



University of Colorado **Anschutz Medical Campus**

# Alpine module stack and miniforge

*By Kevin Fotso  
09/15/2025*



# Introduction:

This workshop will cover the following topics:

- What is LMOD?
- How does LMOD work?
- Basics Alpine module stack commands.
- Best practices with LMOD.



University of Colorado  
Anschutz Medical Campus

# Introduction:

This workshop will cover the following topics:

- Miniforge on Alpine.
- How to set up Miniforge?
- Best practices with Miniforge.



University of Colorado  
Anschutz Medical Campus

# Audience:

Users who want to run software on Alpine



University of Colorado **Anschutz Medical Campus**

# Part 1 - LMOD

# LMOD (CONTEXT)

- Modern environment module system for HPC
- Initially introduced by Robert McLay from TACC (UT Austin) in 2011.



University of Colorado  
Anschutz Medical Campus

# LMOD (CONTEXT)

- Users on HPC have different needs.
- Applications, compilers, libraries, versions being used might be different.



# LMOD (benefits)

- Users do not need to know where software is installed.
- Environment variable to interface packages can be set (e.g. picard).
- Very useful for software reproducibility.



# LMOD (benefits)

- Supports software hierarchy.
- Users can set their environment modules at will.



University of Colorado  
Anschutz Medical Campus

# LMOD (Commands)

- Users call the **module** API to interact with LMOD and its module path hierarchy.
- HPC staff or users employ the **lua** programming language to make modules.



University of Colorado  
Anschutz Medical Campus

# References:



## Documentation

[manual](#) · [readme](#) · [books](#) · [papers](#) · [faq](#) · [wiki](#) · [source code](#) · [versions](#) · [bugs](#) · [license](#) · [getting started](#)

### ❖ Reference manual

The official definition of the Lua language is its [reference manual](#), which describes the syntax and the semantics of Lua, the standard libraries, and the C API.

The [reference manuals](#) for all versions of Lua are [available online](#) in English. Some are available in other languages and as a printed book.

The current version is the [reference manual for Lua 5.4](#).

### ❖ Technical documentation

For more technical information, see the [Frequently Asked Questions \(FAQ\)](#) and some old [seminar slides](#). For detailed technical information on specific topics, see our old series of [Lua Technical Notes](#) and the [wiki](#) at [lua-users.org](#), especially the [tutorial](#). You may also browse the [source code](#).

### ❖ Books

Lua books are available at the main online stores and also as e-books at [Feisty Duck](#). When you buy a copy of a book published by Lua.org, you help to [support the Lua project](#).

#### Reference manual

The official definition of the Lua language:



#### [Lua 5.1 Reference Manual](#)

by R. Ierusalimschy, L. H. de Figueiredo, W. Celes,  
Lua.org, August 2006  
ISBN 8590379833

The reference manuals for later versions of Lua are available [online](#).

Source: <https://www.lua.org/docs.html>

# References:



## Table of Contents

Lmod: A New Environment Module System

Monthly Zoom Meeting

- PURPOSE
  - OVERVIEW
  - Lmod Web Sites
  - Introduction to Lmod
  - Installing Lmod
  - Advanced Topics
  - Overview and Internal Structure of Lmod
  - Topics yet to be written
- Indices and tables

## Next topic

User Guide for Lmod

## This Page

Show Source

## Quick search

Go

## Lmod: A New Environment Module System

### Monthly Zoom Meeting

**NOTE** Lmod is holding Monthly Zoom meeting to discuss various topics. Typically it is the first Tuesday of the Month at 9:30 U.S. Central (which is 14:30 UTC or 15:30 UTC in the winter months). Beginners are welcome. There is always a Q/A session at the beginning. Topic announcements are sent to the Lmod mailing list.

See: <https://github.com/TACC/Lmod/wiki> for details.

### PURPOSE

Lmod is a Lua based module system that easily handles the MODULEPATH Hierarchical problem. Environment Modules provide a convenient way to dynamically change the users' environment through modulefiles. This includes easily adding or removing directories to the PATH environment variable. Modulefiles for Library packages provide environment variables that specify where the library and header files can be found.

### OVERVIEW

This guide is written to explain what Environment Modules are and why they are very useful for both users and system administrators. Lmod is an implementation of Environment Modules, much of what is said here is true for any environment modules system but there are many features which are unique to Lmod.

Environment Modules provide a convenient way to dynamically change the users' environment through modulefiles. This includes easily adding or removing directories to the PATH environment variable.

# How to access Alpine

First let's login to Alpine!



1

Go to <https://ondemand-rmacc.rc.colorado.edu>

2

It should redirect you to CILogon which is how you authenticate your Alpine session. Make sure you select "ACCESS CI (XSEDE)" as your identity provider and then press "Log On"



The image shows a two-step login process. Step 1: A consent screen titled "Consent to Attribute Release" from "Open OnDemand" asking for permission to access user identifier, name, email, and affiliation. Step 2: A "Select an Identity Provider" screen where "ACCESS CI (XSEDE)" is selected from a dropdown menu. A blue arrow points from the "CILogon" logo to this second screen.

6



University of Colorado  
Anschutz Medical Campus

Source: [https://github.com/kf-cuanschutz/CU-Anschutz-HPC-documentation/blob/main/Workshops/Alpine\\_Noob\\_Introduction\\_to\\_HPC\\_and\\_Alpine-120723.pdf](https://github.com/kf-cuanschutz/CU-Anschutz-HPC-documentation/blob/main/Workshops/Alpine_Noob_Introduction_to_HPC_and_Alpine-120723.pdf)

# How to access Alpine

## First let's login to Alpine!



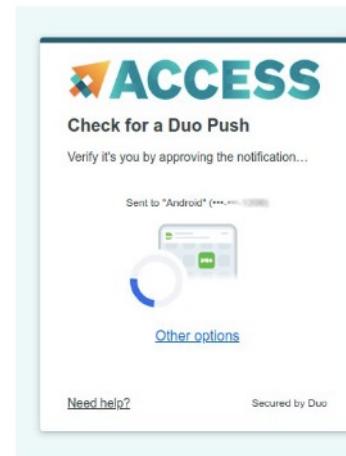
3

Next it will take you to this page where you will put in your ACCESS ID and ACCESS password and press Login. This is NOT your CU Anschutz ID!!

A screenshot of the ACCESS CILogon login page. The page has a light blue header with the ACCESS logo. Below the header, there is a form titled "Login to CILogon" with two input fields: "ACCESS Username" and "ACCESS Password". To the left of the form, there is a note: "ACCESS ID and ACCESS password not CU Anschutz credentials!!". Below the input fields is a "Login" button. To the right of the form, there is a "CILogon" logo with the text "CILogon facilitates secure access to CyberInfrastructure (CI)." and links for "Register for an ACCESS Account", "Forgot your password?", and "Need Help?".

4

This will prompt a DUO MFA push to your phone or whichever way you have DUO set up on your phone to authenticate. Accept the push sent to your device.



7



University of Colorado  
Anschutz Medical Campus

Source: [https://github.com/kf-cuanschutz/CU-Anschutz-HPC-documentation/blob/main/Workshops/Alpine\\_Noob\\_Introduction\\_to\\_HPC\\_and\\_Alpine-120723.pdf](https://github.com/kf-cuanschutz/CU-Anschutz-HPC-documentation/blob/main/Workshops/Alpine_Noob_Introduction_to_HPC_and_Alpine-120723.pdf)

# How to access Alpine

## First let's login to Alpine!



5

This will take to the official CURC page. Let's select the Alpine terminal

A screenshot of the Research Computing OnDemand interface. At the top, there is a navigation bar with links for "Files", "Jobs", "Clusters", "Interactive Apps", and "My Interactive Sessions". The "Interactive Apps" link is highlighted. A dropdown menu is open under "Interactive Apps", showing two options: "Alpine Shell" and "Blanca Shell". An orange arrow points from the number "5" in the top left corner to this dropdown menu. Below the navigation bar, the University of Colorado Boulder Research Computing logo is displayed. The main content area contains a message: "OnDemand provides an integrated, single access point for all of your HPC resources." followed by a "Message of the Day" section. The "Message of the Day" section includes a welcome message, "Welcome to the University of Colorado Research Computing.", and a "Quick Links" section with links to "CU Boulder RC Status", "Research Computing User Guide", "Research Computing at CU Boulder", "RMACC @ Ask.Cyberinfrastructure", and "Need help? Email (rc-help@colorado.edu)".

