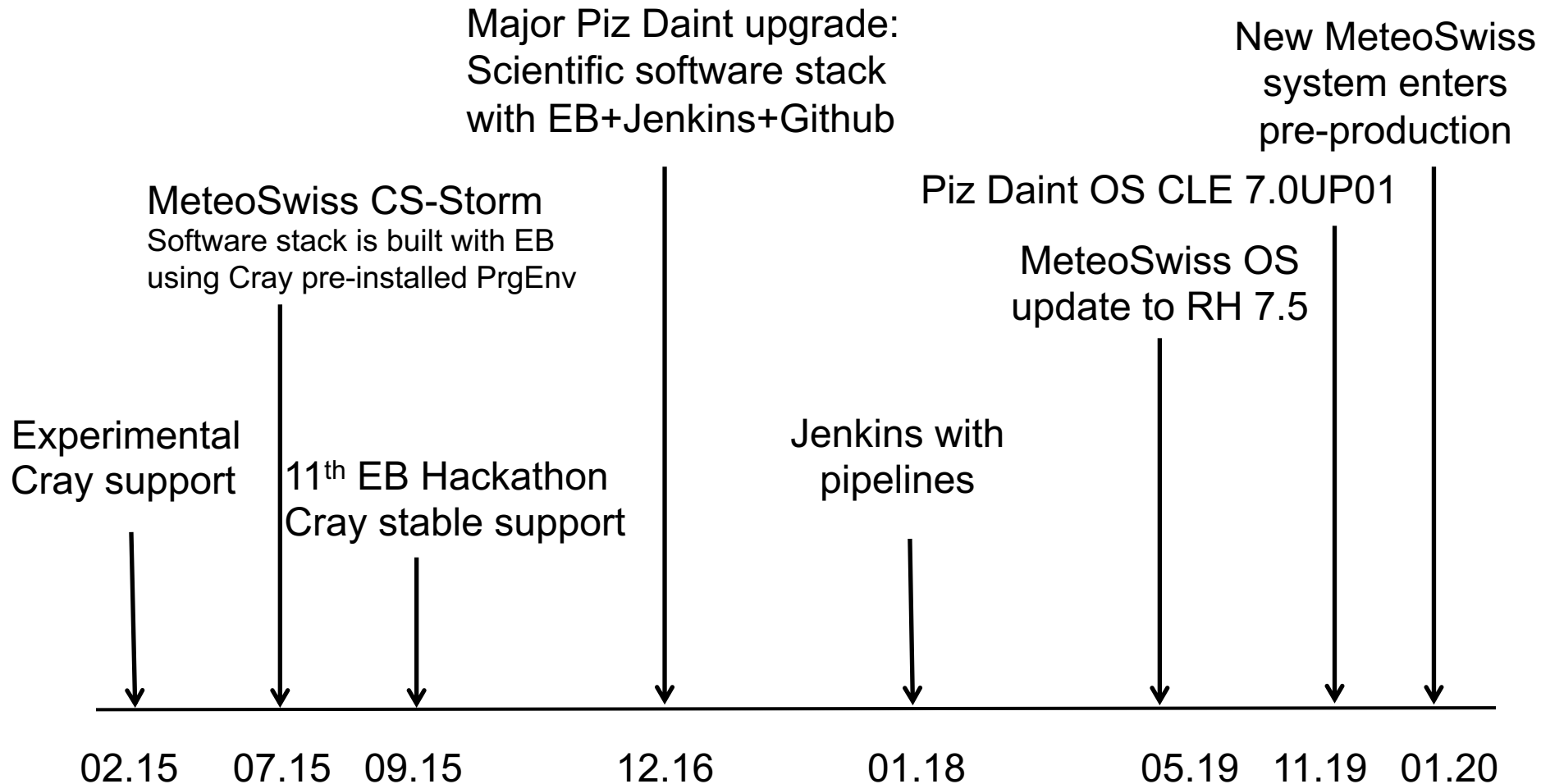
Scientific Computing Support, Swiss National Supercomputing Center (CSCS)

