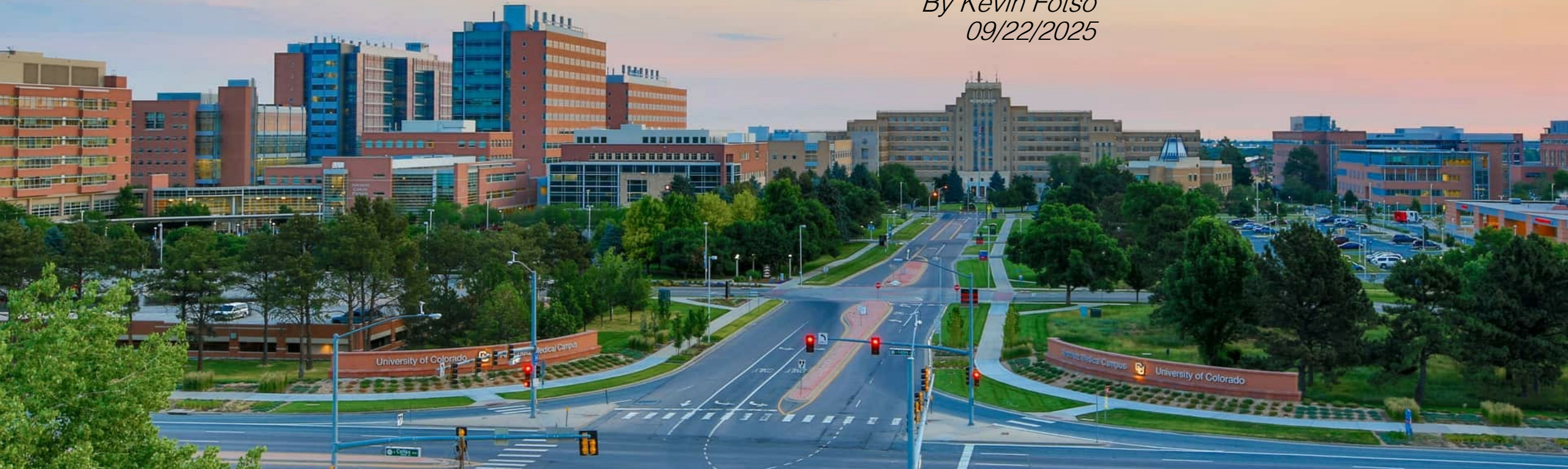
University of Colorado **Anschutz Medical Campus**

How to use R on Alpine?

By Kevin Fotso
09/22/2025



Introduction:

This workshop will cover the following topics:

