EasyBuild @SURF

Maxim Masterov
HPC advisor
SURF, The Netherlands

SURF

- Supercomputing
- Clustercomputing
- Scientific visualisation
- Data services
- Research Cloud
- Grid/Spider
- HPML
- Digital platforms
- Security, trust & identity
- Network connectivity
- Training, consultancy
- …

SURF

HPCV group

- **Supercomputing**
- **Clustercomputing**
- **Scientific visualisation**
- Data services
- Research Cloud
- Grid/Spider

- HPML
- Digital platforms
- Security, trust & identity
- Network connectivity
- **Training, consultancy**
- …

Systems

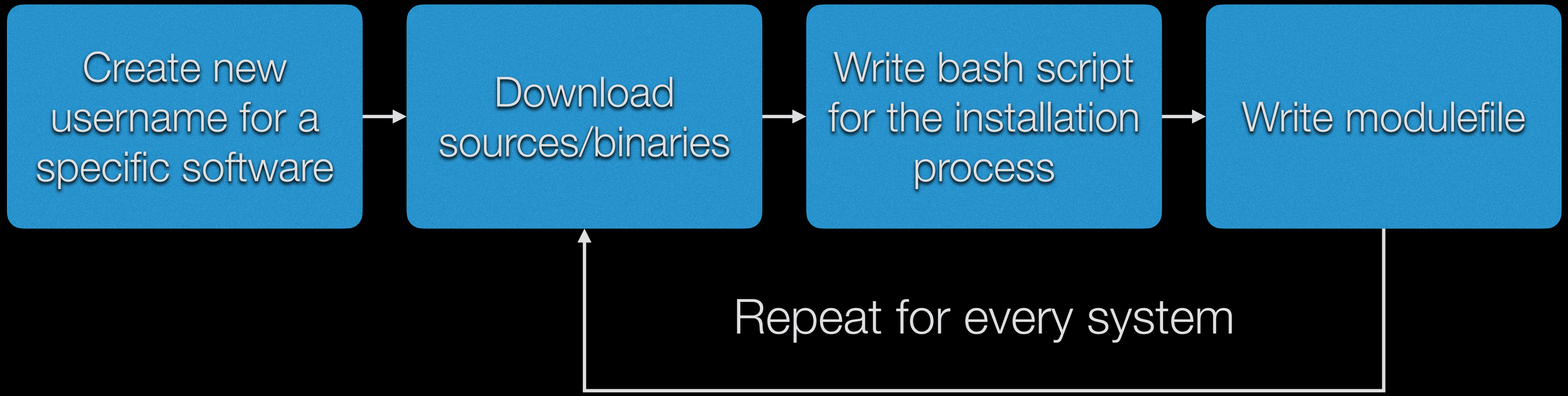
- **Cartesius** - former national supercomputer (Broadwell, Haswell, Ivy Bridge, Sandy Bridge, KNL)
- **Snellius** - new national supercomputer (Ice Lake, AMD ROME)
- **Lisa** - national cluster (Skylake, Cascade Lake)
- **ESC** - test system (Cascade Lake, AMD ROME)

Problems

- Large number of modules (we provide software stacks for our users)
- Some modules are “unique” for a particular users group or project (software may not comply with our software policy)
- Heterogeneous systems (different architectures, features)
- Multiple systems (need to install software on all of them)

A bit of history

- The old way of installing/maintaining the software



A bit of history

- Huge number of “software users”
- Manually written scripts
- Manually written modulefiles
- No tests or sanity checks
- Independent software environments on all systems
- Different module environments (Tmod, Lmod)
- …

Current management

- **Lmod** - module environment
- **Jenkins** - automation
  - **EasyBuild** - software installation
  - **ReFrame** - software testing
  - **Confluence** - docs
- **Xalt** - usage monitoring

Problem #1

- Large number of modules (we provide software stacks for our users)
- Some modules are “unique” for a particular users group or project (software may not comply with our software policy)
- Heterogeneous systems (different architectures, features)
- Multiple systems (need to install software on all of them)

Module environment

- Lmod - same module environment on all systems
  - Spider cache on (local and global)
  - **LMOD_EXACT_MATCH = yes**. Users have to specify full module names. Simplifies our helpdesk duties
  - **LMOD_EXTENDED_DEFAULT = no**. No partial match of a version
  - **LMOD_CASE_INDEPENDENT_SORTING = yes**. Simplifies module search for users

Module environment

- Global cache is updated at the end of Jenkins pipeline
- Local cache (in $HOME) is updated after calling “eblocalinstall”
- “eblocalinstall” is a wrapper over the “eb” command. It re-defines EASYBUILD_INSTALLPATH_XXX envars

```
$ cat /opt/lmod/lmod/init/lmodrc.lua | tail -n 10
scDescriptT = {
  {
    dir = “/sw/arch/lmod_cache/cacheDir",
    timestamp = "/sw/arch/lmod_cache/system.txt",
  },
  {
    dir = "~/.lmod.d/.cache",
    timestamp = "~/.lmod.d/users.txt",
  },
}
```

```
$ cat eblocalinstall | tail -n 8
# Update the local cache (Lmod only)
read module_system_version < <( _check_module_system ) || exit 1
if [ "$module_system_version" == "Lmod" ]
then
  $LMOD_PKG/libexec/update_lmod_system_cache_files \
  -d ~/.lmod.d/.cache -t ~/.lmod.d/users.txt \
  $EASYBUILD_INSTALLPATH_MODULES
fi

exit $exitcode
```

Software stacks

- One software stack release per year
- Release in August/September
- At most three software stacks on a system:
  - Production - full support, new software installation
  - PreviousProduction - limited support, patching
  - Deprecated - no support
- We use **Xalt** to track the software usage

```
$ module av

-------------- /sw/noarch/modulefiles/environment --------------
   2019    2020    2021    slurm-tools

$ module load 2020
$ module av

-------------- /sw/noarch/modulefiles/environment --------------
   2019    2020 (L)    2021    slurm-tools

---- /sw/arch/Debian10/EB_production/2020/modulefiles/phys -----
   Elk/6.3.2-foss-2020a            VASP5/5.4.4.pl2-intel-2020a
   Elk/6.3.2-intel-2020a           VASP5/5.4.4.pl2-intelcuda-2020a
   UDUNITS/2.2.26-GCCcore-9.3.0    VASP6/6.1.1-intelcuda-2020a

---- /sw/arch/Debian10/EB_production/2020/modulefiles/perf -----
   CubeGUI/4.4.4-GCCcore-9.3.0          OPARI2/2.0.5-GCCcore-9.3.0
   CubeLib/4.4.4-GCCcore-9.3.0          OTF2/2.2-GCCcore-9.3.0
   CubeWriter/4.4.3-GCCcore-9.3.0       PAPI/6.0.0-GCCcore-9.3.0
...
```

Software stacks

Year/Stack	2019	2020	2021	2022	2023	…
2019a	Production	Limited support	Deprecated	Deleted		
2020a		Production	Limited support	Deprecated	Deleted	
2021a			Production	Limited support	Deprecated	
2022a				Production	Limited support	
2023a					Production	
…						