8



University of Colorado  
Anschutz Medical Campus

Source: [https://github.com/kf-cuanschutz/CU-Anschutz-HPC-documentation/blob/main/Workshops/Alpine\\_Noob\\_Introduction\\_to\\_HPC\\_and\\_Alpine-120723.pdf](https://github.com/kf-cuanschutz/CU-Anschutz-HPC-documentation/blob/main/Workshops/Alpine_Noob_Introduction_to_HPC_and_Alpine-120723.pdf)

# Make sure that you can access Alpine

## First let's login to Alpine!



6

This will log you into the head node of Alpine. It will always default you to your home directory.

```
Host: login-cil.rc.int.colorado.edu
Welcome to University of Colorado Boulder Research Computing!

Full documentation is available in our user guide at
https://www.rc.colorado.edu/support/user-guide. If you have a question
that's not answered there, contact us at rc-help@colorado.edu.

A number of directories have been created for you already:
* `/home/$USER`, your home directory
* `/projects/$USER`, your project directory

Run the command `module avail` to see a list of available software.

To prevent this README from being displayed at login, edit your
`.bash_profile` or `.login` files.

Welcome to CU-Boulder Research Computing.

* Website http://colorado.edu/rc
* Questions? rc-help@colorado.edu
* Subscribe to system announcements: https://curc.statuspage.io/
* Please type rc-help for the Acceptable Use Policy and a short help page.

You are using login node: login-cil

Users who had jobs in the queue prior to the planned maintenance should check
to confirm these jobs are still queued. Some jobs, particularly those scheduled
since midnight today (Wed June 7), may have been canceled during the
maintenance period.
@xsede.org@login-cil      @xsede.org]$
```

9



University of Colorado  
Anschutz Medical Campus

Source: [https://github.com/kf-cuanschutz/CU-Anschutz-HPC-documentation/blob/main/Workshops/Alpine\\_Noob\\_Introduction\\_to\\_HPC\\_and\\_Alpine-120723.pdf](https://github.com/kf-cuanschutz/CU-Anschutz-HPC-documentation/blob/main/Workshops/Alpine_Noob_Introduction_to_HPC_and_Alpine-120723.pdf)

# Filesystems

/home/\$USER

**2GB**

- Do not use this for anything regarding data transfer, storage, software installs, etc...
- One access pre-existing configuration files here if you need to modify them (.bashrc, .bash\_profile, .condrc, etc...)

/projects/\$USER

**250GB**

- Store all code, scripts, and sbatch files here
- Local software that you need to install yourself should go here
- If small enough, reference files that will be re-used, i.e. genome index, genome fasta or gtf annotation files, etc...

/scratch/alpine/\$USER

**10TB**

- Location for all data transfers and large data to be used to computation and analysis
- Output files generated by software should be redirected here



! There is a 90-day automated data purge from the date of file creation/transfer; not to be used for permanent storage or any data that is single copy

/pl/

**1TB-??**

- OPTIONAL – only available to those that are paying for an allocation
- Permanent data storage mounted to your Alpine user space
- There is a cost associated with it

# LMOD ON the login node

```
|  )@xsede.org@login-ci1 ~]$ module avail
```

```
----- /curc/sw/modules/slurm -----
StdEnv (L)  curc-quota/latest (L)  slurm/alpine (L,D)  slurm/blanca  slurm/core  slurmtools

----- /usr/share/lmod/lmod/modulefiles/Core -----
lmod      settarg
```

Where:

L: Module is loaded  
D: Default Module

- We are on the login node.



University of Colorado  
Anschutz Medical Campus

# LMOD ON the login node

```
|  )@xsede.org@login-ci1 ~]$ module avail
```

```
----- /curc/sw/modules/slurm -----
StdEnv (L)  curc-quota/latest (L)  slurm/alpine (L,D)  slurm/blanca  slurm/core  slurmtools

----- /usr/share/lmod/lmod/modulefiles/Core -----
lmod      settarg
```

Where:

L: Module is loaded  
D: Default Module



- We are located in the home directory (~) sign.



University of Colorado  
Anschutz Medical Campus

# LMOD ON the login node

```
|  )@xsede.org@login-ci1 ~]$ module avail
```

StdEnv (L)	curc-quota/latest (L)	slurm/alpine (L,D)	slurm/blanca	slurm/core	slurmtools
----- /curc/sw/modules/slurm -----					
lmod	settarg	----- /usr/share/lmod/lmod/modulefiles/Core -----			

Where:

L: Module is loaded  
D: Default Module



- “module avail” to check all the modules available on the login node.



University of Colorado  
Anschutz Medical Campus

# LMOD ON the login node

```
|  )@xsede.org@login-ci1 ~]$ module avail
```

```
----- /curc/sw/modules/slurm -----
StdEnv (L) curc-quota/latest (L) slurm/alpine (L,D) slurm/blanca slurm/core slurmtools
----- /usr/share/lmod/lmod/modulefiles/Core -----
lmod settarg
```

Where:

L: Module is loaded  
D: Default Module

- curc-quota is loaded (L)
- curc-quota prints how much space is available in the main filesystems that you have access to.



University of Colorado  
Anschutz Medical Campus

# LMOD ON the login node

```
[@login-ci3 ~]$ curc-quota
```

	Used	Avail	Quota	Limit
/home/	1.5G	551M		2.0G
/projects/	177G	74G		250G
/scratch/alpine1	481G	9056G		9537G
/pl/active/	220G	780G		1000G

- Max space for “/home/username@xsede.org” is 2.0G

# LMOD ON the login node

```
[@login-ci3 ~]$ curc-quota
```

	Used	Avail	Quota	Limit
/home/	1.5G	551M		2.0G
/projects/	177G	74G		250G
/scratch/alpine1	481G	9056G		9537G
/pl/active/	220G	780G		1000G

- Max space for “/projects/username@xsede.org” is 250G

# LMOD ON the login node

```
[@login-ci3 ~]$ curc-quota
```

	Used	Avail	Quota	Limit
/home/	1.5G	551M		2.0G
/projects/	177G	74G		250G
/scratch/alpine1	481G	9056G		9537G
/pl/active/	220G	780G		1000G

- Max space for “/scratch/alpine/username@xsede.org” is around 10TB

# LMOD ON the login node

```
[@login-ci3 ~]$ curc-quota
```

	Used	Avail	Quota	Limit
/home/	1.5G	551M		2.0G
/projects/	177G	74G		250G
/scratch/alpine1	481G	9056G		9537G
/pl/active/	220G	780G		1000G

- Minimum space for “/pl/active” is 1TB

# LMOD ON the login node

```
@xsede.org@login-ci1 ~]$ module overview
```

```
----- /curc/sw/modules/slurm -----
StdEnv (1)  curc-quota (1)  slurm (3)  slurmtools (1)

----- /usr/share/lmod/lmod/modulefiles/Core
lmod (1)  settarg (1)
```

- Module overview displays the number of modules for each name.
- For instance, “**slurm (3)**” counts “**slurm/alpine**”, “**slurm/blanca**” and “**slurm/core**” as shown on slide 21.

# LMOD ON a compute node

```
@xsede.org@login-ci1 ~]$ acompile --ntasks=4 --time=12:00:00
acompile: submitting job... salloc --nodes=1 --partition=acompile --ntasks=4 --time=12:00:00 --mem-per-cpu=3480M --qos=compile --
job-name=acompile --bell --oversubscribe srun --pty /bin/bash
salloc: Granted job allocation 17666833
salloc: Nodes c3cpu-a2-u32-3 are ready for job
```

- Let's access a compute node so that we can navigate through the hpc installed packages.