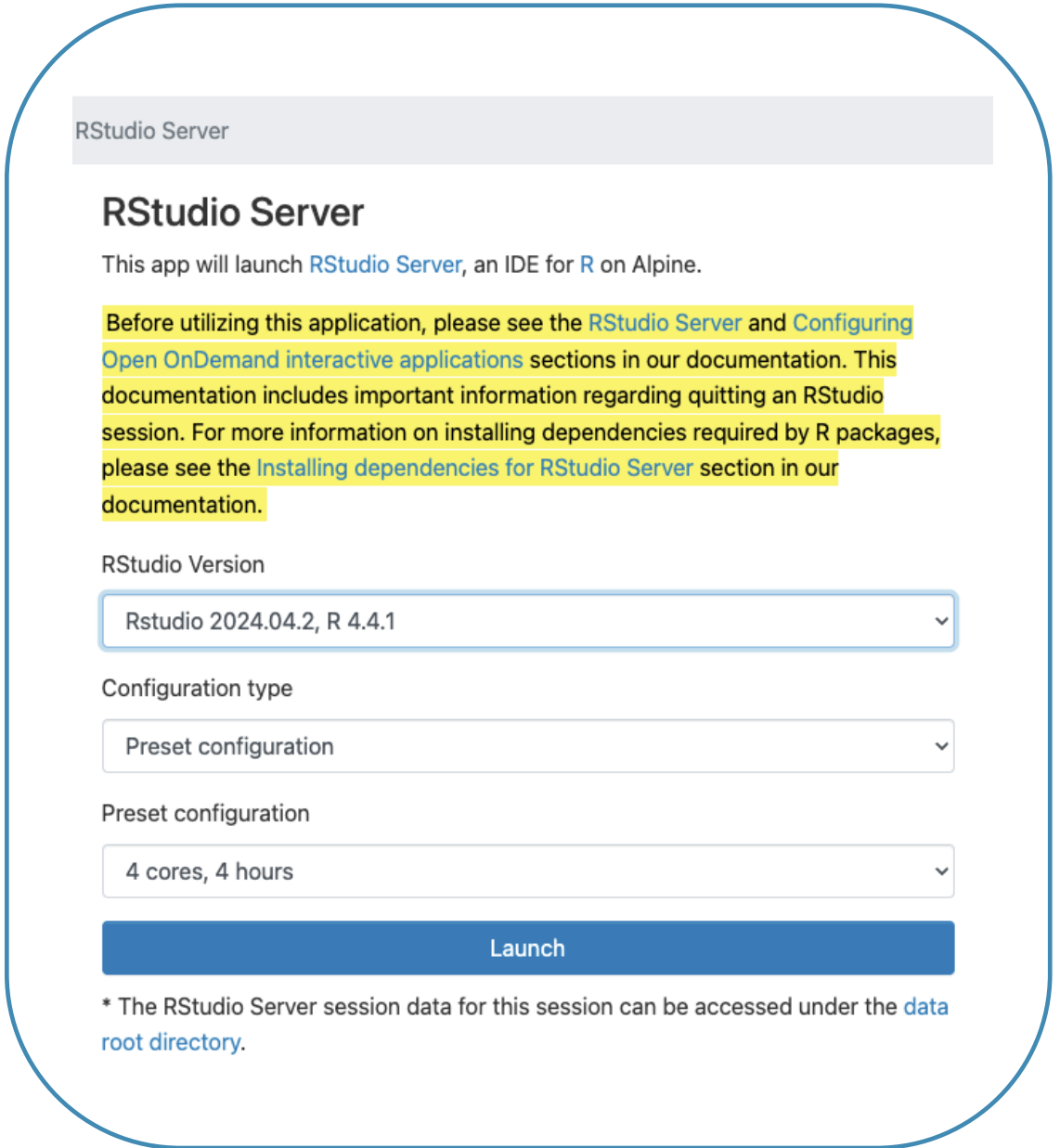
- Overview of R on Alpine.
- How to use R with Ondemand?
- How to use R with the LMOD module stack?
- How to use R with miniforge?

Audience:

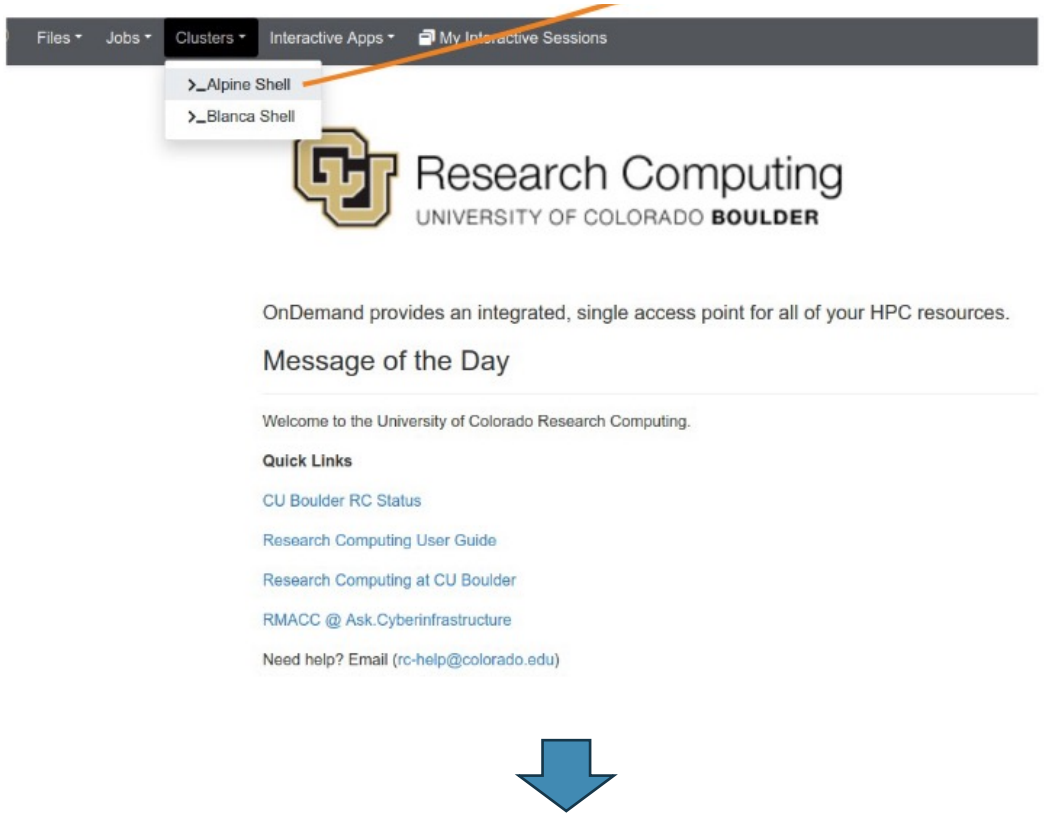
Any R user who is new to Alpine.