Software stacks

- Prior to **2021a** we installed all software with both “foss” and “intel” toolchains
- Since **2021a** we install everything with “foss”. Only a few packages are still installed with “intel” (e.g. AMS, NWChem, OpenMolcas)
- We provide an additional set of compilers and development tools for experienced users: OneAPI, LLVM, NVHPC, AOCC
- We have relatively strict software policy

Software policy

- Requests for new software:
  - We **do not install** software that is older than 2 years unless the software is still maintained and relevant
  - We **don't install more than two different versions** of software per toolchain system-wide
  - We **don't install versions older than** the one already **available** system-wide (unless the software has no backward compatibility). These situations should be assessed on a case-by-case basis
  - Modules for **new versions** will **replace** the **existing** modules with older **versions** (unless the software has no backward compatibility)
  - The **default approach** is **"users build software themselves in the local prefix"**. We may assist in installing software locally, but we avoid doing it ourselves (unless a user has consultancy hours added to the project)

Modules environment

- 2018 - approx. 700 modules
- 2019 - approx. 600 modules (introduced software policy)
- 2020 - approx. 470 modules (introduced Xalt)
- 2021 - approx. 380 modules (only foss toolchain)

Problem #2

- Large number of modules (we provide software stacks for our users)
- Some modules are “unique” for a particular users group or project (software installation may not comply with our software policy)
- Heterogeneous systems (different architectures, features)
- Multiple systems (need to install software on all of them)

Sub-stacks

- We are involved in multiple projects (Deltares, OSSC, CompBiomed, ReaxPro, etc.)
- Some projects require installation of additional software (e.g. proprietary software or software that depends on “intel” toolchain)
- We do not want to “spoil” our users by sharing additional software with them :)
- We want to avoid duplicates among the installed software

Sub-stacks

- We introduce **sub-stacks**
- The sub-stack depends on the basic software stack from the same year, but extends it
- Only some users can see/use a sub-stacks (ACLs)
- The MODULEPATH is populated in the sub-stack modulefile

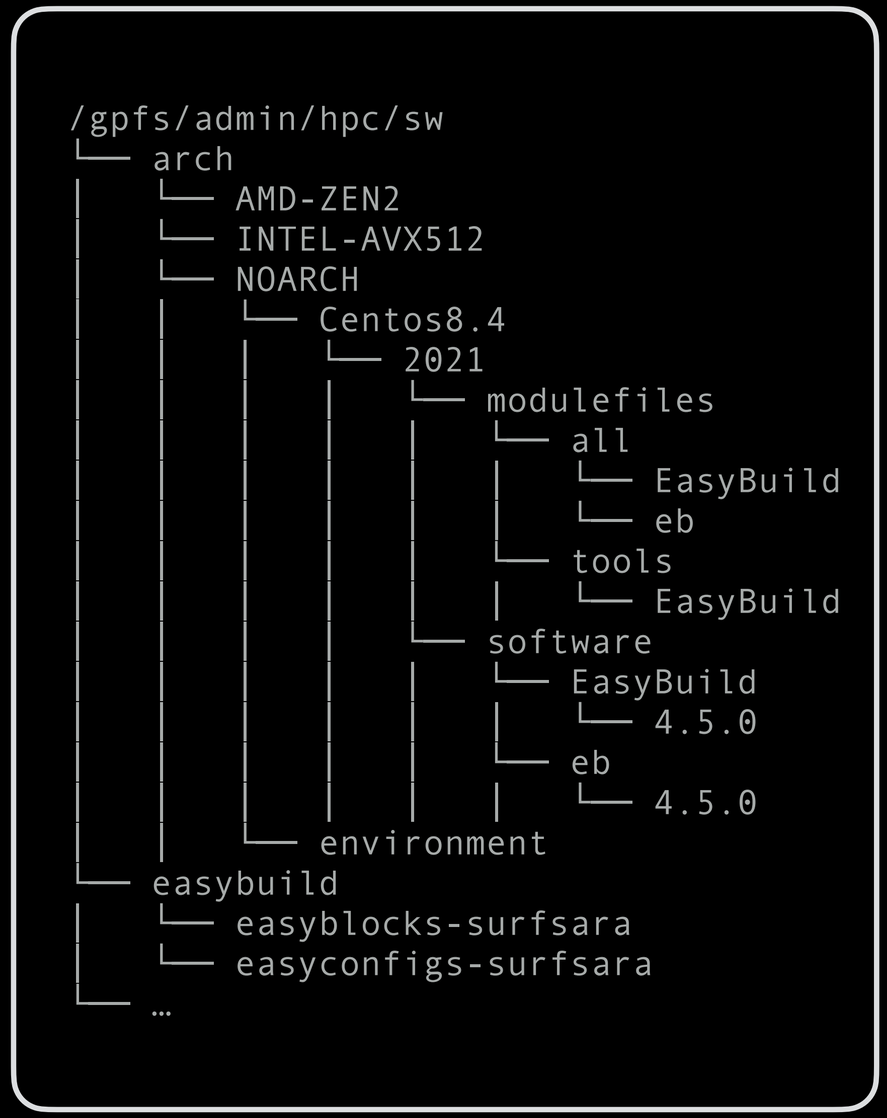
```
$ module av

------------------------------- /sw/noarch/environment --------------------------------
   2020    2021    2021_Delft3D    2021_OSSC
```

Problem #3

- Large number of modules (we provide software stacks for our users)
- Some modules are “unique” for a particular users group or project (software may not comply with our software policy)
- Heterogeneous systems (different architectures, features)
- Multiple systems (need to install software on all of them)

Heterogeneity

- The “common” symlinks (Snellius):
  - AMD nodes:
    - /sw/arch -> /gpfs/admin/hpc/sw/arch/AMD-ZEN2
  - Intel nodes:
    - /sw/arch -> /gpfs/admin/hpc/sw/arch/INTEL-AVX512
  - All nodes (architecture agnostic modules/software):
    - /sw/noarch -> /gpfs/admin/hpc/sw/arch/NOARCH



Problem #4

- Large number of modules (we provide software stacks for our users)
- Some modules are “unique” for a particular users group or project (software may not comply with our software policy)
- Heterogeneous systems (different architectures, features)
- Multiple systems (need to install software on all of them)

Jenkins

- Automated software installation (Groovy, Bash + EasyBuild)
- Automated regression tests (Groovy, Bash + ReFrame)
- Automated documentation generation (Groovy, Bash, Python + REST API)
- Ideal workflow: install -> test -> document
- Currently, all three are decoupled (WIP)

Jenkins

- Multiple pipelines
- Some are executed automatically (regression tests)
- Some must be started manually (sw installation)