Outline

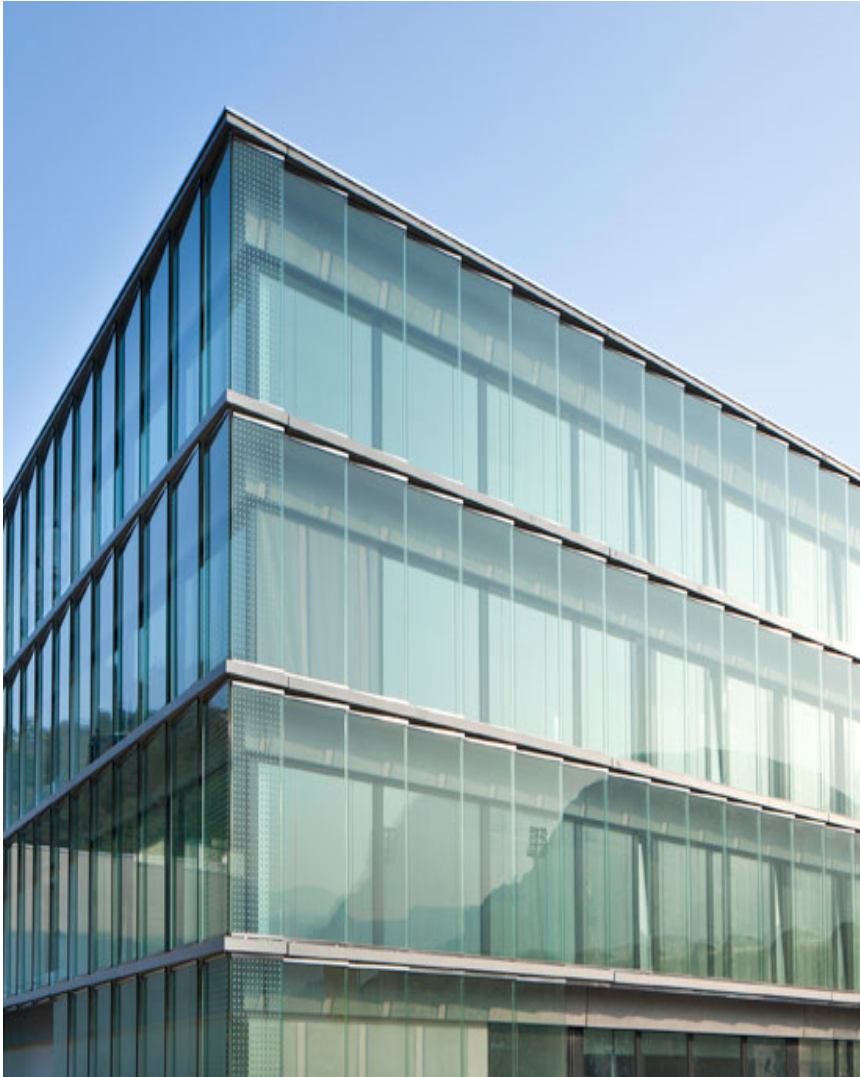


- **EasyBuild timeline @ CSCS**
- Overview of CSCS HPC systems
 - MeteoSwiss systems
 - Piz Daint
- EasyBuild for CSCS Users
- Easybuild with Jenkins
 - CSCS EB production repository
 - Jenkins pipelines
 - EB Testing pipeline
 - EB Regression pipeline
- Final Remarks

EasyBuild timeline @ CSCS



Outline



- EasyBuild timeline @ CSCS
- **Overview of CSCS HPC systems**
 - **MeteoSwiss systems**
 - **Piz Daint**
- EasyBuild for CSCS Users
- Easybuild with Jenkins
 - CSCS EB production repository
 - Jenkins pipelines
 - EB Testing pipeline
 - EB Regression pipeline
- Final Remarks