Ways to use R on Alpine

- Rstudio on Ondemand with versions **4.4.1** and **4.2.2**.
- The version **4.4.1** is strongly recommended!



Ways to use R on Alpine



- Under the Alpine shell, there is an R version 4.4.0 that is accessible by calling **LMOD**, the modern environment module system for HPC.

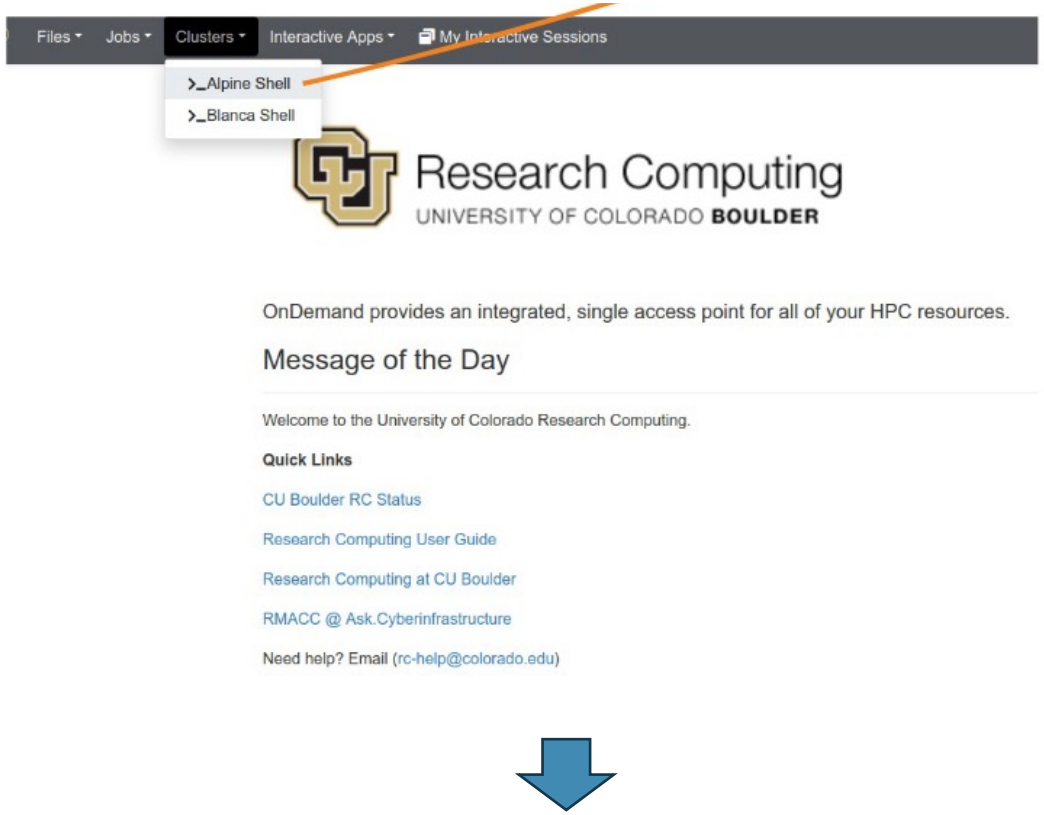
- We covered LMOD last week here:

  https://github.com/kf-cuanschutz/CU-Anschutz-HPC-documentation/blob/main/Workshops/LMOD_and_anaconda_v2.pdf

```
Currently Loaded Modules:
  1) jdk/18.0.1.1   2) R/4.4.0

module load R/4.4.0
module list
```

Ways to use R on Alpine

One may decide to install R using **miniforge** or anaconda. The latest R version they can install is R **4.4.3.**




```
module load miniforge
```