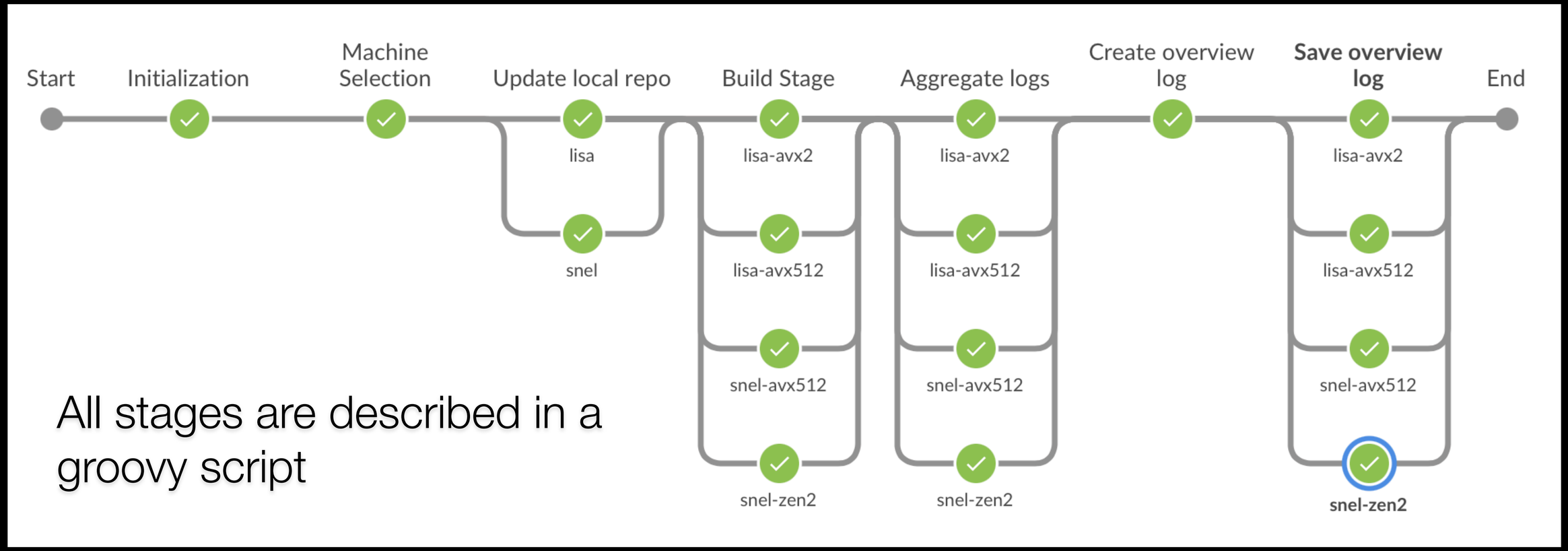
EasyConfigDelft3D		-	-	☆
EasyConfigNextProduction		-	-	☆
EasyConfigOSSC		-	-	☆
EasyConfigPreviousProduction		-	-	☆
EasyConfigProduction		-	-	☆
EasyConfigRegression		-	-	☆
EasyConfigTest		-	-	☆
ReFrameOSUpgradeMajor		-	-	☆
ReFrameOSUpgradeMinor		-	-	☆
ReFrameProductionMonthly		-	-	☆
ReFrameProductionWeekly		-	-	☆
ReFrameProductionYearly		-	-	☆
ReFrameSABS2020		-	-	☆
ReFrameSlurmUpgrade		-	-	☆
ReFrameTestingPipeline		-	-	☆

Jenkins

- Workflow (software installation):
  - parse a “buildlist” to check what should be installed
    - “buildlist” - plain text file with names of easyconfigs + some options (e.g. --from-pr)
  - allocate resources on a target machine and architecture
  - load modules (software stack + EasyBuild)
  - run installation
  - update global spider cache

```
#### Basic Compilers ####
GCCcore-10.3.0.eb
intel-compilers-2021.2.0.eb \
--accept-eula-for=Intel-oneAPI
#
#### Basic Components ####
binutils-2.36.1.eb
binutils-2.36.1-GCCcore-10.3.0.eb
pkg-config-0.29.2.eb
pkg-config-0.29.2-GCCcore-10.3.0.eb
Autotools-20210128-GCCcore-10.3.0.eb
ncurses-5.9.eb --from-pr 14144 \
# we need this specific version for Stata-17
ncurses-6.2-GCCcore-10.3.0.eb
git-2.32.0-GCCcore-10.3.0-nodocs.eb
Mercurial-5.8-GCCcore-10.3.0.eb
#### MPI Libraries ###
OpenMPI-4.1.1-GCC-10.3.0.eb \
--hooks=/sw/eb/easyconfigs-surf/hooks/\
mpi_hook.py --include-easyblocks=/sw/eb/\
easyblocks-surf/openmpi.py
OpenMPI-4.1.1-intel-compilers-2021.2.0.eb \
--hooks=/sw/eb/easyconfigs-surf/hooks/\
mpi_hook.py --include-easyblocks=/sw/eb/\
easyblocks-surf/openmpi.py
impi-2021.2.0-intel-compilers-2021.2.0.eb \
--accept-eula-for=Intel-oneAPI \
--hooks=/sw/\eb/ easyconfigs-surf/hooks/\
mpi_hook.py
MPICH-3.4.2-GCC-10.3.0.eb
#
…
```