# LMOD ON a compute node

```
@xsede.org@c3cpu-a2-u32-3 ~]$ module avail
```

----- /curc/sw/modules/slurm -----					
StdEnv	(L,D)	currc-quota/latest	(L,D)	slurm/alpine	(L,D)
ools	(D)			slurm/blanca	
----- /usr/share/lmod/lmod/modulefiles/Core -----					
StdEnv	lmod	(D)	settarg	(D)	
----- Compilers -----					
aocc/3.1.0	(D)	gcc/11.2.0	intel/2022.1.2	(m) nvhpc_sdk/2022.229	nvhpc_sdk/2025.255
aocc/3.2.0		gcc/13.2.0	intel/2024.2.1	(D) nvhpc_sdk/2023.233	(D)
gcc/10.3.0		gcc/14.2.0	(D) nvhpc_sdk/2021.213		nvhpc_sdk/2025.251
----- Independent Applications -----					
R/3.6.3		emacs/27.2	mathematica/11.1.0	(D) rocm/5.5.0	(g)
R/4.2.2		emacs/30.1	(D) matlab/R2018b	rocm/5.6.0	(g)
R/4.4.0	(D)	expat/2.1.1	matlab/R2019b	rocm/6.1.0	(g,D)
allinea/6.0.4	(m)	expat/2.3.0	matlab/R2020b	ruby/2.3.1	
anaconda/2020.11		ffmpeg/4.4	matlab/R2021b	(D) ruby/3.0.0	(D)
anaconda/2022.10		gdb/8.1	matlab/R2022b	singularity/3.6.4	(D)
anaconda/2023.09	(D)	gdb/10.1	(D) matlab/R2023b	singularity/3.7.4	
antlr/4.13.1		ghostscript/9.56.0	matlab/R2024b	slurmtools/0.0.1	
arm-forge/19.1.3	(m)	git-lfs/3.1.2	maven/3.8.1	spack/0.20.1	
autotools/2.69		git/2.31.0	miniforge/24.11.3-0	subversion/1.8.16	
autotools/2.71	(D)	gmsh/2.16.0	ncl/6.3.0	subversion/1.10.2	
chimerax/1.2.5		gmsh/4.11.1	(D) nco/4.8.1	subversion/1.14.1	(D)
clustershell/1.9.2		gnu_parallel/20160622	papi/5.4.3	swig/4.1.1	
cmake/3.5.2		gnu_parallel/20210322	(D) papi/5.5.1	tcltk/8.6.5	
cmake/3.9.2		gnuplot/5.4.3	(D) paraview/5.0.1	tcltk/8.6.11	
cmake/3.14.1		idl/8.7	paraview/5.6.0	tcltk/9.0.1	(D)

- You can see that the sections “Compilers” and “Independent Applications” are now visible.



# LMOD ON a compute node

Bioinformatics					
alphafold/2.2.0	bcftools/1.16	fastqc/0.11.9	nextflow/23.04		qiime2/2024.10_amplicon_gg2
alphafold/2.3.1	bedtools/2.29.1	gatk/4.3.0.0	nextflow/24.04.4	(D)	qiime2/2024.10_amplicon
alphafold/3.0.0	bowtie2/2.5.0	homer/4.11	picard/2.27.5		samtools/1.16.1
alphafold/3.0.1 (D)	bwa/0.7.17	htslib/1.16	plink2/2.00a2.3		sra-toolkit/3.0.0
bamtools/2.5.2	cellranger/7.1.0	multiqc/1.14	qiime2/2023.5		star/2.7.10b
bbtools/39.01	cutadapt/4.2	nextflow/22.10.6	qiime2/2024.2_amplicon (D)		trimmomatic/0.39

----- /usr/share/lmod/8.7.55/modulefiles/Core -----

StdEnv lmod settarg

- Hit “Enter” to scroll down and see the Bioinformatics section.

# LMOD ON a compute node

```
| @xsede.org@c3cpu-a2-u32-3 ~]$ module load R/4.4.0  
| @xsede.org@c3cpu-a2-u32-3 ~]$ module list
```

Currently Loaded Modules:

```
1) slurm/alpine 2) curc-quota/latest 3) StdEnv 4) jdk/18.0.1.1 5) R/4.4.0
```

—

- To load R, we load “module R/4.4.0”
- Since the 4.4.0 is the default version, “module load R” or “ml R” would have worked too.

# LMOD ON a compute node

```
| @xsede.org@c3cpu-a2-u32-3 ~]$ module load R/4.4.0  
| @xsede.org@c3cpu-a2-u32-3 ~]$ module list
```

Currently Loaded Modules:

```
1) slurm/alpine 2) curc-quota/latest 3) StdEnv 4) jdk/18.0.1.1 5) R/4.4.0
```

—

- To load R, we load “module R/4.4.0”
- Since the 4.4.0 is the default version, “module load R” or “ml R” would have worked too.
- Do not load any R version lower than **4.4.0** !

# LMOD ON a compute node

```
| @xsede.org@c3cpu-a2-u32-3 ~]$ module load R/4.4.0  
| @xsede.org@c3cpu-a2-u32-3 ~]$ module list
```

Currently Loaded Modules:

```
1) slurm/alpine 2) curc-quota/latest 3) StdEnv 4) jdk/18.0.1.1 5) R/4.4.0
```

—

Note: If you absolutely need to use an R version lower than **4.4.0** for your research, please open a ticket by emailing [HPCSupport@cuanschutz.edu](mailto:HPCSupport@cuanschutz.edu).

# LMOD ON a compute node

```
| @xsede.org@c3cpu-a2-u32-3 ~]$ module load R/4.4.0  
| @xsede.org@c3cpu-a2-u32-3 ~]$ module list
```

Currently Loaded Modules:

```
1) slurm/alpine 2) curc-quota/latest 3) StdEnv 4) jdk/18.0.1.1 5) R/4.4.0
```

—

- “module list” shows all the loaded modules.
- We can also see that lmod loaded automatically the “jdk/18.0.1.1” dependency in order to load R.

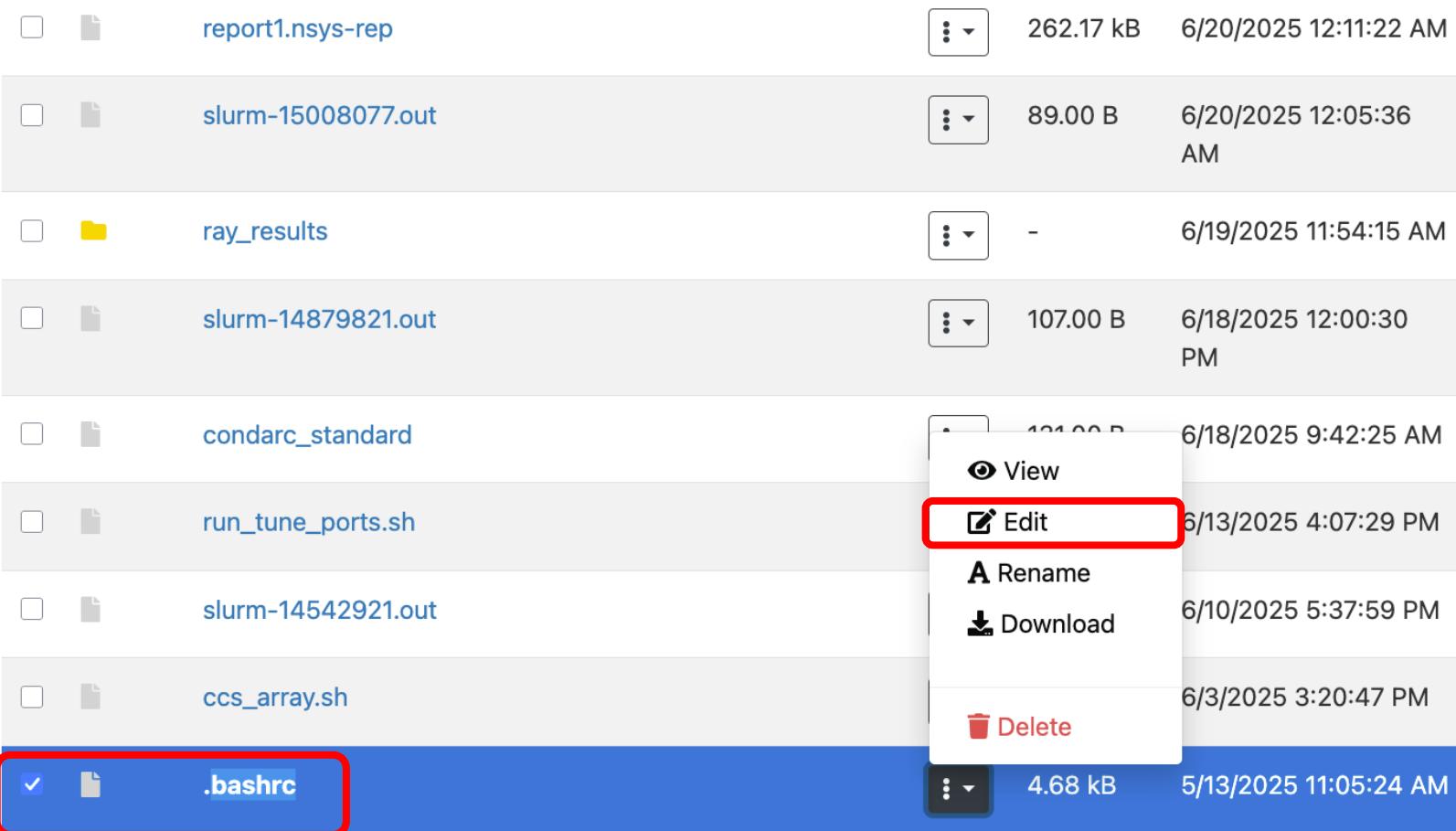
# You can save that command in your .bashrc

The screenshot shows a web-based file manager interface. At the top, there is a navigation bar with icons for Home, Files (which is highlighted with a red box), Jobs, Clusters, Interactive Apps, and Help. Below the navigation bar is a toolbar with buttons for Open in Terminal, Refresh, New File, New Directory, Upload, Download, Copy/Move, and Delete. The main area displays a file listing for the user's home directory (@xsede.org). The directory path is shown as /home/. The interface includes a breadcrumb trail with arrows for navigating up or down the directory structure, and a "Change directory" input field. There are checkboxes for "Show Owner/Mode" and "Show Dotfiles" (which is checked and highlighted with a red box), and a "Filter:" input field. The file listing shows two items: ".condarc" and "&1". Both files are listed with their type (file), name, size (119.00 B and 0.00 B respectively), and last modified date (9/15/2025 9:56:22 AM and 9/15/2025 5:13:14 AM). Each file entry has a three-dot menu icon.

