

Building Reproducible HPC Software Stacks with EasyBuild

High-Performance Computing (HPC) environments often involve a complex web of libraries, compilers, and dependencies. For users, compiling scientific software manually can quickly turn into a frustrating maze of missing headers, mismatched MPI versions, or incompatible compilers. This is where **EasyBuild** steps in - a tool designed to *automate, document, and reproduce* software builds in HPC environments.

What is EasyBuild?

EasyBuild is a framework that builds and installs scientific software and their dependencies in a consistent and reproducible way. It uses pre-defined "easyconfigs" (configuration files) that describe how to build a package - including compiler versions, dependencies, and environment modules. This means that once a software is built with EasyBuild, anyone can reproduce the exact same setup simply by using the same configuration file.

Why Users Should use EasyBuild

While system administrators often deploy software globally, users can also benefit by using EasyBuild within their **home directory**. This allows them to:

- Build custom versions of libraries or compilers not available system-wide.
- Reproduce and share software setups with collaborators.
- Avoid dependency conflicts between different research projects.

Practical Guide: Installing a Package with EasyBuild

1. Load Easybuild module

```
## Load easybuild module to use

[testuser2@loginnode1 ~] $ module load easybuild

// Activating the Lmod Environment

Loading easybuild

Loading requirement: python/3.8.5 lua luarocks lmod
```

2. Search for available software recipes

```
## We will search FFTW package from easybuild repository

[testuser2@loginnode1 ~] $ eb --search fftw
```

```
testuser2@loginnode1:~$ eb --search fftw
== found valid index for /common_library/easybuild/easyconfigs, so using it...
* /common_library/easybuild/easybuild/easyconfigs/e/Speak-NG/eSpeak-NG-1.50_sonic_fftw.patch
* /common_library/easybuild/easybuild/easyconfigs/f/FFTW.MPI/FFTW.MPI-3.3.10-gmpich-2024.06.eb
* /common_library/easybuild/easybuild/easyconfigs/f/FFTW.MPI/FFTW.MPI-3.3.10-gompi-2022.05.eb
* /common_library/easybuild/easybuild/easyconfigs/f/FFTW.MPI/FFTW.MPI-3.3.10-gompi-2022.10.eb
* /common_library/easybuild/easybuild/easyconfigs/f/FFTW.MPI/FFTW.MPI-3.3.10-gompi-2022a.eb
* /common_library/easybuild/easybuild/easyconfigs/f/FFTW.MPI/FFTW.MPI-3.3.10-gompi-2022b.eb
* /common_library/easybuild/easybuild/easyconfigs/f/FFTW.MPI/FFTW.MPI-3.3.10-gompi-2023.09.eb
* /common_library/easybuild/easybuild/easyconfigs/f/FFTW.MPI/FFTW.MPI-3.3.10-gompi-2023a.eb
* /common_library/easybuild/easybuild/easyconfigs/f/FFTW.MPI/FFTW.MPI-3.3.10-gompi-2023b.eb
* /common_library/easybuild/easybuild/easyconfigs/f/FFTW.MPI/FFTW.MPI-3.3.10-gompi-2024.05.eb
* /common_library/easybuild/easybuild/easyconfigs/f/FFTW.MPI/FFTW.MPI-3.3.10-gompi-2024a.eb
* /common_library/easybuild/easybuild/easyconfigs/f/FFTW.MPI/FFTW.MPI-3.3.10-gompi-2025a.eb
* /common_library/easybuild/easybuild/easyconfigs/f/FFTW.MPI/FFTW.MPI-3.3.10-gompi-2025b.eb
* /common_library/easybuild/easybuild/easyconfigs/f/FFTW.MPI/FFTW.MPI-3.3.10-imp-2023a.eb
* /common_library/easybuild/easybuild/easyconfigs/f/FFTW.MPI/FFTW.MPI-3.3.10-nvcomp-2022.07.eb
* /common_library/easybuild/easybuild/easyconfigs/f/FFTW/FFTW-3.3.9-gompi-2021a.eb
* /common_library/easybuild/easybuild/easyconfigs/f/FFTW/FFTW-3.3.9-intel-2021a.eb
* /common_library/easybuild/easybuild/easyconfigs/f/FFTW/FFTW-3.3.10-GCC-11.3.0.eb
* /common_library/easybuild/easybuild/easyconfigs/f/FFTW/FFTW-3.3.10-GCC-12.2.0.eb
* /common_library/easybuild/easybuild/easyconfigs/f/FFTW/FFTW-3.3.10-GCC-12.3.0.eb
* /common_library/easybuild/easybuild/easyconfigs/f/FFTW/FFTW-3.3.10-GCC-13.2.0.eb
* /common_library/easybuild/easybuild/easyconfigs/f/FFTW/FFTW-3.3.10-GCC-13.3.0.eb
* /common_library/easybuild/easybuild/easyconfigs/f/FFTW/FFTW-3.3.10-GCC-14.2.0.eb
* /common_library/easybuild/easybuild/easyconfigs/f/FFTW/FFTW-3.3.10-GCC-14.3.0.eb
* /common_library/easybuild/easybuild/easyconfigs/f/FFTW/FFTW-3.3.10-NVHPC-22.7-CUDA-11.7.0.eb
```