Jenkins



All stages are described in a groovy script

Jenkins

Shell Script 1m 22s

```
AMS-2021.101-intelmpi.eb                                    ALREADY INSTALLED
Gaussian-g16.c01-AVX2.eb                                    ALREADY INSTALLED
NAMD-2.14-foss-2021a-mpi.eb                                 ALREADY INSTALLED
NAMD-2.14-foss-2021a-mpi-memopt.eb                          ALREADY INSTALLED
ORCA-5.0.1-gompi-2021a.eb                                   ALREADY INSTALLED
PLUMED-2.7.2-foss-2021a.eb                                  ALREADY INSTALLED
TURBOMOLE-7.5.1.eb                                          ALREADY INSTALLED
Libint-2.6.0-GCC-10.3.0-lmax-6-cp2k.eb                      ALREADY INSTALLED
NWChem-7.0.2-intel-2021a.eb                                 ALREADY INSTALLED
CP2K-8.2-foss-2021a.eb                                      ALREADY INSTALLED
Lumerical-2021-R2.3-2834-e18f3c9-OpenMPI-4.1.1.eb           ALREADY INSTALLED
Lumerical-2021-R2.3-2834-e18f3c9-impi-2021.2.0.eb           ALREADY INSTALLED
ANSYS-2021R2.eb                                             ALREADY INSTALLED
OpenFOAM-v2106-foss-2021a.eb                                ALREADY INSTALLED
UDUNITS-2.2.28-GCCcore-10.3.0.eb                            ALREADY INSTALLED
Elk-7.2.42-foss-2021a.eb                                    ALREADY INSTALLED
VASP5-5.4.4.pl2-foss-2021a-CUDA-11.3.1.eb                   ALREADY INSTALLED
VASP5-5.4.4-intel-2021a-CUDA-11.3.1.eb                      ALREADY INSTALLED
VASP6-6.2.1-foss-2021a-CUDA-11.3.1.eb                       ALREADY INSTALLED
VASP6-6.2.1-intel-2021a-CUDA-11.3.1.eb                      ALREADY INSTALLED
CubeWriter-4.6-GCCcore-10.3.0.eb                            ALREADY INSTALLED
CubeLib-4.6-GCCcore-10.3.0.eb                               ALREADY INSTALLED
CubeGUI-4.6-GCCcore-10.3.0.eb                               ALREADY INSTALLED
gperftools-2.9.1-GCCcore-10.3.0.eb                          ALREADY INSTALLED
HPL-2.3-foss-2021a.eb                                       ALREADY INSTALLED
OTF2-2.3-GCCcore-10.3.0.eb                                  ALREADY INSTALLED
PAPI-6.0.0.1-GCCcore-10.3.0.eb                              ALREADY INSTALLED
Paraver-4.9.2-foss-2021a.eb                                 ALREADY INSTALLED
Score-P-7.0-gompi-2021a.eb                                  ALREADY INSTALLED
Scalasca-2.6-gompi-2021a.eb                                 ALREADY INSTALLED
HarfBuzz-2.8.1-GCCcore-10.3.0.eb                            ALREADY INSTALLED
Pango-1.48.5-GCCcore-10.3.0.eb                              ALREADY INSTALLED
ATK-2.36.0-GCCcore-10.3.0.eb                                ALREADY INSTALLED
GTK+-2.24.33-GCCcore-10.3.0.eb                              ALREADY INSTALLED
Gdk-Pixbuf-2.42.6-GCCcore-10.3.0.eb                         ALREADY INSTALLED
GTS-0.7.6-GCCcore-10.3.0.eb                                 ALREADY INSTALLED
Graphviz-2.47.2-GCCcore-10.3.0.eb                           ALREADY INSTALLED
libgd-2.3.1-GCCcore-10.3.0.eb                               ALREADY INSTALLED
SWIG-4.0.2-GCCcore-10.3.0.eb                                ALREADY INSTALLED
FFmpeg-4.3.2-GCCcore-10.3.0.eb                              ALREADY INSTALLED
libGLU-9.0.1-GCCcore-10.3.0.eb                              ALREADY INSTALLED
gnuplot-5.4.2-GCCcore-10.3.0.eb                             ALREADY INSTALLED
matplotlib-3.4.2-foss-2021a.eb                              ALREADY INSTALLED
VTK-9.0.1-foss-2021a.eb                                     ALREADY INSTALLED
wxWidgets-3.1.5-GCC-10.3.0.eb                               ALREADY INSTALLED
Blender-2.93.7-lts.eb                                       SUCCESS
ParaView-5.9.1-foss-2021a-mpi.eb                            ALREADY INSTALLED
TurboVNC-2.2.6-GCCcore-10.3.0.eb                            ALREADY INSTALLED
VirtualGL-2.6.5-GCCcore-10.3.0.eb                           ALREADY INSTALLED
OpenCV-4.5.3-foss-2021a-CUDA-11.3.1-contrib.eb              ALREADY INSTALLED
ASAP-2.0-foss-2021a-CUDA-11.3.1.eb                          ALREADY INSTALLED
remotevis-git.eb                                            ALREADY INSTALLED
ncview-2.1.8-gompi-2021a.eb                                 ALREADY INSTALLED
WRF-4.3-foss-2021a-dmpar.eb                                 ALREADY INSTALLED
WPS-4.3.1-foss-2021a-dmpar.eb                               ALREADY INSTALLED
ESMF-8.1.1-foss-2021a.eb                                    ALREADY INSTALLED
TensorFlow-2.6.0-foss-2021a-CUDA-11.3.1.eb                  ALREADY INSTALLED
Horovod-0.22.1-foss-2021a-CUDA-11.3.1-TensorFlow-2.6.0.eb   ALREADY INSTALLED
PyTorch-1.10.0-foss-2021a-CUDA-11.3.1.eb                    ALREADY INSTALLED
torchvision-0.11.1-foss-2021a-CUDA-11.3.1.eb                ALREADY INSTALLED
Horovod-0.23.0-foss-2021a-PyTorch-1.10.0.eb                 ALREADY INSTALLED
IPython-7.25.0-GCCcore-10.3.0.eb                            ALREADY INSTALLED
JupyterHub-1.4.1-GCCcore-10.3.0.eb                          ALREADY INSTALLED
jupyter-server-proxy-3.1.0-GCCcore-10.3.0.eb                ALREADY INSTALLED
jupyterlmod-2.0.2-GCCcore-10.3.0.eb                         ALREADY INSTALLED
IRkernel-1.2-foss-2021a.eb                                  ALREADY INSTALLED
supporting-utils.eb                                         SUCCESS
==========
Built succesfully: 2/215
Already installed: 213/215
Failed: 0/215
Skipped: 0/215
Missing easyconfigs: 0/215
==========
Return status: 0
```