Type	Name	Size	Modified at
File	.condarc	119.00 B	9/15/2025 9:56:22 AM
File	&1	0.00 B	9/15/2025 5:13:14 AM

- Under “Files”, select the Home directory and “Show Dotfiles”

# You can save that command in your .bashrc



- Select the .bashrc so that you can edit it.

# You can save that command in your .bashrc

```
12 # Source global definitions
13 if [ -f /etc/bashrc ]; then
14     . /etc/bashrc
15 fi
16
17
18
19 module load R/4.4.0
```

- That way R/4.4.0 will always be available to any compute node automatically!

# Other module commands

```
@xsede.org@c3cpu-a2-u32-3 ~]$ module show R/4.4.0
```

```
/curc/sw/alpine-modules/idep/R/4.4.0.lua:
```

```
load("jdk/18.0.1.1")
setenv("CURC_R_ROOT","/curc/sw/R/4.4.0")
setenv("CURC_R_LIB","/curc/sw/R/4.4.0/lib64")
setenv("CURC_R_BIN","/curc/sw/R/4.4.0/bin")
help([[R is a free software environment for statistical computing and graphics. It compiles and runs on a wide variety of UNIX platforms, Windows and MacOS.
]])
whatis("Name          : R")
whatis("Version      : 4.4.0")
whatis("Module       : R/4.4.0")
whatis("Category    : language")
whatis("Keyword     : statistical")
whatis("URL         : http://www.r-project.org/")
whatis("License      : GPL")
whatis("Description  : R statistical language")
whatis("Prefix       : /curc/sw/R/4.4.0")
```



- “module show” to find where the .lua config script is located.



# Other module commands

```
@xsede.org@c3cpu-a2-u32-3 ~]$ module show R/4.4.0
```

```
/curc/sw/alpine-modules/idep/R/4.4.0.lua:  
  
load("jdk/18_0_1_1")  
setenv("CURC_R_ROOT","/curc/sw/R/4.4.0")  
setenv("CURC_R_LIB","/curc/sw/R/4.4.0/lib64")  
setenv("CURC_R_BIN","/curc/sw/R/4.4.0/bin")  
help([[R is a free software environment for statistical computing  
and graphics. It compiles and runs on a wide variety of UNIX platforms,  
Windows and MacOS.  
]])  
whatis("Name  
whatis("Version  
whatis("Module  
whatis("Category  
whatis("Keyword  
whatis("URL  
whatis("License  
whatis("Description  
whatis("Prefix : R")  
: 4.4.0")  
: R/4.4.0")  
: language")  
: statistical")  
: http://www.r-project.org/")  
: GPL")  
: R statistical language")  
: /curc/sw/R/4.4.0")
```

- Path to the Root, the library and the binaries.



University of Colorado  
Anschutz Medical Campus

# Other module commands

```
@xsede.org@c3cpu-a2-u32-3 ~]$ module whatis R
```

R/4.4.0	:	Name	:	R
R/4.4.0	:	Version	:	4.4.0
R/4.4.0	:	Module	:	R/4.4.0
R/4.4.0	:	Category	:	language
R/4.4.0	:	Keyword	:	statistical
R/4.4.0	:	URL	:	<a href="http://www.r-project.org/">http://www.r-project.org/</a>
R/4.4.0	:	License	:	GPL
R/4.4.0	:	Description	:	R statistical language
R/4.4.0	:	Prefix	:	/curc/sw/R/4.4.0

- “whatis” parameter, gives more details about the software being used.

# Other module commands

```
@xsede.org@c3cpu-a2-u32-3 ~]$ module load openmpi  
Lmod has detected the following error: These module(s) or extension(s) exist  
but cannot be loaded as requested: "openmpi"  
Try: "module spider openmpi" to see how to load the module(s).
```



- As you can see, we cannot load module openmpi, but why?

# Other module commands

```
@xsede.org@c3cpu-a2-u32-3 ~]$ module spider openmpi
```

openmpi:

Description:

Open Message Passing Interface

Versions:

openmpi/4.1.1  
openmpi/4.1.4  
openmpi/5.0.6

For detailed information about a specific "openmpi" package  
Note that names that have a trailing (E) are extensions pro  
For example:

```
$ module spider openmpi/5.0.6
```

- “module spider” shows that there are different versions of openmpi.
- But they don’t share the same dependency space/requirements.



University of Colorado  
Anschutz Medical Campus

# Other module commands

```
@xsede.org@c3cpu-a2-u32-3 ~]$ module spider openmpi
```

---

openmpi:

---

Description:

Open Message Passing Interface

Versions:

openmpi/4.1.1

openmpi/4.1.4

openmpi/5.0.6

---

For detailed information about a specific "openmpi" package  
Note that names that have a trailing (E) are extensions pro  
For example:

```
$ module spider openmpi/5.0.6
```

---



- We run “module spider” on a specific version to find out more.



# Other module commands

```
@xsede.org@c3cpu-a2-u32-3 ~]$ module spider openmpi/5.0.6
```

---

```
openmpi: openmpi/5.0.6
```

---

Description:  
Open Message Passing Interface

You will need to load all module(s) on any one of the lines below before the "openmpi/5.0.6" module is available to load.

```
gcc/14.2.0
```

Help:

This module loads the OpenMPI provider

C: mpicc

C++: mpicxx

Fortran: mpif90

For more information on the individual compilers and their suboptions  
refer to the man page for the individual compilers.



- We will need to load gcc/14.2.0 as a requirement to access openmpi/5.0.6



# Other module commands

```
[@xsede.org@c3cpu-a2-u32-3 ~]$ module load gcc/14.2.0 openmpi/5.0.6  
[@xsede.org@c3cpu-a2-u32-3 ~]$ module list
```

Currently Loaded Modules:

```
1) slurm/alpine 2) curc-quota/latest 3) StdEnv 4) gcc/14.2.0 5) openmpi/5.0.6
```

# Other module commands

```
@xsede.org@c3cpu-a2-u32-3 ~]$ module avail openmpi
```

openmpi/5.0.6 (L)

Where:

L: Module is loaded

If the available list is too long consider trying:

"module --default avail" or "ml -d av" to just list the default modules.

"module overview" or "ml ov" to display the number of modules for each name.

Use "module consider" to find all possible modules and extensions.

Use "module search keyword key1 key2 ..." to search for all possible modules matching any of the "keys".

- We can see that openmpi/5.0.6 is there but we do not see any other version of openmpi.



# Other module commands

```
@xsede.org@c3cpu-a2-u32-3 ~]$ module avail openmpi
```

openmpi/5.0.6 (L)

Where:

L: Module is loaded

If the available list is too long consider trying:

"module --default avail" or "ml -d av" to just list the default modules.

"module overview" or "ml ov" to display the number of modules for each name.

Use "module consider" to find all possible modules and extensions.

Use "module search keyword key1 key2 ..." to search for all possible modules matching any of the "keys".

- We can see that openmpi/5.0.6 is there but we do not see any other version of openmpi.
- Because they do not share the same dependencies/requirements space.

# Other module commands

```
@xsede.org@c3cpu-a2-u32-3 ~]$ module avail openmpi
```

openmpi/5.0.6 (L)

Where:

L: Module is loaded

If the avail list is too long consider trying:

"module --default avail" or "ml -d av" to just list the default modules.

"module overview" or "ml ov" to display the number of modules for each name.

Use "module spider" to find all possible modules and extensions.

Use "module keyword key1 key2 ..." to search for all possible modules matching any of the "keys".

- Remember to run “**module spider <package>** if you do not see a package at first.