Overview of CSCS HPC systems

System	Scope	Accelerators / node	Architecture and OS
Piz Daint	CSCS User Lab	1 GPU	XC40 and XC50 CLE 7.0UP01
Escha	MeteoSwiss	16 GPU	Cray CS-Storm RedHat 7.5
Kesch	MeteoSwiss	16 GPU	Cray CS-Storm RedHat 7.5
Arolla / Tsa	MeteoSwiss	8 GPU	Intel SkyLake Tesla V100, RH 7.6
Fulen	OpenStack and large memory use cases	Mix of CPU only and GPU nodes	Intel Broadwell and SkyLake (fat), RH 7.4

MeteoSwiss Production System

MeteoSwiss (MCH): Swiss Weather Forecasting Service with a dedicated production system at CSCS

Escha and Kesch are two identical systems for production and failover featuring CS Storm nodes:

- EB software stack in production since 09.15
- GCC initially provided by Cray was unable to assemble AVX2 instructions for Haswell processor
- Successfully updated OS to RedHat 7.5 in 05.19

Architecture:

- 2 x Intel Haswell E5-2690v3 2.6 GHz
 - 12-core CPUs per node
 - 24 E5-2690v3 processors
- 256 GB 2133 MHz DDR4 memory per node
 - 3 TB total memory on the system
- 8 NVIDIA® Tesla® K80 GPU devices per node
 - 192 GPUs in total



MeteoSwiss Pre-production System

Arolla and Tsa are the names of the new system:

- Compute nodes ft. Intel Skylake and Tesla V100
- EB software stack in pre-production since 01.20

Module names are lowercase, with few exceptions:

- EasyBuild-custom (CSCS EasyBuild modulefile)
- PrgEnv-gnu
- PrgEnv-pgi

Meta-modules provide a hierarchical environment:

E.g.: PrgEnv-pgi/19.9 unfolds the following modules

- hdf5/1.10.5-pgi-19.9-gcc-8.3.0
- netcdf-c++/4.3.0-pgi-19.9-gcc-8.3.0
- netcdf-fortran/4.4.5-pgi-19.9-gcc-8.3.0
- netcdf/4.7.0-pgi-19.9-gcc-8.3.0
- openmpi/4.0.2-pgi-19.9-gcc-8.3.0-cuda-10.1
pgi/19.9-gcc-8.3.0



Piz Daint

Model	Cray XC50 / XC40
XC50 node	Intel® Xeon® E5-2690 v3 (Haswell) @ 2.60GHz (12 cores, 64GB RAM) and NVIDIA® Tesla® P100 16GB
XC40 node	Intel® Xeon® E5-2695 v4 (Broadwell) @ 2.10GHz (18 cores, 64/128 GB RAM)
Login node	Intel® Xeon® CPU E5-2650 v3 @ 2.30GHz (10 cores, 256 GB RAM)
Interconnect	Aries routing and communications ASIC, and Dragonfly network topology
Scratch	8.8 PB (Lustre / Sonexion 3000)

Flagship production system with hybrid nodes dedicated to User Lab at CSCS:

- EB software stack in production since 11.16
- Successfully updated OS to CLE 7.0UP01 in 11.19
- Automated conversion of Easyconfig files in production for OS updates