Shell Script 3m 15s

```
OpenMOLCAS-21.06-intel-2021a.eb                             ALREADY INSTALLED
Libint-2.6.0-GCC-10.3.0-lmax-6-cp2k.eb                      ALREADY INSTALLED
libecpint-1.0.7-foss-2021a.eb                               FAILED
    Standard output: /gpfs/admin/_hpc/sw/arch/INTEL-AVX512/Centos8/EB_production/2021/logs/20220112_102349/libecpint-1.0.7-foss-2021a.stdout.log
    EasyBuild log: /gpfs/admin/_hpc/sw/arch/INTEL-AVX512/Centos8/EB_production/2021/logs/20220112_102349/libecpint-1.0.7-foss-2021a.ebout.log
    Last succesful build: Never build successfully
NWChem-7.0.2-intel-2021a.eb                                 ALREADY INSTALLED
CP2K-8.2-foss-2021a.eb                                      ALREADY INSTALLED
Lumerical-2021-R2.3-2834-e18f3c9-OpenMPI-4.1.1.eb           ALREADY INSTALLED
Lumerical-2021-R2.3-2834-e18f3c9-impi-2021.2.0.eb           ALREADY INSTALLED
ANSYS-2021R2.eb                                             ALREADY INSTALLED
OpenFOAM-v2106-foss-2021a.eb                                ALREADY INSTALLED
UDUNITS-2.2.28-GCCcore-10.3.0.eb                            ALREADY INSTALLED
Elk-7.2.42-foss-2021a.eb                                    ALREADY INSTALLED
Crystal17-1.0.2-intel-2021a.eb                              ALREADY INSTALLED
VASP5-5.4.4.pl2-foss-2021a-CUDA-11.3.1.eb                   ALREADY INSTALLED
VASP5-5.4.4-intel-2021a-CUDA-11.3.1.eb                      ALREADY INSTALLED
VASP6-6.2.1-foss-2021a-CUDA-11.3.1.eb                       ALREADY INSTALLED
VASP6-6.2.1-intel-2021a-CUDA-11.3.1.eb                      ALREADY INSTALLED
CubeWriter-4.6-GCCcore-10.3.0.eb                            ALREADY INSTALLED
CubeLib-4.6-GCCcore-10.3.0.eb                               ALREADY INSTALLED
CubeGUI-4.6-GCCcore-10.3.0.eb                               ALREADY INSTALLED
gperftools-2.9.1-GCCcore-10.3.0.eb                          ALREADY INSTALLED
HPL-2.3-foss-2021a.eb                                       ALREADY INSTALLED
HPL-2.3-intel-2021a.eb                                      ALREADY INSTALLED
OTF2-2.3-GCCcore-10.3.0.eb                                  ALREADY INSTALLED
Paraver-4.9.2-foss-2021a.eb                                 ALREADY INSTALLED
HarfBuzz-2.8.1-GCCcore-10.3.0.eb                            ALREADY INSTALLED
Pango-1.48.5-GCCcore-10.3.0.eb                              ALREADY INSTALLED
ATK-2.36.0-GCCcore-10.3.0.eb                                ALREADY INSTALLED
GTK+-2.24.33-GCCcore-10.3.0.eb                              ALREADY INSTALLED
Gdk-Pixbuf-2.42.6-GCCcore-10.3.0.eb                         ALREADY INSTALLED
GTS-0.7.6-GCCcore-10.3.0.eb                                 ALREADY INSTALLED
Graphviz-2.47.2-GCCcore-10.3.0.eb                           ALREADY INSTALLED
libgd-2.3.1-GCCcore-10.3.0.eb                               ALREADY INSTALLED
SWIG-4.0.2-GCCcore-10.3.0.eb                                ALREADY INSTALLED
FFmpeg-4.3.2-GCCcore-10.3.0.eb                              ALREADY INSTALLED
libGLU-9.0.1-GCCcore-10.3.0.eb                              ALREADY INSTALLED
gnuplot-5.4.2-GCCcore-10.3.0.eb                             ALREADY INSTALLED
matplotlib-3.4.2-foss-2021a.eb                              ALREADY INSTALLED
VTK-9.0.1-foss-2021a.eb                                     ALREADY INSTALLED
wxWidgets-3.1.5-GCC-10.3.0.eb                               ALREADY INSTALLED
Blender-2.93.1-lts.eb                                       ALREADY INSTALLED
ParaView-5.9.1-foss-2021a-mpi.eb                            ALREADY INSTALLED
TurboVNC-2.2.6-GCCcore-10.3.0.eb                            ALREADY INSTALLED
VirtualGL-2.6.5-GCCcore-10.3.0.eb                           ALREADY INSTALLED
OpenCV-4.5.3-foss-2021a-CUDA-11.3.1-contrib.eb              ALREADY INSTALLED
ASAP-2.0-foss-2021a-CUDA-11.3.1.eb                          ALREADY INSTALLED
remotevis-git.eb                                            ALREADY INSTALLED
ncview-2.1.8-gompi-2021a.eb                                 ALREADY INSTALLED
WRF-4.3-foss-2021a-dmpar.eb                                 ALREADY INSTALLED
WPS-4.3.1-foss-2021a-dmpar.eb                               ALREADY INSTALLED
ESMF-8.1.1-foss-2021a.eb                                    ALREADY INSTALLED
TensorFlow-2.6.0-foss-2021a-CUDA-11.3.1.eb                  ALREADY INSTALLED
Horovod-0.22.1-foss-2021a-CUDA-11.3.1-TensorFlow-2.6.0.eb   ALREADY INSTALLED
PyTorch-1.10.0-foss-2021a-CUDA-11.3.1.eb                    ALREADY INSTALLED
torchvision-0.11.1-foss-2021a-CUDA-11.3.1.eb                ALREADY INSTALLED
Horovod-0.23.0-foss-2021a-PyTorch-1.10.0.eb                 ALREADY INSTALLED
IPython-7.25.0-GCCcore-10.3.0.eb                            ALREADY INSTALLED
JupyterHub-1.4.1-GCCcore-10.3.0.eb                          ALREADY INSTALLED
jupyter-server-proxy-3.1.0-GCCcore-10.3.0.eb                ALREADY INSTALLED
jupyterlmod-2.0.2-GCCcore-10.3.0.eb                         ALREADY INSTALLED
IRkernel-1.2-foss-2021a.eb                                  ALREADY INSTALLED
supporting-utils.eb                                         SUCCESS
==========
Built succesfully: 1/215
Already installed: 213/215
Failed: 1/215
Skipped: 0/215
Missing easyconfigs: 0/215
==========
srun: error: gcn20: task 0: Exited with exit code 1
srun: launch/slurm: _step_signal: Terminating StepId=304053.0
Return status: 1
script returned exit code 1
```

EasyBuild

- **Old workflow:**
  - download easyconfig from the public repo
  - modify
  - test installation locally
  - push to the local repo
  - execute a jenkins pipeline

EasyBuild

- A lot of site-specific easyconfigs and easyblocks in the local repo, almost no use of hooks
- Often only minor differences with the public repo (=> duplicated scripts)
- => easy to lost track of what version to use next time (public or local)
- Hardcoded “hot-fixes”
- Only local tests on locally available machine(s)

EasyBuild

- **Current workflow:**
  - modify/write easyconfig or easyblock
  - test locally
  - create PR to the public repo
  - wait for acceptance / resolve reviews
  - use “--from-pr" in the build list

EasyBuild

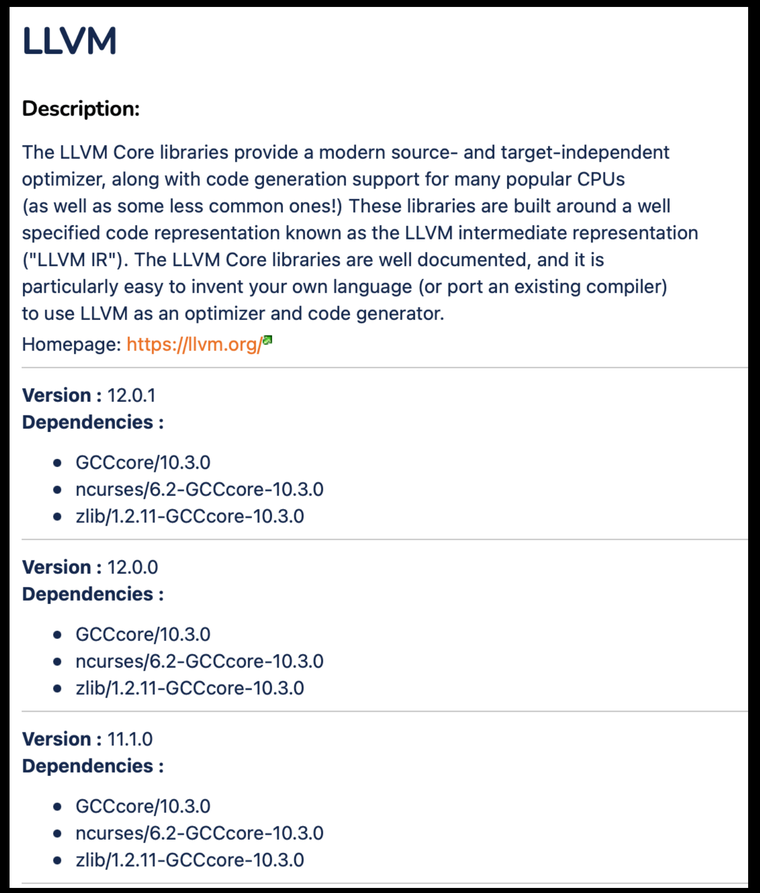
- If site-specific modifications needed (e.g. “modextravars”) - write a hook
- Feedbacks and checks from the community
- Tests on different systems
- No duplicated scripts