# Other module commands

```
[@xsede.org@c3cpu-a2-u32-3 ~]$ module unload openmpi  
[@xsede.org@c3cpu-a2-u32-3 ~]$ module list
```

Currently Loaded Modules:

1) slurm/alpine    2) curc-quota/latest    3) StdEnv    4) gcc/14.2.0

- “module unload openmpi” removes the openmpi package.

# Other module commands

```
[@xsede.org@c3cpu-a2-u32-3 ~]$ module purge
[@xsede.org@c3cpu-a2-u32-3 ~]$ module list
No modules loaded
| @xsede.org@c3cpu-a2-u32-3 ~]$ █
```

- “module purge” removes all packages that were loaded.

# Other module commands

```
[@xsede.org@c3cpu-a2-u32-3 ~]$ module load gcc/11  
[@xsede.org@c3cpu-a2-u32-3 ~]$ module list
```

Currently Loaded Modules:

- 1) gcc/11.2.0

```
[@xsede.org@c3cpu-a2-u32-3 ~]$ module swap gcc/14
```

The following have been reloaded with a version change:

- 1) gcc/11.2.0 => gcc/14.2.0

- “module swap” to load a different version of the package.



University of Colorado  
Anschutz Medical Campus

# Build a Slurm script with lmod

```
#!/bin/bash

#SBATCH --partition=amilan
#SBATCH --job-name=first_mpirun
#SBATCH --output=first_mpirun-job.%j.out
#SBATCH --error=first_mpirun-job.%j.err
#SBATCH --account=amc-general
#SBATCH --qos=normal
#SBATCH --nodes=1
#SBATCH --ntasks=4
#SBATCH --mail-type=ALL
#SBATCH --mail-user=
#SBATCH --time=00:00:01

module load gcc
module load openmpi
mpirun -n $SLURM_NTASKS hostname
```

- Packages **are not loaded automatically!**  
You need to always load them.
- Unless you saved that command into the .bashrc!

# Slurm helpful tip.

```
[@xsede.org@login-ci1 ~]$ which sbatch
alias sbatch='sbatch --export=NONE'
/usr/bin/sbatch
[@xsede.org@login-ci1 ~]$ sbatch slurm_wrapper_script.sh
```

- If you run “which sbatch” on the login node you can see that it is an “alias” for a much longer command.
- This helps **to not** carry all the env variables from the login a node to a compute node

# Slurm helpful tip.

```
[@xsede.org@login-ci1 ~]$ which sbatch
alias sbatch='sbatch --export=NONE'
    /usr/bin/sbatch
[@xsede.org@login-ci1 ~]$ sbatch slurm_wrapper_script.sh
```

- This will help make sure that LMOD is set up properly on any compute node.



# Slurm helpful tip.

```
#!/bin/bash

# Setting up sbatch to avoid dependency error
alias sbatch='sbatch --export=N0NE'

job1=$(sbatch test_pipeline.sh )
```



- Any custom wrapper script for slurm batch submission should contain the following alias command above.

# Custom lua script

- This module loads MultiQC

```
-- Load the package defaults
local PACKAGE_PREFIX = "/curc/sw/install/bio/multiqc/1.14_env"
local USER = os.getenv("USER")

-- JDK dependency
always_load("jdk")

-- Set the paths
prepend_path("PATH", pathJoin(PACKAGE_PREFIX, "bin"))
prepend_path("LD_LIBRARY_PATH", pathJoin(PACKAGE_PREFIX, "lib"))

os.getenv("USER")

-- Set environment variables
setenv("CURC_MULTIQC_ROOT", PACKAGE_PREFIX)
setenv("CURC_MULTIQC_BIN", pathJoin(PACKAGE_PREFIX, "bin"))
setenv("CURC_MULTIQC_LIB", pathJoin(PACKAGE_PREFIX, "lib"))
setenv("CURC_MULTIQC_INC", pathJoin(PACKAGE_PREFIX, "include"))
setenv("MQC_HOME", pathJoin("/scratch/alpine", USER))
setenv("MQC_TEMP", pathJoin("/scratch/alpine", USER))
setenv("MQC_WORK", pathJoin("/scratch/alpine", USER))
```



University of Colorado  
Anschutz Medical Campus

- You can install your software and decide to make your custom lua script.

# Custom lua script

- This module loads MultiQC

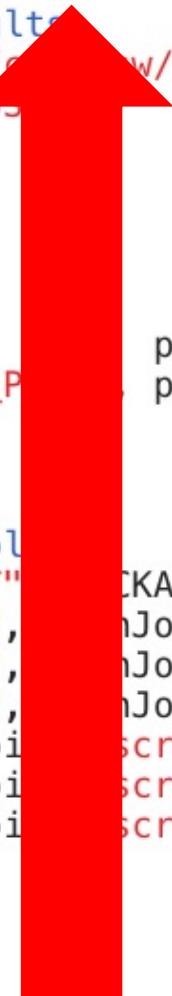
```
-- Load the package defaults
local PACKAGE_PREFIX = "/opt/bio/install/bio/multiqc/1.14_env"
local USER = os.getenv("USER")

-- JDK dependency
always_load("jdk")

-- Set the paths
prepend_path("PATH",
pathJoin(PACKAGE_PREFIX, "bin"))
prepend_path("LD_LIBRARY_PATH",
pathJoin(PACKAGE_PREFIX, "lib"))

os.getenv("USER")

-- Set environment variables
setenv("CURC_MULTIQC_ROOT", PACKAGE_PREFIX)
setenv("CURC_MULTIQC_BIN", pathJoin(PACKAGE_PREFIX, "bin"))
setenv("CURC_MULTIQC_LIB", pathJoin(PACKAGE_PREFIX, "lib"))
setenv("CURC_MULTIQC_INC", pathJoin(PACKAGE_PREFIX, "include"))
setenv("MQC_HOME", pathJoin("$scratch/alpine", USER))
setenv("MQC_TEMP", pathJoin("$scratch/alpine", USER))
setenv("MQC_WORK", pathJoin("$scratch/alpine", USER))
```



- “--” means that it is a comment.



# Custom lua script

– This module loads MultiQC

```
-- Load the package defaults
local PACKAGE_PREFIX = "/curc/sw/install/bio/multiqc/1.14_env"
local USER = os.getenv("USER")

-- JDK dependency
always_load("jdk")

-- Set the paths
prepend_path("PATH", pathJoin(PACKAGE_PREFIX, "bin"))
prepend_path("LD_LIBRARY_PATH", pathJoin(PACKAGE_PREFIX, "lib"))

os.getenv("USER")

-- Set environment variables
setenv("CURC_MULTIQC_ROOT", PACKAGE_PREFIX)
setenv("CURC_MULTIQC_BIN", pathJoin(PACKAGE_PREFIX, "bin"))
setenv("CURC_MULTIQC_LIB", pathJoin(PACKAGE_PREFIX, "lib"))
setenv("CURC_MULTIQC_INC", pathJoin(PACKAGE_PREFIX, "include"))
setenv("MQC_HOME", pathJoin("/scratch", USER))
setenv("MQC_TEMP", pathJoin("/scratch", USER))
setenv("MQC_WORK", pathJoin("/scratch", USER))
```



- This is going to be the root path where the package is installed.
- But in your case please install packages under “/projects” or “/pl/active”!



# Custom lua script

- This module loads MultiQC

```
-- Load the package defaults
local PACKAGE_PREFIX = "/curc/sw/install/bio/multiqc/1.14_env"
local USER = os.getenv("USER")

-- JDK dependency
always_load("jdk")

-- Set the paths
prepend_path("PATH", pathJoin(PACKAGE_PREFIX, "bin"))
prepend_path("LD_LIBRARY_PATH", pathJoin(PACKAGE_PREFIX, "lib"))

os.getenv("USER")

-- Set environment variables
setenv("CURC_MULTIQC_ROOT", PACKAGE_PREFIX)
setenv("CURC_MULTIQC_BIN", pathJoin(PACKAGE_PREFIX, "bin"))
setenv("CURC_MULTIQC_LIB", pathJoin(PACKAGE_PREFIX, "lib"))
setenv("CURC_MULTIQC_INC", pathJoin(PACKAGE_PREFIX, "include"))
setenv("MQC_HOME", pathJoin("batch/alpine", USER))
setenv("MQC_TEMP", pathJoin("batch/alpine", USER))
setenv("MQC_WORK", pathJoin("batch/alpine", USER))
```

# Custom lua script

-- This module loads MultiQC

-- Load the package defaults

```
local PACKAGE_PREFIX = "/curc/sw/install/bio/multiqc/1.14_env"  
local USER = os.getenv("USER")
```