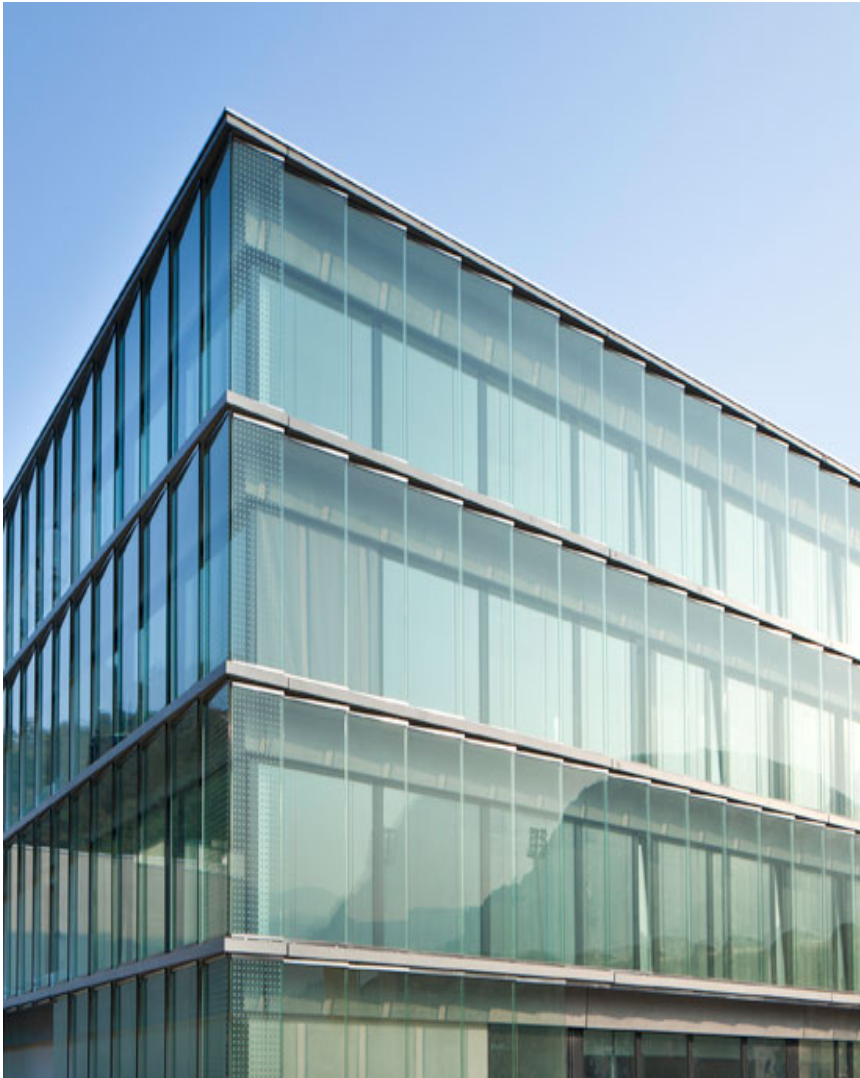
Work in progress: new toolchain CrayIntelMKL for direct usage of MKL library

Scientific Software built with EasyBuild on Piz Daint

Amber/18-14-14-CrayGNU-19.10-cuda-10.1
Boost/1.70.0-CrayGNU-19.10
CDO/1.9.5-CrayGNU-19.10
CP2K/6.1-CrayGNU-19.10-cuda-10.1
CPMD/4.1-CrayIntel-19.10
Charm++/6.8.0-CrayIntel-19.10
ELPA/2016.11.001.pre-CrayGNU-19.10
GREASY/19.03-cscs-CrayGNU-19.10
GROMACS/2019.1-CrayGNU-19.10-cuda-10.1
GSL/2.5-CrayGNU-19.10
HDFView/2.14
HPX/1.3.0-CrayGNU-19.10
Horovod/0.18.1-CrayGNU-19.10-tf-2.0.0
IDL/8.7.2-CSCS
LAMMPS/22Aug18-CrayGNU-19.10-cuda-10.1
Libint/1.1.4-CrayGNU-19.10
MATLAB/R2019a
NAMD/2.13-CrayIntel-19.10-cuda-10.1
NCL/6.4.0
NCO/4.8.1-CrayGNU-19.10
ParaView/5.7.0-CrayGNU-19.10-EGL
PLUMED/2.5.1-CrayGNU-19.10