EasyBuild

- Users can perform local installation using the wrapper-script (“eblocalinstall”)
  - $HOME/.local/easybuild/…
  - Generic installation (i.e. optimised for the least performant architecture)
  - The wrapper script supports all keys from the “eb” command (--from-pr, --include-easyblock, etc.)
  - Users can also use site-specific easyconfigs and easyblocks and some hooks

Docs

- A simple Python script (~340 LoC)
- Atlassian python API (wrapper over REST API)
- The script parses modulefiles by category and generates/updates the wiki page
- Wiki lists description from the modulefile, all available versions of the software and all corresponding dependencies



LLVM

Description:

The LLVM Core libraries provide a modern source- and target-independent optimizer, along with code generation support for many popular CPUs (as well as some less common ones!) These libraries are built around a well specified code representation known as the LLVM intermediate representation ("LLVM IR"). The LLVM Core libraries are well documented, and it is particularly easy to invent your own language (or port an existing compiler) to use LLVM as an optimizer and code generator.

Homepage: https://llvm.org/

Version : 12.0.1
Dependencies :

- GCCcore/10.3.0
- ncurses/6.2-GCCcore-10.3.0
- zlib/1.2.11-GCCcore-10.3.0

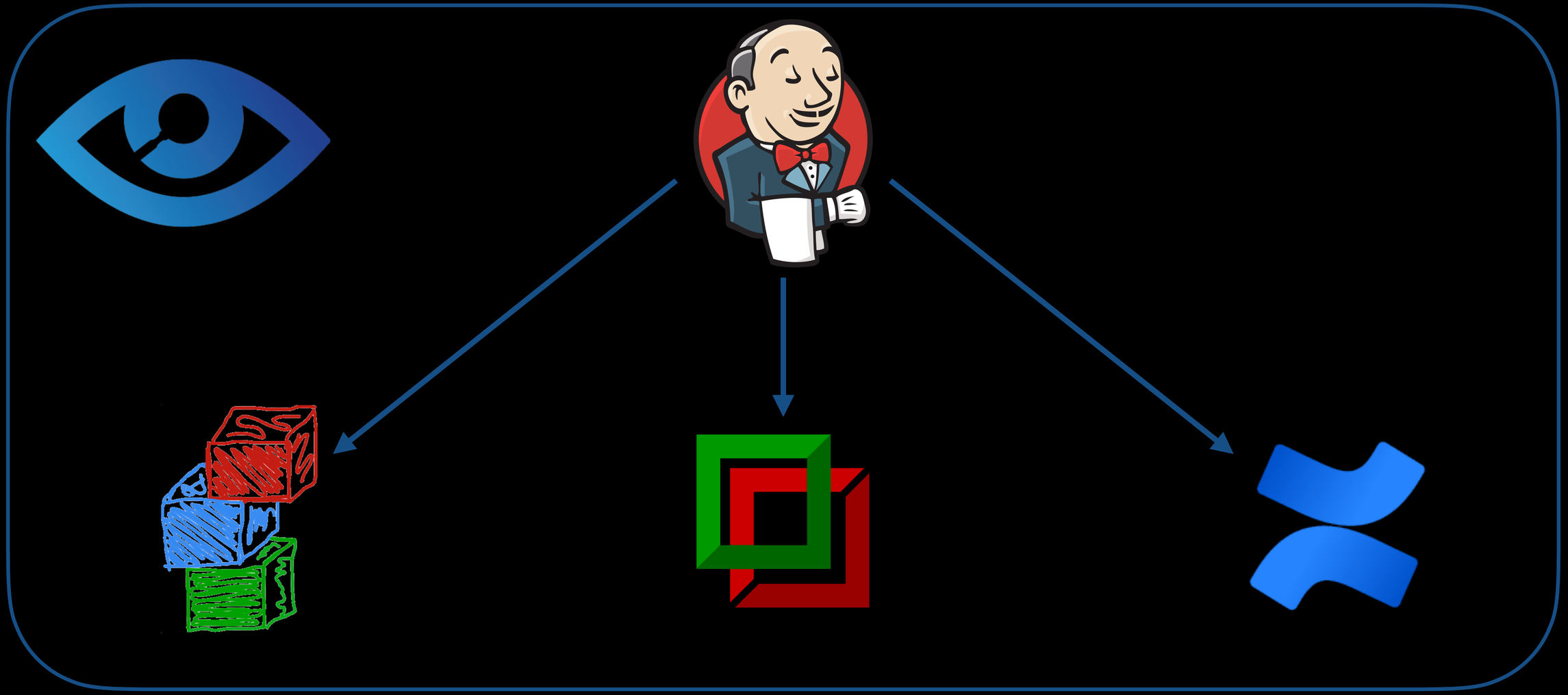
Version : 12.0.0
Dependencies :