-- JDK dependency

```
always_load("jdk")
```

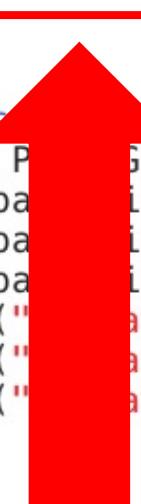
-- Set the paths

```
prepend_path("PATH", pathJoin(PACKAGE_PREFIX, "bin"))  
prepend_path("LD_LIBRARY_PATH", pathJoin(PACKAGE_PREFIX, "lib"))
```

```
os.getenv("USER")
```

-- Set environment variables

```
setenv("CURC_MULTIQC_ROOT", PACKAGE_PREFIX)  
setenv("CURC_MULTIQC_BIN", pathJoin(PACKAGE_PREFIX, "bin"))  
setenv("CURC_MULTIQC_LIB", pathJoin(PACKAGE_PREFIX, "lib"))  
setenv("CURC_MULTIQC_INC", pathJoin(PACKAGE_PREFIX, "include"))  
setenv("MQC_HOME", pathJoin("batch/alpine", USER))  
setenv("MQC_TEMP", pathJoin("batch/alpine", USER))  
setenv("MQC_WORK", pathJoin("batch/alpine", USER))
```



- Adding install directories to PATH and LD\_LIBRARY\_PATH.
- That's why LMOD is nice because you don't have to worry about that retrospectively.



# Custom lua script

- This module loads MultiQC

```
-- Load the package defaults
local PACKAGE_PREFIX = "/curc/sw/install/bio/multiqc/1.14_env"
local USER = os.getenv("USER")

-- JDK dependency
always_load("jdk")

-- Set the paths
prepend_path("PATH", pathJoin(PACKAGE_PREFIX, "bin"))
prepend_path("LD_LIBRARY_PATH", pathJoin(PACKAGE_PREFIX, "lib"))

os.getenv("USER")

-- Set environment variables
setenv("CURC_MULTIQC_ROOT", PACKAGE_PREFIX)
setenv("CURC_MULTIQC_BIN", pathJoin(PACKAGE_PREFIX, "bin"))
setenv("CURC_MULTIQC_LIB", pathJoin(PACKAGE_PREFIX, "lib"))
setenv("CURC_MULTIQC_INC", pathJoin(PACKAGE_PREFIX, "include"))
setenv("MQC_HOME", pathJoin("/scratch/alpine", USER))
setenv("MQC_TEMP", pathJoin("/scratch/alpine", USER))
setenv("MQC_WORK", pathJoin("/scratch/alpine", USER))
```

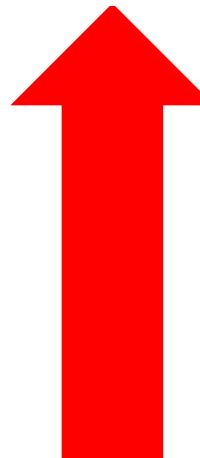


- Additional system ENV need to be set up, including the temporary directory.



# Custom lua script

```
help([[  
MultiQC:  
MultiQC is a reporting tool that parses results and statistics from bioinformatics tool outputs, such as log files and c  
experiments containing multiple samples and multiple analysis steps. It's designed to be placed at the end of pipelines  
nished running your tools.  
  
For detailed instructions, go to:  
https://multiqc.info/docs/  
]])  
whatis("Version: 1.14")  
whatis("MultiQC: Summarize analysis results for multiple tools and samples in a single report.  
Philip Ewels, Måns Magnusson, Sverker Lundin and Max Käller Bioinformatics (2016) doi: 10.1093/bioinformatics/btw  
PMID: 27312411")
```

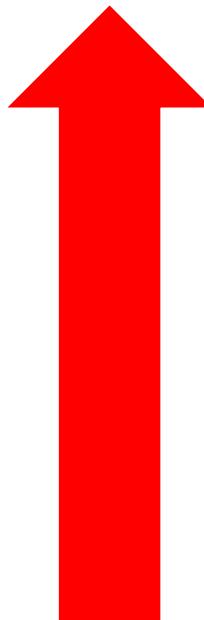


- For the software description, software version and reference.



# Custom lua script

```
[      @xsede.org@c3cpu-a2-u32-3 ~]$ module whatis multiqc  
multiqc/1.14          : Version: 1.14  
multiqc/1.14          : Citation: MultiQC: Summarize analysis |  
Philip Ewels, Måns Magnusson, Sverker Lundin and Max Käller |  
MID: 27312411
```



- This is how it appears on LMOD.



# Custom lua script

```
@login-ci5 bio]$ pwd  
/projects/lmod-lua-local/bio  
@login-ci5 bio]$ tree
```

```
.  
├── bbtools  
│   └── 39.01.lua  
├── fastQC  
│   └── 22.10.6.lua  
├── multiQC  
│   └── 1.14.lua  
├── nextflow  
│   └── 22.10.6.lua  
├── samtools  
└── star  
    └── 1.14.lua
```

6 directories, 5 files

- Make sure that you have a common root directory that will contain all the lua folders.



# Custom lua script

```
@login-ci5 bio]$ pwd  
/projects/lmod-lua-local/bio  
@login-ci5 bio]$ tree
```

```
•  
  └── bbtools  
      └── 39.01.lua  
  └── fastQC  
      └── 22.10.6.lua  
  └── multiQC  
      └── 1.14.lua  
  └── nextflow  
      └── 22.10.6.lua  
  └── samtools  
  └── star  
      └── 1.14.lua
```

6 directories, 5 files

- I have a multiQC folder that contains a lua file.
- The lua file is named after the package version.



University of Colorado  
Anschutz Medical Campus

# Custom lua script

```
@c3cpu-a2-u32-3 lmod-lua-local]$ module use --append /projects/      /lmod-lua-local/bio  
@c3cpu-a2-u32-3 lmod-lua-local]$ module avail multiqc  
----- Bioinformatics -----  
multiqc/1.14  
----- /projects/      /lmod-lua-local/bio -----  
multiQC/1.14
```



- Use "module –append" to add that path to the LMOD environment variable.



# Custom lua script

```
@c3cpu-a2-u32-3 lmod-lua-local]$ module use --append /projects/      /lmod-lua-local/bio  
@c3cpu-a2-u32-3 lmod-lua-local]$ module avail multiqc  
-----  
multiqc/1.14                                         Bioinformatics  
-----  
multiQC/1.14                                         /projects/      /lmod-lua-local/bio  
-----
```



- Now Imod can see both the system configured multiqc as well as my own multiqc configured module.



# Custom lua script

```
@c3cpu-a2-u32-3 lmod-lua-local]$ module use --append /projects/      /lmod-lua-local/bio  
@c3cpu-a2-u32-3 lmod-lua-local]$ module avail multiqc  
-----  
multiqc/1.14                                         Bioinformatics  
-----  
multiQC/1.14                                         /projects/      /lmod-lua-local/bio  
-----
```



- Note that you will need to add “use --append” to all your slurm script or your .bashrc file **otherwise your module won’t load !**





University of Colorado **Anschutz Medical Campus**

# Part 2 – Miniforge

## Option A: Things we can do once we are interactively inside of a compute node...

- **Install our own software – using** miniforge or anaconda



Recall, that your home directory is really, really small (only 2GB), so we always want to make sure software is installed in our projects directory (/projects/\$USER/). By default conda likes to install in home, so before do any conda work, we must change our .condarc file to include the following lines:

```
pkgs_dirs:  
  - /projects/$USER/.conda_pkgs  
envs_dirs:  
  - /projects/$USER/software/anaconda/envs
```

This only every needs to be done 1 time for the lifetime of your Alpine account!

# Option A: Things we can do once we are interactively inside of a compute node...

- Through a miniforge or anaconda environment

- 1 Navigate to your Home Directory within your OnDemand session:

The screenshot shows the XSEDE OnDemand web interface. At the top, there is a navigation bar with links for Files, Jobs, Clusters, Interactive Apps, and a user icon. Below the navigation bar, a sidebar on the left lists directory paths: Home Directory, /scratch/alpine/, /projects, and /pl. The 'Home Directory' link is highlighted with a red circle. The main content area shows a breadcrumb trail: / home / @xsede.org /. There are checkboxes for 'Show Owner/Mode', 'Show Dotfiles', and a 'Filter' input field. Below the trail, it says 'Showing 3 of 36 rows - 0 rows selected'. A table lists three items: 'ondemand' (modified 9/6/2022 3:39:58 PM), 'perl5' (modified 7/6/2023 1:20:16 PM), and 'README.mdwn' (modified 2/1/2018 8:35:24 AM). The table has columns for Type, Name, Size, and Modified at.