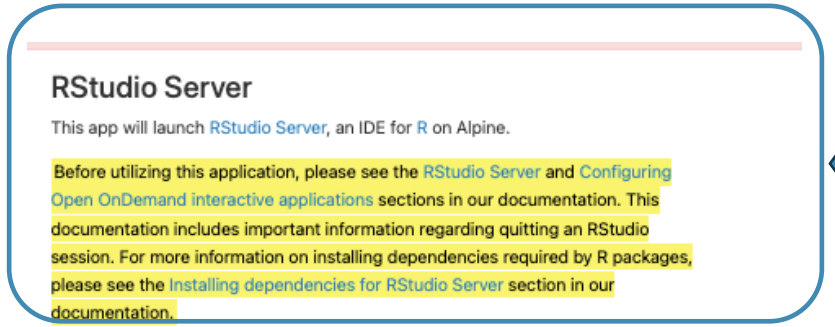
Part I- Ondemand R

Ondemand Rstudio

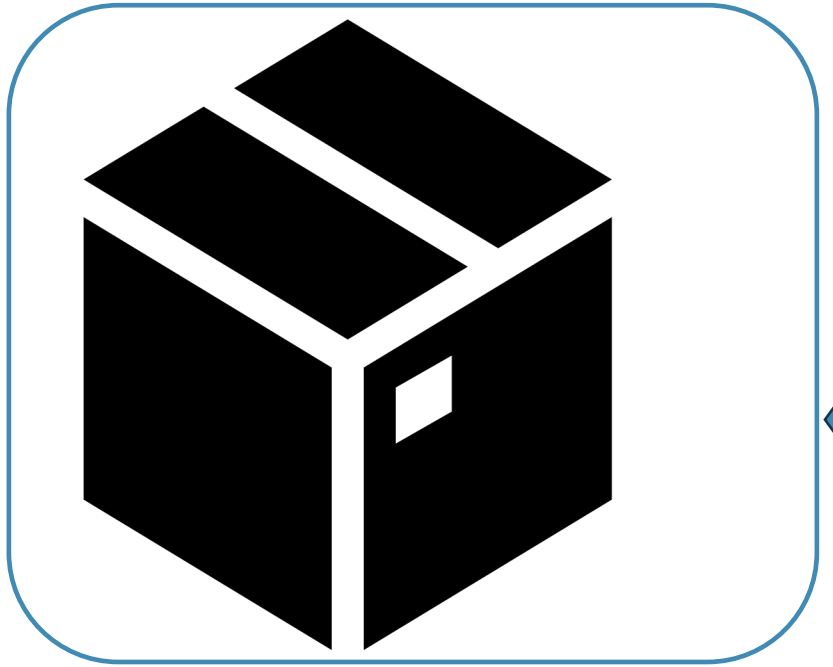
What is Ondemand Rstudio?

- It was built on top of an apptainer container image, thus, it runs on a different OS than Alpine.

- A container is a tool that enables you to run different applications that were built on different OS than Alpine.