- GCCcore/10.3.0
- ncurses/6.2-GCCcore-10.3.0
- zlib/1.2.11-GCCcore-10.3.0

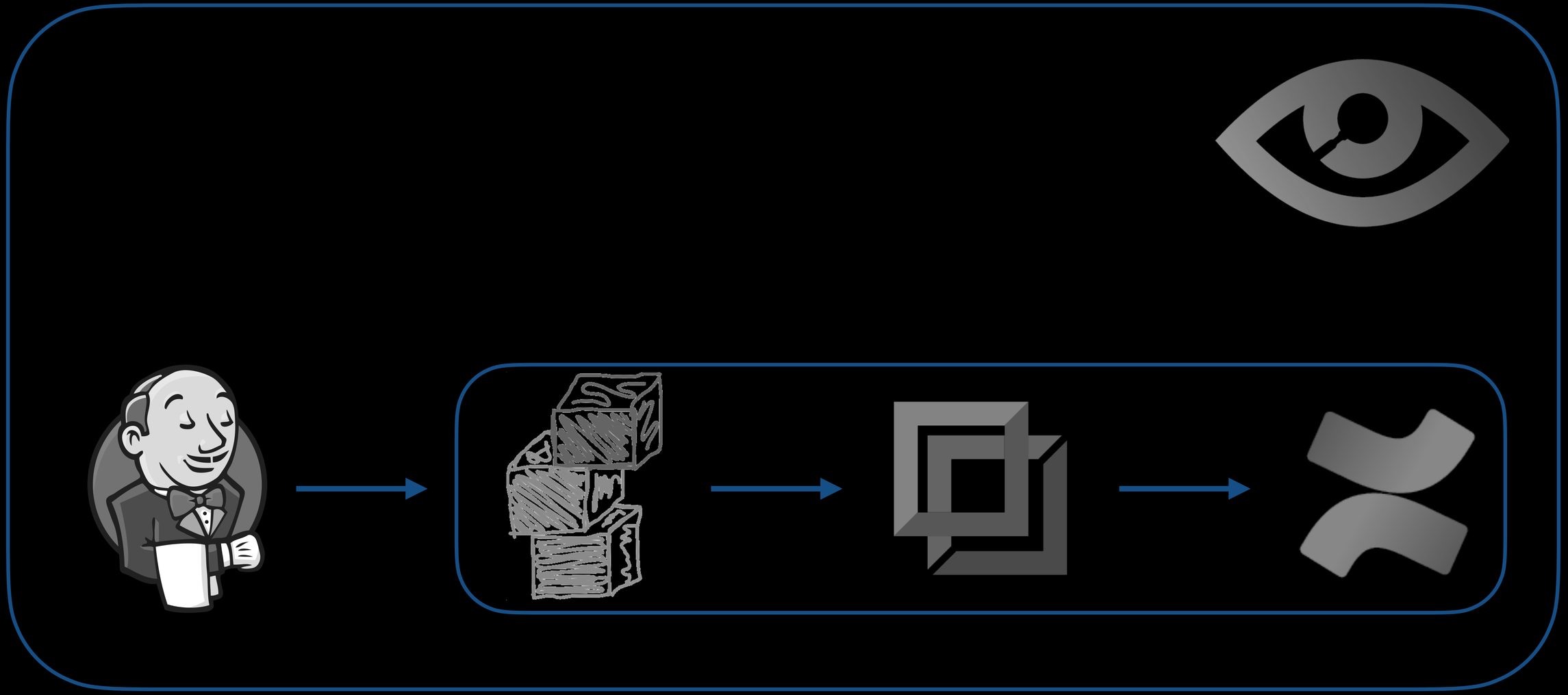
Version : 11.1.0
Dependencies :