Type	Name	Size	Modified at
Folder	ondemand	-	9/6/2022 3:39:58 PM
Folder	perl5	-	7/6/2023 1:20:16 PM
File	README.mdwn	562 Bytes	2/1/2018 8:35:24 AM

# Option A: Things we can do once we are interactively inside of a compute node...

- Through a miniforge or anaconda environment

2 Make sure to click on the box that say "Show Dotfiles"

The screenshot shows a file management interface with a toolbar at the top. The toolbar includes buttons for Open in Terminal, New File, New Directory, Upload, Download, Copy/Move, and Delete. Below the toolbar, a message states: "Notice: Users will be limited to a maximum of 8 cores per Core Desktop session through mid-September due to ongoing maintenance." On the left, there is a sidebar with navigation links: Home Directory, /scratch/alpine, /projects/, and /pl. The main area displays a list of files in the user's home directory. The list includes hidden files like .cache, .conda, .config, .cpan, and .dbus. The "Show Dotfiles" checkbox in the toolbar is checked and highlighted with a red circle. The table below shows the file details:

Type	Name	Size	Modified at
Folder	.cache	-	8/22/2023 12:45:26 PM
Folder	.conda	-	6/8/2023 1:05:28 PM
Folder	.config	-	8/21/2023 4:20:25 PM
Folder	.cpan	-	7/6/2023 1:22:45 PM
Folder	.dbus	-	5/5/2023 11:32:05 AM

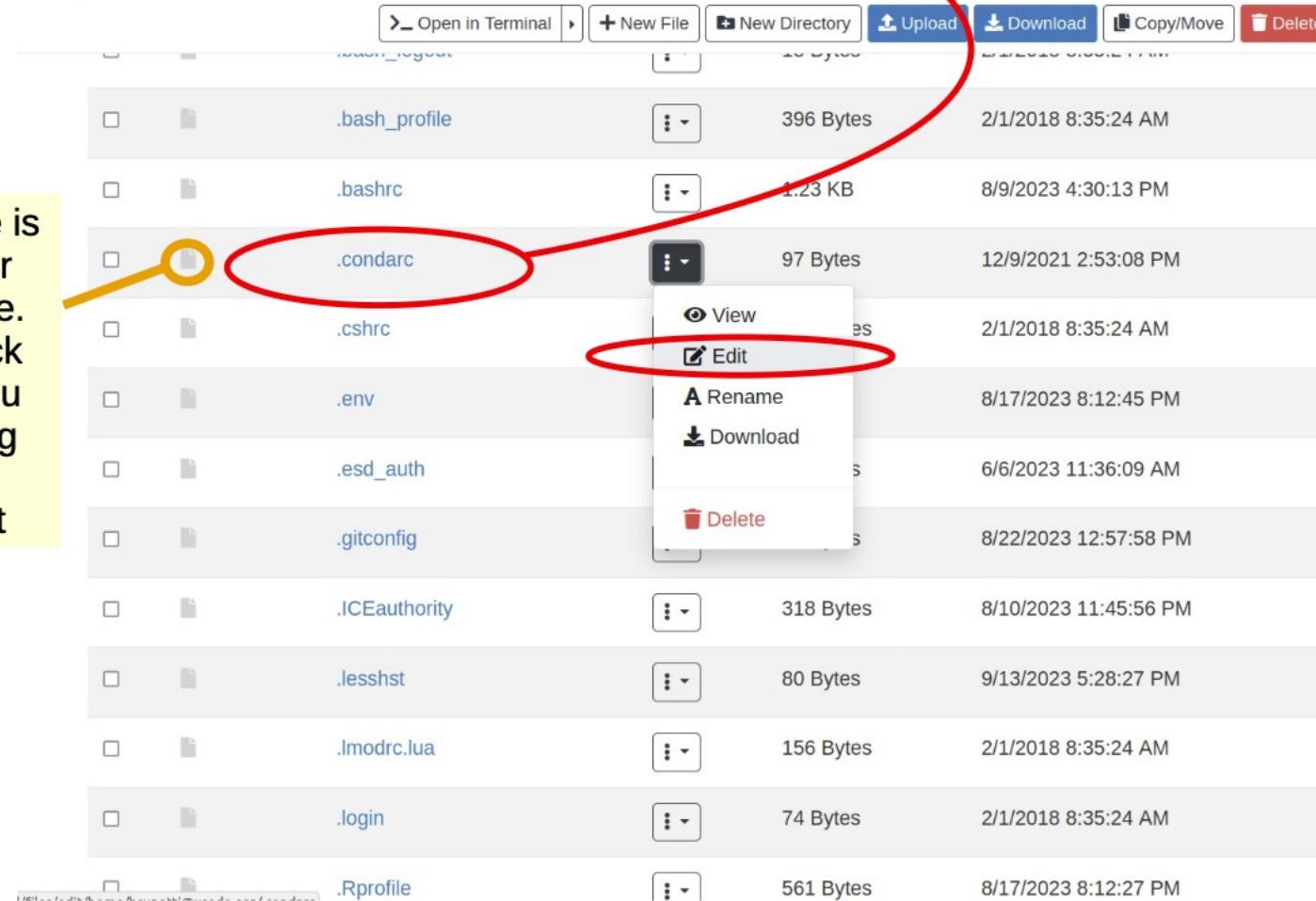
# Option A: Things we can do once we are interactively inside of a compute node...

- Through a miniforge or anaconda environment

- 3 Scrolls down your list and find the file called .condarc and click the three dots and select Edit from the drop down menu



Be careful! There is a .condarc folder and a .condarc file. Make sure to click the file, which you can tell by seeing the little paper image to the left