PyExtensions/2.7.15.7-CrayGNU-19.10
PyExtensions/3.6.5.7-CrayGNU-19.10
QuantumESPRESSO/6.4.1-CrayIntel-19.10-cuda-10.1
Spark/2.3.1-CrayGNU-19.10-Hadoop-2.7
TensorFlow/2.0.0-CrayGNU-19.10-cuda-10.1.168
VASP/5.4.4-CrayIntel-19.10-cuda-10.
VMD/1.9.3-egl
VTK/8.2.0-EGL-CrayGNU-19.10
dask/2.2.0-CrayGNU-19.10-python3
h5py/2.8.0-CrayGNU-19.10-python2-parallel
ipykernel/4.8.2-CrayGNU-19.10-python2
ipyparallel/6.2.4-CrayGNU-19.10-python3
jupyterhub/1.0.0-CrayGNU-19.10
jupyterlab/1.1.1-CrayGNU-19.10
libxc/4.2.3-CrayGNU-19.10
magma/2.4.0-CrayIntel-19.10-cuda-10.1
ncview/2.1.7-CrayGNU-19.10
netcdf-python/1.4.1-CrayGNU-19.10-python3
numpy/1.17.2-CrayGNU-19.10
ospray/1.8.5-CrayGNU-19.10
pycuda/2018.1.1-CrayGNU-19.10-python3-cuda-10.1
singularity/3.5.2-daint

Outline



- EasyBuild timeline @ CSCS
- Overview of CSCS HPC systems
 - MeteoSwiss systems
 - Piz Daint
- **EasyBuild for CSCS Users**
- Easybuild with Jenkins
 - CSCS EB production repository
 - Jenkins pipelines
 - EB Testing pipeline
 - EB Regression pipeline
- Final Remarks

EasyBuild for CSCS Users

- User are given EasyBuild recipes when requesting software
 - Instead of error-prone manual steps on how to build and run
- EasyBuild documentation on the CSCS User Portal



CSCS

Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

CSCS User Portal

Getting Started ▾

Scientific Computing ▾

Storage ▾

My Projects ▾

HOME

Getting Started

Storage

Lates

23.11.2

[Piz Daint

Supported Applications

Code Compilation

Code Analysis

Technical Report

service

(10PM).

EasyBuild for CSCS Users



CSCS User Portal Getting Started ▾ Scientific Computing ▾ Storage ▾ My Projects ▾

SCIENTIFIC COMPUTING

Supported Applications

ABCpy

Amber

CP2K

CPMD

EasyBuild framework

Loading the environment

The [EasyBuild](#) framework is available at CSCS through the module `EasyBuild-custom`. This module defines the location of the EasyBuild configuration files, recipes and installation directories.

```
module load EasyBuild-custom
```

Outline



- EasyBuild timeline @ CSCS
- Overview of CSCS HPC systems
 - MeteoSwiss systems
 - Piz Daint
- EasyBuild for CSCS Users
- **Easybuild with Jenkins**
 - **CSCS EB production repository**
 - **Jenkins pipelines**
 - **EB Testing pipeline**
 - **EB Regression pipeline**
- Final Remarks

EasyBuild with Jenkins

- Jenkins for Continuous Integration
 - Test new pull requests submitted by staff and users
 - Deploy new software packages on the systems in production
 - Weekly Regression testing of current Easyconfigs
- Jenkins project names run with EasyBuild
 - EB Testing is triggered when a new Easyconfig appears on Github
 - EB Production builds the new recipes once they have been merged
 - EB Regression runs Easybuild from scratch to ensure reproducibility
- Jenkins projects defined by **Pipelines**
 - Enhanced flexibility of the actions performed by Jenkins
 - **Jenkinsfile** script of each project is version controlled
 - The CI can run in parallel optimizing the available resources

CSCS EB production repository on GitHub

Contributing back

How to submit a pull request:

1. Add the EasyBuild configuration files to a **new branch**, including **all the required dependencies**
2. Create and **assign yourself a pull request** following this policy for the title:
 - the title **must match a supported system** in the list `daint dom kesch leone monch`, otherwise the build will fail immediately. The system names **have to be enclosed in square brackets** to be distinguished from the actual pull request title and be parsed by the corresponding Jenkins project.
 - if the title matches `WIP` ("Work In Progress"), then the test build will be aborted immediately, as work in progress is not supposed to be tested.
 - Dom and Piz Daint can test both software stacks `-gpu` and `-mc` at once:
 - a. if the title matches only `${system}-gpu` or `${system}-mc`, only that software stack will be used:
 - `[dom-gpu]` NAMD will build using `-gpu`, `[dom-mc]` NAMD will use `-mc`.
 - b. if the title matches both or none, then both will be used, one after another in a loop:
 - `[dom]` NAMD will build using both `-gpu` and `-mc` in a loop.
 - `[dom-gpu, dom-mc]` NAMD will do the same.
3. The CSCS Jenkins project `TestingEB` will test the build of new EasyBuild recipes with respect to the master. The corresponding pipeline of `TestingEB` is contained in the `jenkins/JenkinsfileTestingEB` script.
4. If the build is successful, you should **ask for a review**: the pull request will only be merged when approved.
5. In order to re-trigger the testing of the pull request without committing a change, add the comment `retest this please` which will notify the `TestingEB` Jenkins project.
6. (CSCS only) for production builds, please update the appropriate production build list [here](#).

EB Testing Pipeline

Github Pull Request on [daint-gpu]

The screenshot shows a GitHub Pull Request (PR) for the repository 'eth-cscs / production'. The PR title is '[daint-gpu] Caffe2 & Caffe(1) easyconfigs #531'. The PR is merged and shows a list of commits and comments. A red box highlights the PR title, and a red arrow points from it to the 'daint-gpu' step in the pipeline screenshot on the right.

Pipeline triggered to run on [daint-gpu]

The screenshot shows the 'TestingEB 204' pipeline run. The pipeline is triggered by a commit on the 'daint-gpu' branch. The pipeline status is 'Succeeded' and the 'daint-gpu' step is highlighted with a red box. A red arrow points from the PR title in the left screenshot to this step.

Steps daint-gpu

Step	Duration
> General SCM	12s
> Shell Script	<1s
> Shell Script	<1s
> true -- Shell Script	<1s
> Shell Script	57m 25s

EB Testing Pipeline

Github Pull Request on [daint-gpu]

This screenshot shows a GitHub Pull Request (PR) for the repository 'eth-cscs/production'. The PR is titled '[daint] otf/2.1 #536'. A red box highlights the '[daint]' prefix in the title. The PR is created by user 'jgphpc' and is currently in a 'Review requested' state. The PR description is empty. The PR is linked to a commit '3bfe7ca'. The PR is also linked to a branch 'otf/2.1'.

Pipeline triggered to run on [daint-gpu]

This screenshot shows the 'TestingEB 209' pipeline interface. The pipeline is triggered by a GitHub pull request. The pipeline status is 'Succeeded'. The pipeline steps are: Start, Initialization, Machine Sel..., Build Stage, and End. The 'Build Stage' is highlighted with a red box and contains two sub-steps: 'daint-gpu' and 'daint-mc'. Below the pipeline diagram, the 'Steps daint-mc' are listed with their durations.

Step	Duration
> General SCM	24s
> Shell Script	<1s
> Shell Script	<1s
> true — Shell Script	<1s
> Shell Script	2m 40s

EB Regression Pipeline

- Runs weekly on each supported system
- Builds easyconfigs in production in parallel
- Software packages are downloaded from scratch
- Inspect results with the **Blue Ocean** interface

RegressionEB 740 Pipeline

Branch: — 1d 4h 19m 18s Changes by jfavre, noreply, schoengens, Samuel Omlin

Commit: — a day ago Started by timer

Start Initialization Machine Sel... Build Stage End

Build Stage tasks:

- daint-gpu
- daint-mc
- dom-gpu
- dom-mc
- kesch
- leone
- monch

Steps monch

✓	> General SCM	21s
✓	> Shell Script	<1s
✓	> Shell Script	<1s
✓	> The unuse path is: /abos/monch/UES/ienkins/RH6.9-17.06/easybuild/modules/all/ — Print Message	<1s
✓	> #!/bin/bash -l status=0 if [-d /mnt/lnec/ienscscs/RegressionEB/monch]: then rm -rf ... — Shell Script	2h 46m 3s