- GCCcore/10.3.0
- ncurses/6.2-GCCcore-10.3.0
- zlib/1.2.11-GCCcore-10.3.0

Current workflow

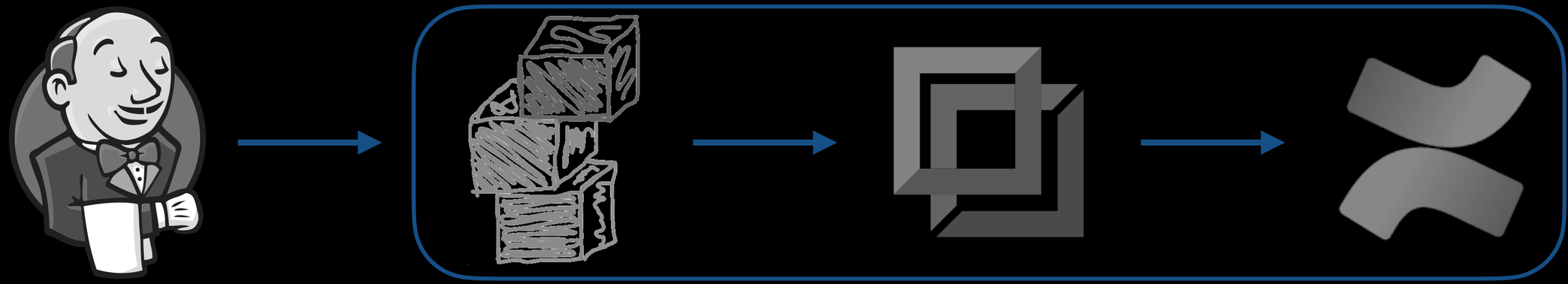


Workflow we aim for



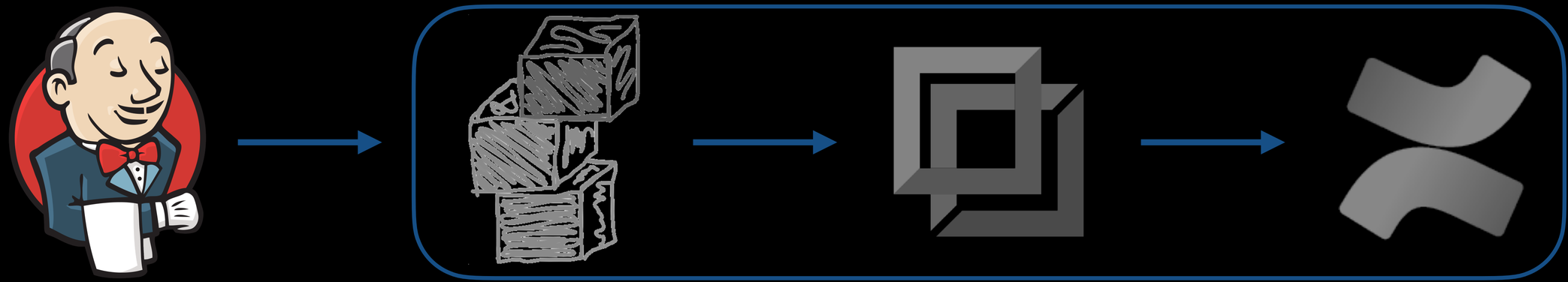
Workflow we aim for

- Analyse software usage with Xalt



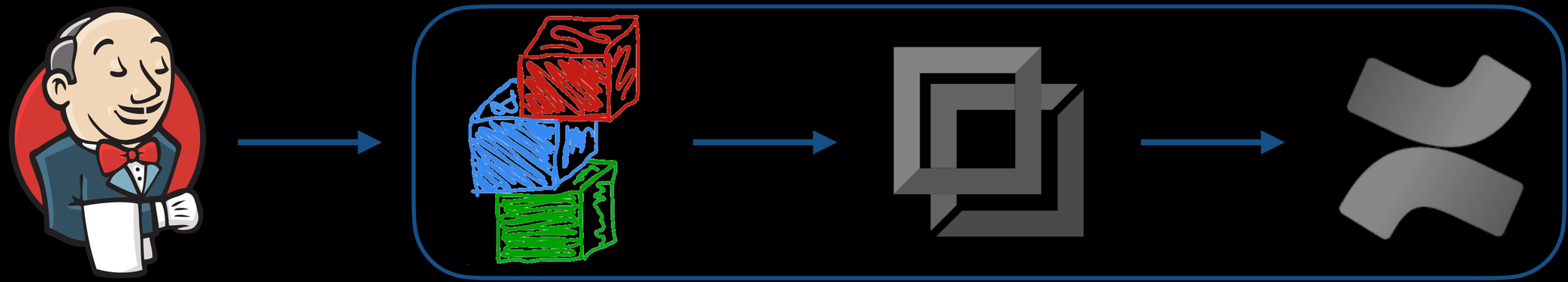
Workflow we aim for

- Analyse software usage with Xalt
- Automate all processes with Jenkins



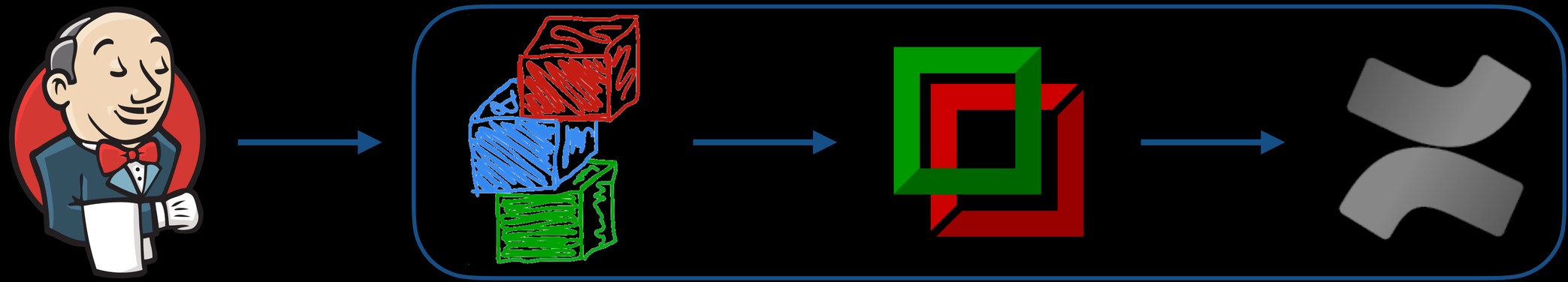
Workflow we aim for

- Analyse software usage with Xalt
- Automate all processes with Jenkins
- Modify/create easyconfigs/easyblocks, PR



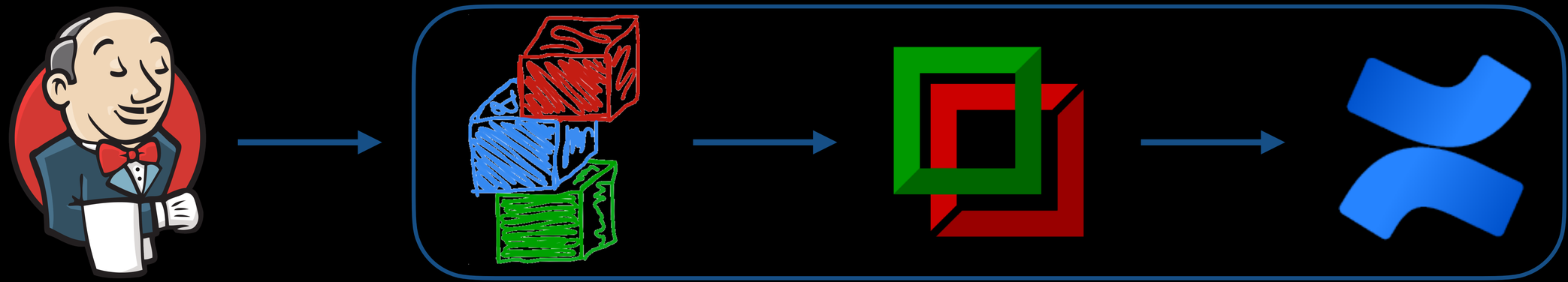
Workflow we aim for

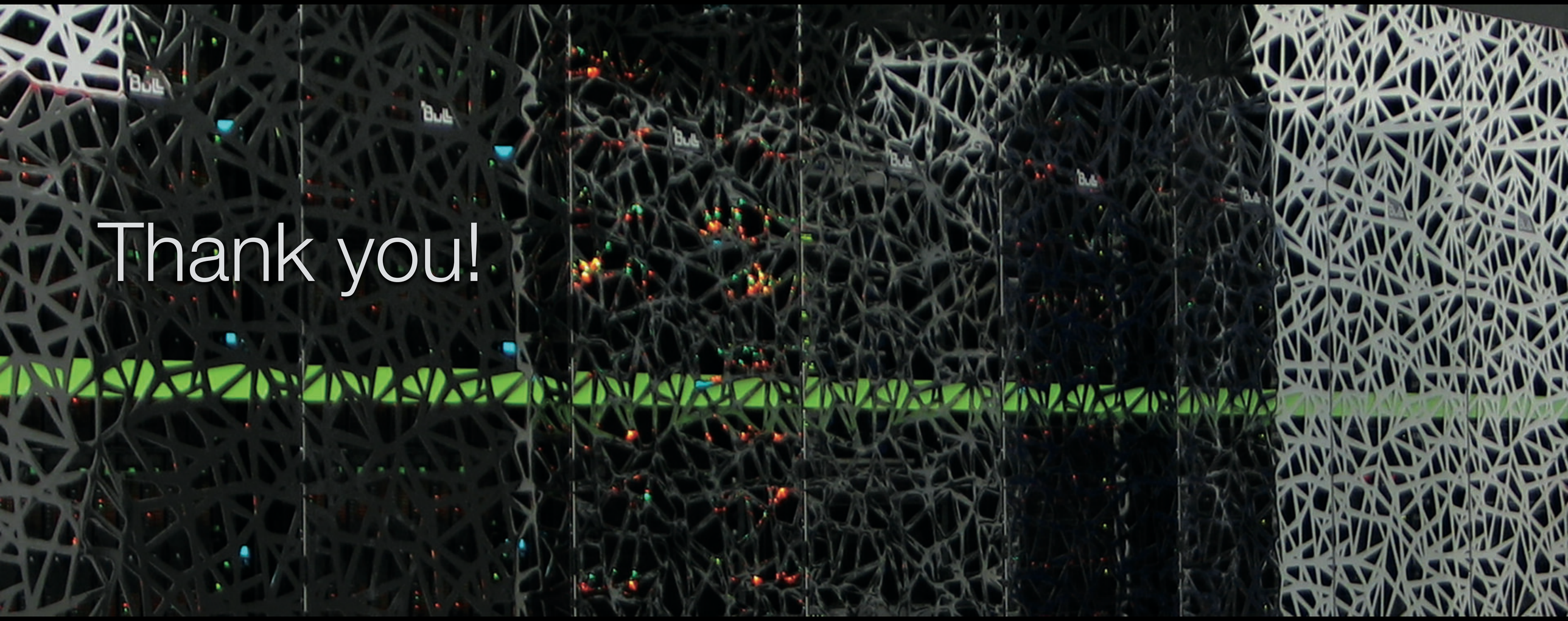
- Analyse software usage with Xalt
- Automate all processes with Jenkins
- Modify/create easyconfigs/easyblocks, PR
- Run regression tests



Workflow we aim for

- Analyse software usage with Xalt
- Automate all processes with Jenkins
- Modify/create easyconfigs/easyblocks, PR
- Run regression tests
- Create documentation




Thank you!