



How to use EasyBuild on Dardel

<https://docs.easybuild.io/en/latest/>

Methods of installing software at PDC

- EasyBuild

https://www.pdc.kth.se/support/documents/software_development/easybuild.html

- Spack

https://www.pdc.kth.se/support/documents/software_development/spack.html

- Manually

https://www.pdc.kth.se/support/documents/software_development/development_darwin.html

What is EasyBuild

- Started in 2008 by HPC team at Ghent university, Belgium
- Software build and installation framework
- Tailored towards High Performance Computing (HPC) systems
- Installs HPC software using procedures described in recipes

What modules are available

For local installations

```
ml PDC  
ml EasyBuild-user
```

- `INSTALLPATH: $HOME/.local/easybuild`
- `SOURCEPATH: $HOME/.local/easybuild/sources`
- `adds to MODULEPATH: $HOME/.local/easybuild/modules/all`
- `Temporary folder: /tmp/[user]`

How is EasyBuild configured

```
$ eb --show-config
# (C: command line argument, D: default value, E: environment variable, F: configuration file)
#
buildpath          (E) = /tmp/eb-build
...
sourcepath         (D) = /cfs/klemming/home/h/hzazzi/.local/easybuild/sources
```

- Current EasyBuild configuration can be changed if needed

```
$ eb --show-config --<CONFIG-NAME>=<VALUE>
```

Do it yourself: Exercise 1

- Load the EasyBuild-user module
- Check how it is configured
- Change some of the configuration parameters

What are easyconfigs files

- Plain text file to define installation parameters
- serves as a build specification for software installation
- Is typically named:

```
<name>-<version>[-<toolchain>][<versionsuffix>].eb
```

- Example

```
GROMACS-2021.3-cpeGNU-21.11.eb
```

What is a toolchain

- Defines what compiler toolchains to install the software with
- On Cray we have compiler wrappers, so all software is installed within the Cray Programming Environment (CPE)

```
cpeGNU version 21.11  
cpeCray version 21.11  
cpeAMD version 21.11
```

- Softwares not in need of parallelization can use the **SYSTEM** toolchain

How to install software using EasyBuild

- Installs dependencies
- Builds and install software
- Create modules for software and dependencies

```
$ eb <FILENAME>.eb
== creating build dir, resetting environment...
== unpacking...
== preparing...
== configuring...
== building...
== testing...
== installing...
== sanity checking...
== cleaning up...
== creating module...
== COMPLETED: Installation ended successfully
```

Other types of installation procedures

Installing software via the Robot paths

- Installs the latest software found in the paths defined in the configuration

```
eb --software-name=GROMACS --toolchain=cpeGNU,21.11
```

Rebuild

- rebuilds the software even if it does exist

```
eb --rebuild ...
```

dry-run

```
eb Boost-1.75.0-cpeGNU-21.09.eb -dr/--dry-run
```

- Test the installation procedure without installing it
- you can also use *-x/--extended-dry-run* for more information

How install dependent software

Automatically installs dependency software using easyconfigs that are available in robot paths

- Automatically install dependencies

```
eb Boost-1.75.0-cpeGNU-21.09.eb -r/--robot [= <PATH>]
```

- Check what dependencies are missing

```
eb Boost-1.75.0-cpeGNU-21.09.eb -M/--missing
```

Supported Robot paths

Contains easyconfigs and easyblocks for building software.

- PDC SoftwareStack
- LUMI SoftwareStack
- CSCS Software stack

How to search for software using EasyBuild

```
eb -S/--search <software>
CFG1=/pdc/software/eb_repo/PDC-SoftwareStack/easybuild/easyconfigs/g
CFG2=/pdc/software/eb_repo/LUMI-SoftwareStack/easybuild/easyconfigs/g/GROMACS
CFG3=/pdc/software/eb_repo/CSCS-production/easybuild/easyconfigs/g/GROMACS
* $CFG1/GROMACS-2020.5-cpeCray-21.09.eb
* $CFG1/GROMACS-2021.3-cpeCray-21.09.eb
* $CFG1/GROMACS-2022-beta1-cpeCray-21.09.eb
```

- Lists available *easyconfig* files
- These can be used to make new easyconfig recipes
- PDC recipes can be found at <https://github.com/PDC-support/PDC-SoftwareStack/blob/master/easybuild/easyconfigs>