3. Build and install FFTW using a predefined config

```
## Lets install FFTW-3.3.10 using GCC-11.3.0

[testuser2@loginnode1 ~] $ eb FFTW-3.3.10-GCC-11.3.0.eb --robot
```

```
testuser2@loginnode1:~$ eb FFTW-3.3.10-GCC-11.3.0.eb --robot
== Temporary log file in case of crash /tmp/eb-dco3qx3b/easybuild-gbty00ac.log
== found valid index for /common_library/easybuild/easyconfigs, so using it...
== found valid index for /common_library/easybuild/easybuild/easyconfigs, so using it...
== resolving dependencies ...
== processing EasyBuild easyconfig /common_library/easybuild/easybuild/easyconfigs/m/M4/M4-1.4.19.eb
== building and installing M4/1.4.19...
> installation prefix: /home/testuser2/easybuild/software/M4/1.4.19
== fetching files and verifying checksums...
>> download succeeded: https://ftpmirror.gnu.org/gnu/m4/m4-1.4.19.tar.gz

== Build succeeded for 13 out of 13
== Summary:
* [SUCCESS] Bison/3.8.2
* [SUCCESS] flex/2.6.4
* [SUCCESS] zlib/1.2.12
* [SUCCESS] binutils/2.38
* [SUCCESS] GCCcore/11.3.0
* [SUCCESS] zlib/1.2.12-GCCcore-11.3.0
* [SUCCESS] help2man/1.49.2-GCCcore-11.3.0
* [SUCCESS] M4/1.4.19-GCCcore-11.3.0
* [SUCCESS] Bison/3.8.2-GCCcore-11.3.0
* [SUCCESS] flex/2.6.4-GCCcore-11.3.0
* [SUCCESS] binutils/2.38-GCCcore-11.3.0
* [SUCCESS] GCC/11.3.0
* [SUCCESS] FFTW/3.3.10-GCC-11.3.0
```

4. Load the installed module

To Load FFTW module, first load easybuild module and than load fftw module.

```
[testuser2@loginnode1 ~] $ module load easybuild
[testuser2@loginnode1 ~] $ module avail FFTW
```

----- /home/testuser2/easybuild/modules/all -----

FFTW/3.3.10-GCC-11.3.0

```
[testuser2@loginnode1 ~] $ module load FFTW/3.3.10-GCC-11.3.0
```

```
[testuser2@loginnode1 ~] $ module list
```