Outline



- EasyBuild timeline @ CSCS
- Overview of CSCS HPC systems
 - MeteoSwiss systems
 - Piz Daint
- EasyBuild for CSCS Users
- Easybuild with Jenkins
 - CSCS EB production repository
 - Jenkins pipelines
 - EB Testing pipeline
 - EB Regression pipeline
- **Final Remarks**

Final Remarks

- Moving HPC software stack to EB takes time
 - Learning curve
 - Resistance to change
- EasyConfigs vs. EasyBlocks
 - EasyBlocks
 - (+) Reusable: Great for well packaged & stable software
 - (-) Too much overhead for bleeding edge software
 - (-) Reproducibility: how to keep track of changes?
 - EasyConfigs
 - (-) Reuse by copy/paste (=> duplication)
 - **(+++)** Self contained recipes

Final Remarks

- Software Stack deployed on the new MCH system
 - Fully automated deployment with hierarchical module environment
 - foss/2019b for post-processing and fosscuda/2019b for computing
 - Custom module naming scheme: lowercase
- Automated recipe conversion for OS updates
 - New toolchain version defined in preparation of OS updates
 - Full software stack conversion managed automatically
 - Tested during the update of Piz Daint OS to CLE 7.0UP01
- Work in progress
 - EasyBuild Jenkins projects with additional ReFrame step for check
 - Full regression testing: build + sanity check + performance check
 - Naming scheme of builds and regression tests need to match

Useful links for EasyBuild @ CSCS

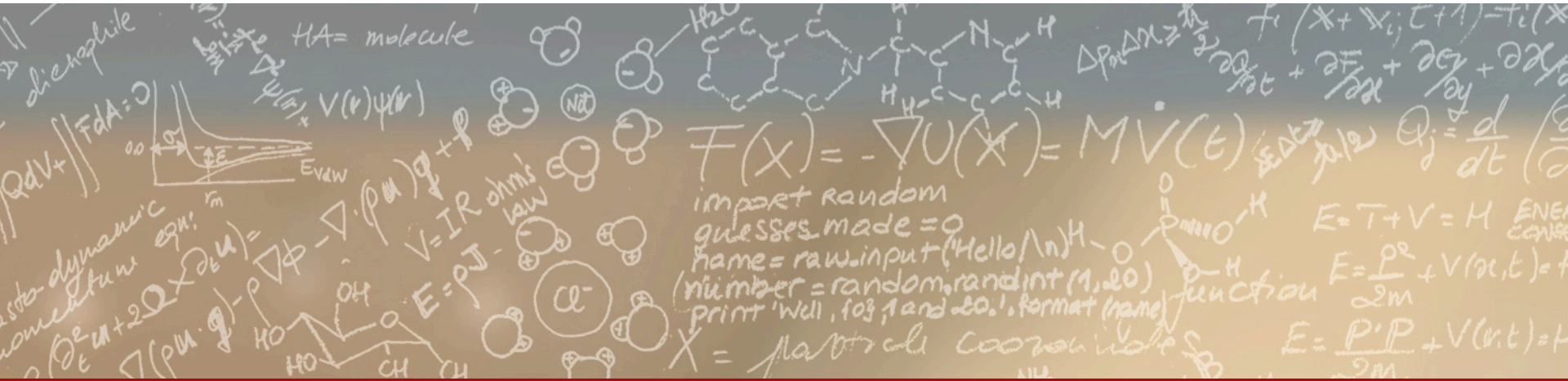
- EasyBuild User Documentation at CSCS
 - <https://user.cscs.ch/computing/compilation/easybuild>
- CSCS EasyBuild repositories
 - List of production builds performed by Jenkins
 - <https://github.com/eth-cscs/production/tree/master/jenkins-builds>
 - Custom easyconfigs:
 - <https://github.com/eth-cscs/production/tree/master/easybuild/easyconfigs>
 - Custom easyblocks:
 - <https://github.com/eth-cscs/production/tree/master/easybuild/easyblocks>
 - CSCS repos mirrored under the EasyBuilders GitHub repository:
 - <https://github.com/easybuilders/CSCS>



CSCS

Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

ETH zürich



Thank you for your kind attention