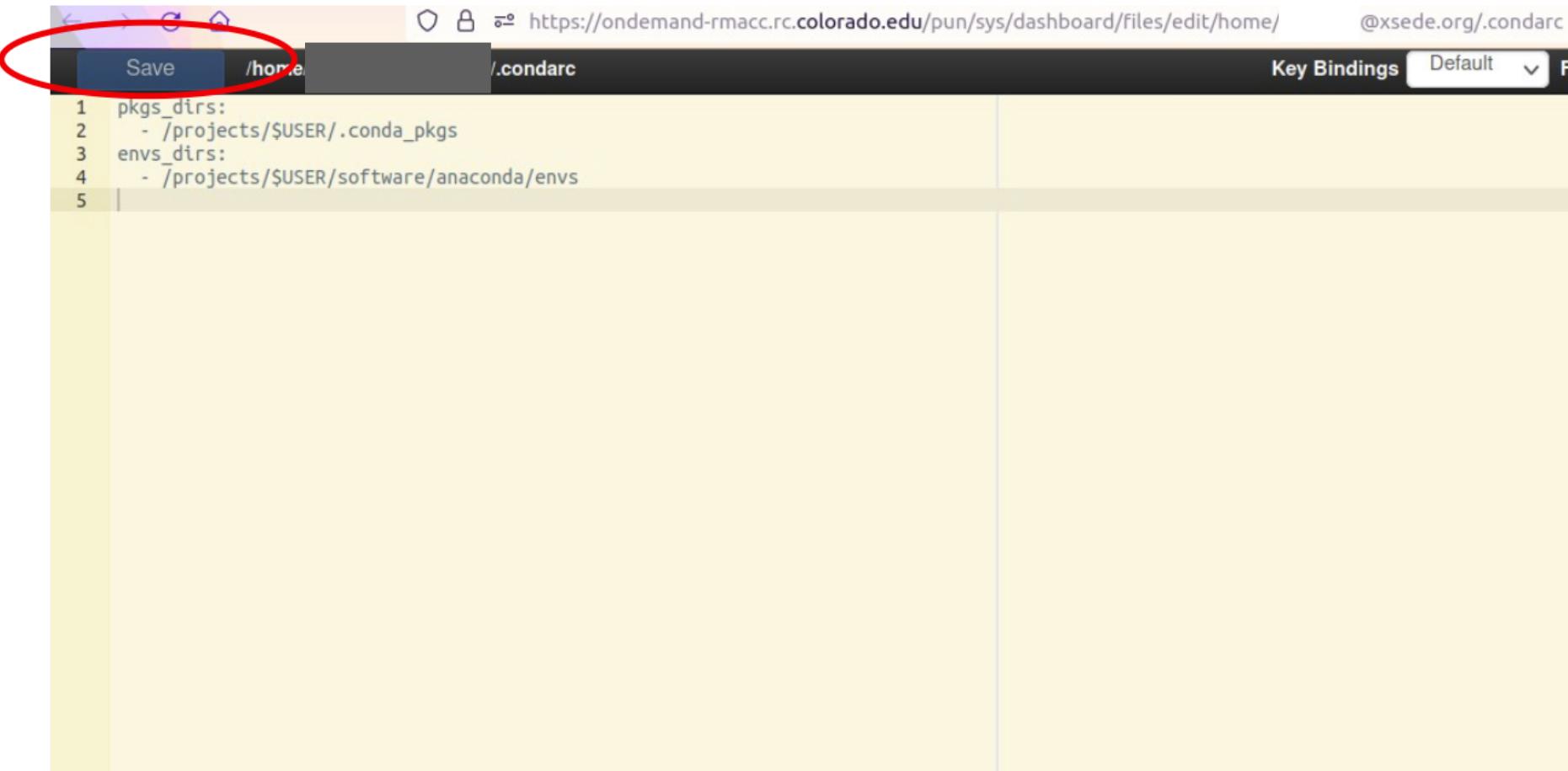


# Option A: Things we can do once we are interactively inside of a compute node...

- Through a miniforge or anaconda environment

4

This takes you to a text editor where you can copy and paste the following lines. Then be sure to press Save in the upper left hand corner



```
1 pkgs_dirs:
2   - /projects/$USER/.conda_pkgs
3 envs_dirs:
4   - /projects/$USER/software/anaconda/envs
5 |
```

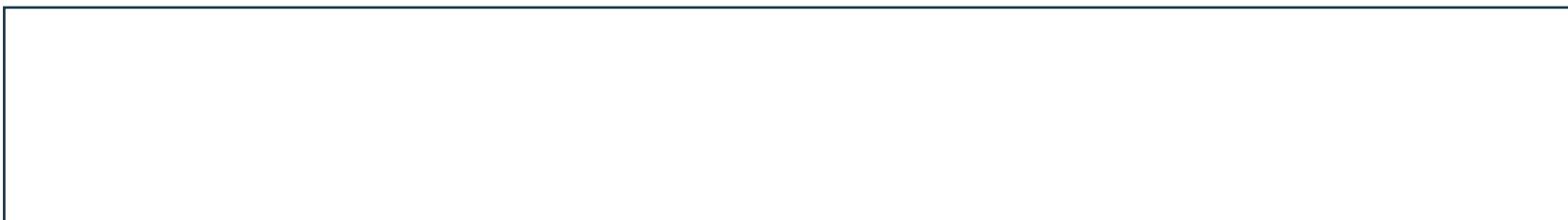
## Option A: Things we can do once we are interactively inside of a compute node...

- **Through a** miniforge or anaconda environment
- 5 After you press save, you can exit the browser tab and you are done configuring conda!

## Option A: Things we can do once we are interactively inside of a compute node...

- **Through a** miniforge or anaconda environment
- 5 After you press save, you can exit the browser tab and you are done configuring conda!

Now that conda is configured, let's go through a tutorial of how to install software that has a conda installation option.



# Conda environment

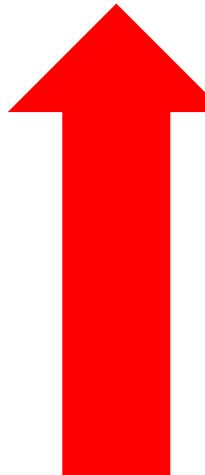
- /home or /tmp have a very small size. Please include the following before installing a package.
- You will also make sure that all the cache from pip get stored in the scratch filesystem.

```
export TMP=/gpfs/alpine1/scratch/$USER/cache_dir  
mkdir -pv $TMP  
export TEMP=$TMP  
export TMPDIR=$TMP  
export TEMPDIR=$TMP  
export PIP_CACHE_DIR=$TMP
```



# Conda environment

```
conda create --name python env python=3.9 -y
```



- Name of our python environment.

# Conda environment

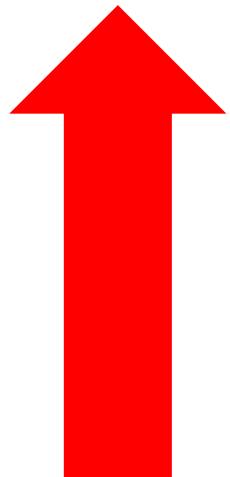
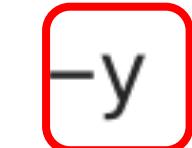
```
conda create --name python_env python=3.9 -y
```



- We installed python 3.9 in the environment.

# Conda environment

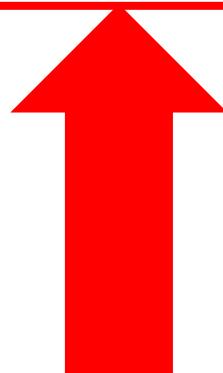
```
conda create --name python_env python=3.9 -y
```



- If we do not add "-y", miniforge will ask us to confirm the list of packages that we want installed.

# Conda environment

```
conda activate python_env
```



- Then we activate it.



# Best practices on Alpine

```
(glibc_env) [1] @c3cpu-a2-u32-3 bio]$ which python  
/curc/sw/install/miniforge3/24.11.3-0/bin/python  
(glibc_env) [2] @c3cpu-a2-u32-3 bio]$
```



- Note that not all environments contain python.
- Here for instance, we are using the system python.
- Always check which python/pip you are using before installing a package.

# Best practices on Alpine

```
(glibc_env) [1] @c3cpu-a2-u32-3 bio]$ which python  
/curc/sw/install/miniforge3/24.11.3-0/bin/python  
(glibc_env) [2] @c3cpu-a2-u32-3 bio]$
```



- To fix that, you can install a python package inside your conda ENV.
- You can also use the system python/pip to install packages, but you will need to specify the target directory.

# Best practices on Alpine

```
# Loading python and ROCM modules
module load python/3.10.2 rocm/5.6.0

# Exporting TMP and cache folders so that they do not affect the filesystem /tmp
export TMP=/gpfs/alpine1/scratch/$USER/cache_dir
mkdir -pv $TMP
export TEMP=$TMP
export TMPDIR=$TMP
export TEMPDIR=$TMP
export PIP_CACHE_DIR=$TMP
```

```
# Setting up install directories and pythonpath:
export PIP_INSTALL_DIR=${PWD}
export PYTHONPATH=$PYTHONPATH:$PIP_INSTALL_DIR

# Installing packages based on requirement file
python -m pip install -r requirements.txt --target=$PIP_INSTALL_DIR
python -m pip install -r requirements_2.txt --target=$PIP_INSTALL_DIR
```

- Here is an example on how to use the system pip

# Best practices on Alpine

```
echo 'CUDNN_PATH=$(dirname $(python -c "import nvidia.cudnn;print(nvidia.cudnn._file_)"))' >>
$CONDA_PREFIX/etc/conda/activate.d/env_vars.sh
echo 'export LD_LIBRARY_PATH=$CONDA_PREFIX/lib/:$CUDNN_PATH/lib:$LD_LIBRARY_PATH' >>
$CONDA_PREFIX/etc/conda/activate.d/env_vars.sh
```

- Sometimes Alpine won't remember the path to your miniforge environment libraries.
- You might need to force those path into your “env\_vars.sh” file or at least add those paths into your slurm script.

# Best practices on Alpine

- Please follow this guide so that you can load your conda environment as a kernel. into Jupyter Ondemand.

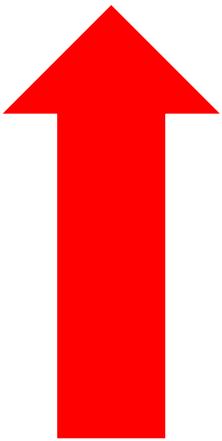
[https://curc.readthedocs.io/en/latest/open\\_ondemand/jupyter\\_session.html#creating-your-own-custom-jupyter-kernel](https://curc.readthedocs.io/en/latest/open_ondemand/jupyter_session.html#creating-your-own-custom-jupyter-kernel)



University of Colorado  
Anschutz Medical Campus

# Best practices on Alpine

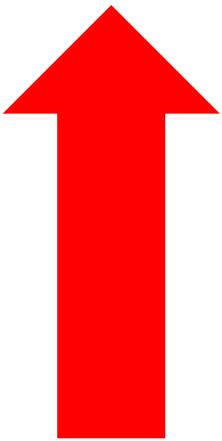
```
conda env export > python_env.yml
```



- You can export your conda environment as a yaml file.
- It improves software reproducibility.

# Best practices on Alpine

```
conda env export > python_env.yml
```



- To import the yaml file please follow the command below.
- `conda env create -n imported_env --file python_env.yaml`



University of Colorado **Anschutz Medical Campus**

# THANK YOU