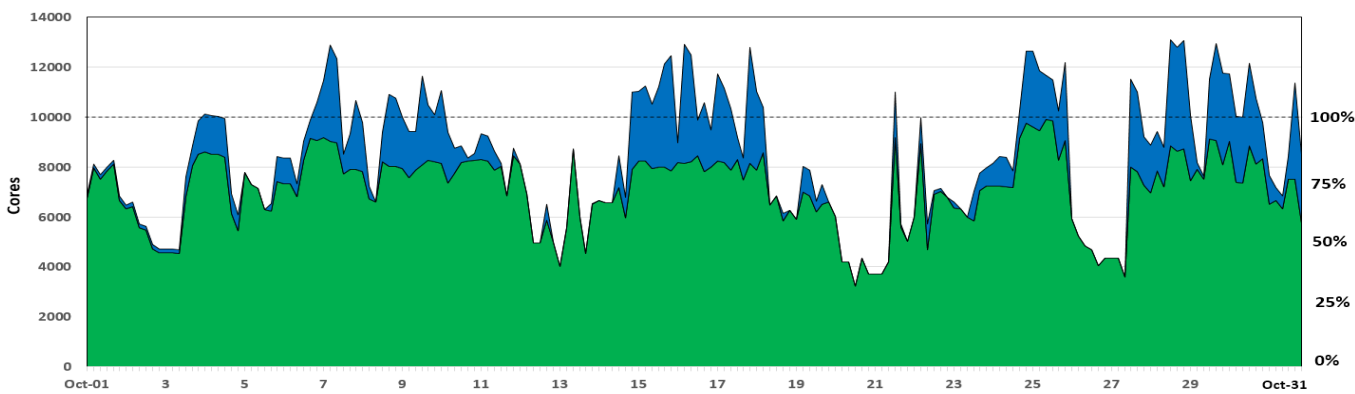
Currently Loaded Modules:

1) GCCcore/11.3.0 2) zlib/1.2.12-GCCcore-11.3.0 3) binutils/2.38-GCCcore-11.3.0 4) GCC/11.3.0 5) FFTW/3.3.10-GCC-11.3.0

With EasyBuild, HPC users can take control of their software environments — building exactly what they need, when they need it, and doing so reproducibly. Integrating EasyBuild with user's workflow can make users' HPC experience smoother, more predictable, and far less error-prone.

ANTYA Utilization: OCTOBER 2025

ANTYA DAILY OBSERVED WORKLOAD



ANTYA HPC Users' Statistics
October 2025

Total Successful Jobs~ 1503

> Top Users (Cumulative Resources)

- CPU Cores [Sagar Choudhary](#)
- Walltime [Janmejay U Buch](#)
- Jobs [Jugal Chowdhury](#)

ANTYA Usage, Updates and News

- **Scheduled Downtime:** There was no downtime of ANTYA for October 2025.
- **Job Submissions:** The highest job loads were observed in the mediumq, serialq, regular, longq and ansysq queues, reflecting sustained user activity across multiple workloads in various queues.
- **Cluster Utilization:** The system maintained an average utilization of ~70.95% and peak utilisation of ~98%.

Packages/Applications Installed: No new modules have been installed this month. To view list of available modules.
> module avail

Other Recent Work on HPC

Degradation mechanism of Monocrotophos by hydroxyl radicals: Reactive molecular Dynamics and DFT Insights	Dr. Ruchi Mishra
Design and CFD driven thermal Performance Analysis of the Primary Chamber of Plasma Pyrolysis Plant for Biomedical Waste disposal: From 50 kg/h experimental Validation to 200 kg/h Scale-Up	Deepak Sharma
Design, Fabrication, Testing and Qualification of Gas Cooled Finger type Mock-up	Alpeshbhai Kantibhai Patel
Performance Evaluation of Tungsten Gas-Cooled Divertor Finger-Type Mock-up	Rajamannar Swamy Kidambi
Role of Turbulence and Coherent Structure in 3-Dimensional MHD Dynamos	Dinesh Kumar S
Electromagnetic Gyrokinetic Simulation of Turbulence in ADITYA-U with Collisions	Amit Kumar Singh
Space Charge Modeling for Negative Ion Beam Transport: A PIC Study	Sidharth Kumar Dash

Acknowledgement

The HPC Team, Computer Division IPR, would like to thank all Contributors for the current issue of *GANANAM*.

Join the HPC Users Community

hpcusers@ipr.res.in

If you wish to contribute an article in *GANANAM*, please write to us.



Contact us

HPC Team (hpcteam@ipr.res.in)
Computer Division, IPR

Disclaimer: “*GANANAM*” is IPR’s informal HPC Newsletter to disseminate technical HPC related work performed at IPR from time to time. Responsibility for the correctness of the Scientific Contents including the statements and cited resources lies solely with the Contributors.