Copy found easyconfigs

Copy found easyconfigs to act as base for creating easyconfigs for your application

```
eb --copy-ec <easyconfig filename>.eb [<new filename>]
```

Example:

```
eb --copy-ec BWA-0.7.17.eb my_easyconfig.eb
```

Do it yourself: Exercise 2

- Install BWA
 - Does it miss any dependencies
 - Run a dry-run to see if it installs properly before installing it
- Install any other software using EasyBuild



How to build easyconfig files

- Parameters and templates
- What is needed in an easyconfig file
 - Name
 - Toolchain
 - Sources
 - Easyblock
 - Dependencies
 - Sanity_check

Writing easyconfig files https://docs.easybuild.io/en/latest/Writing_easyconfig_files.html

Parameters and templates in easyconfig files

A full overview of all known easyconfig parameter

```
eb -a/--avail-easyconfig-params
```

A set of variables that can be used in easyconfig files

```
eb --avail-easyconfig-templates
```

Name

- Specifies the name and version of the software
- module will be named accordingly
- *versionsuffix* is not mandatory

```
name = 'GROMACS'  
version = '2020.5'  
versionsuffix = '-PLUMED-2.7.2'  
  
homepage = 'https://blast.ncbi.nlm.nih.gov/'  
description = ""Blast for searching sequences""
```

Toolchain

If you want to use MPI, OpenMP ...

```
toolchain = {'name': 'cpeGNU', 'version': '21.11'}
```

Will also have an impact on the dependencies for this easyconfig

If you want to use supporting tools, libraries...

```
toolchain = SYSTEM
```

Sources

Specify where you can download your source

```
sources = [{  
    'source_urls': ['https://example.com'],  
    'filename': '%(name)s-%(version)s.tar.gz',  
    'extract_cmd': "tar xf %s", # Optional  
}]
```

```
source_urls = ['ftp://ftp.%(namelower)s.org/pub/%(namelower)s/']  
sources = ['%(namelower)s_%(version_major)s_%(version_minor)s_0.tar.bz2']
```

More information at

https://docs.easybuild.io/en/latest/Writing_easyconfig_files.html#source-files-patches-and-checksums

Easyblock

- A python code to address special needs of the installation
- Addresses that you should first run *configure* > *make* > *make install* or **cmake* > *make* > *make install*

```
easyblock = 'type'
```

- Many EasyBlock are generic as to describe standard installation patterns
- Easyconfigs without an easyblock entry are special and Easybuild will search for EasyBlocks named **EB_[software]**

To find which Easyblock is specially for you...

```
eb --list-easyblocks
```

Examples of useful easyblocks

- **ConfigureMake**: implements the standard `./configure`, `make`, `make install` installation procedure;
- **CMakeMake**: same as **ConfigureMake**, but with `./configure` replaced with `cmake` for the configuration step;
- **PythonPackage**: implements the installation procedure for a single Python package, by default using "python [setup.py](#) install" but other methods like using "pip install" are also supported;
- **Bundle**: a simple generic easyblock to bundle a set of software packages together in a single installation directory;

See information about parameters for easyblocks

https://docs.easybuild.io/en/latest/version-specific/generic_easyblocks.html



Dependencies/builddependencies

- Will be installed if found and a module does not exists.
- *dependencies* are used when the module is loaded
- *builddependencies* are used when the software is installed

Main application toolchain

```
dependencies = [  
    ('Software', 'version'),  
]
```

System toolchain

```
dependencies = [  
    ('Software', 'version', '', ('system', '')),  
]
```


Sanity check

- A test to see everything was installed correctly

```
sanity_check_paths = {  
    'files': ['bin/reframe',  
              'share/completions/file1',  
              'share/completions/file2'],  
    'dirs': ['bin', 'lib', 'share', 'tutorials']  
}  
  
sanity_check_commands = [  
    'software --version',  
    'software --help',  
]
```

Do it yourself: Exercise 3

- Create your own *easyconfig* file
 - Create your *easyconfig* on any recipe you find that is appropriate
 - Edit and make the necessary changes
- Perform a dry-run as to ascertain that there are no installation issues
- Install the software