Rstudio

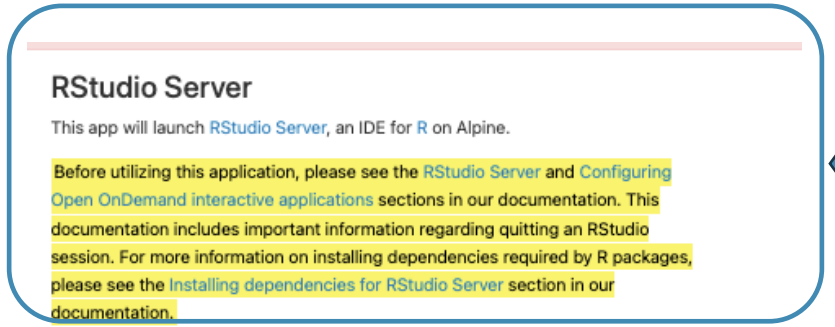


Container

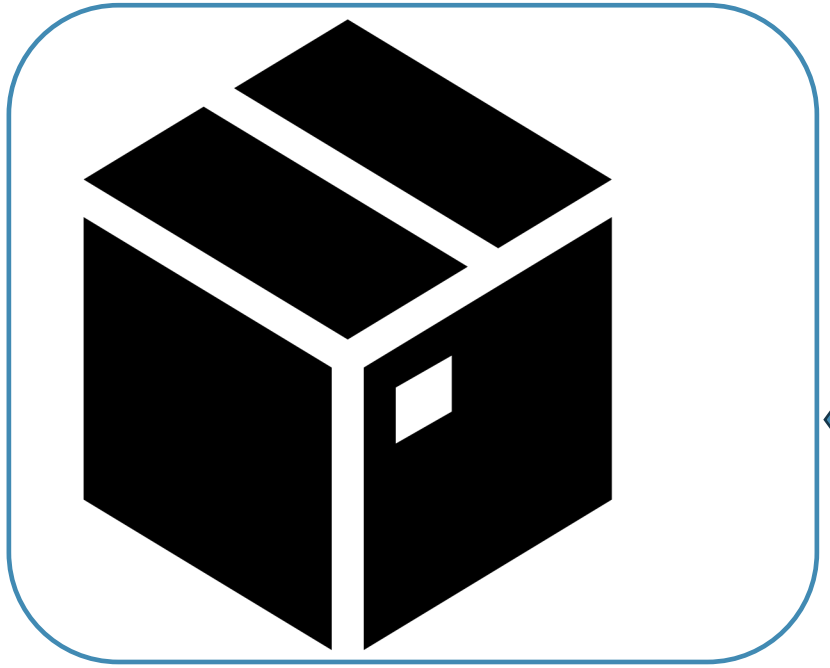
Ondemand Rstudio

- We will cover containers as an alternative way to install packages during the workshop: “How to use Containers on Alpine” on 9/29/25.

- We also have an existing guide here: https://github.com/kf-cuanschutz/CU-Anschutz-HPC-documentation/blob/main/Workshops/Container_lab_workshop_040125.pdf



Rstudio

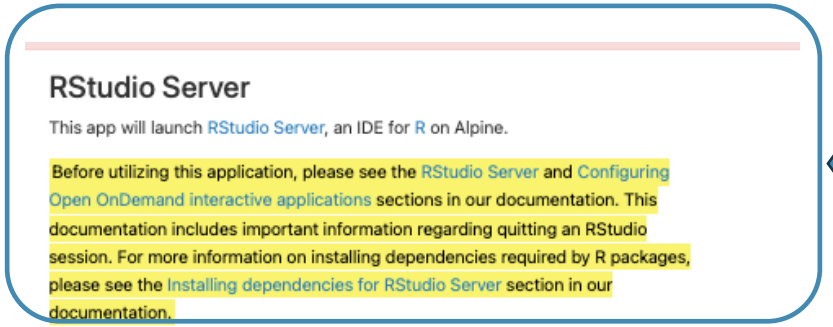


Container

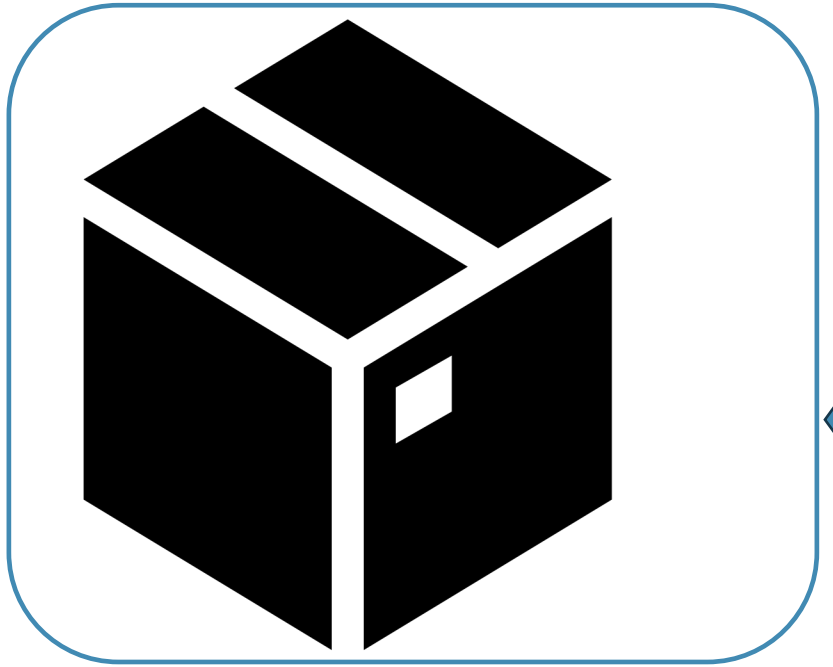
Ondemand Rstudio

- The Rstudio .sif image is located in

/curc/sw/containers/open_ondemand/rstudio-server-4.4.1.sif



Rstudio



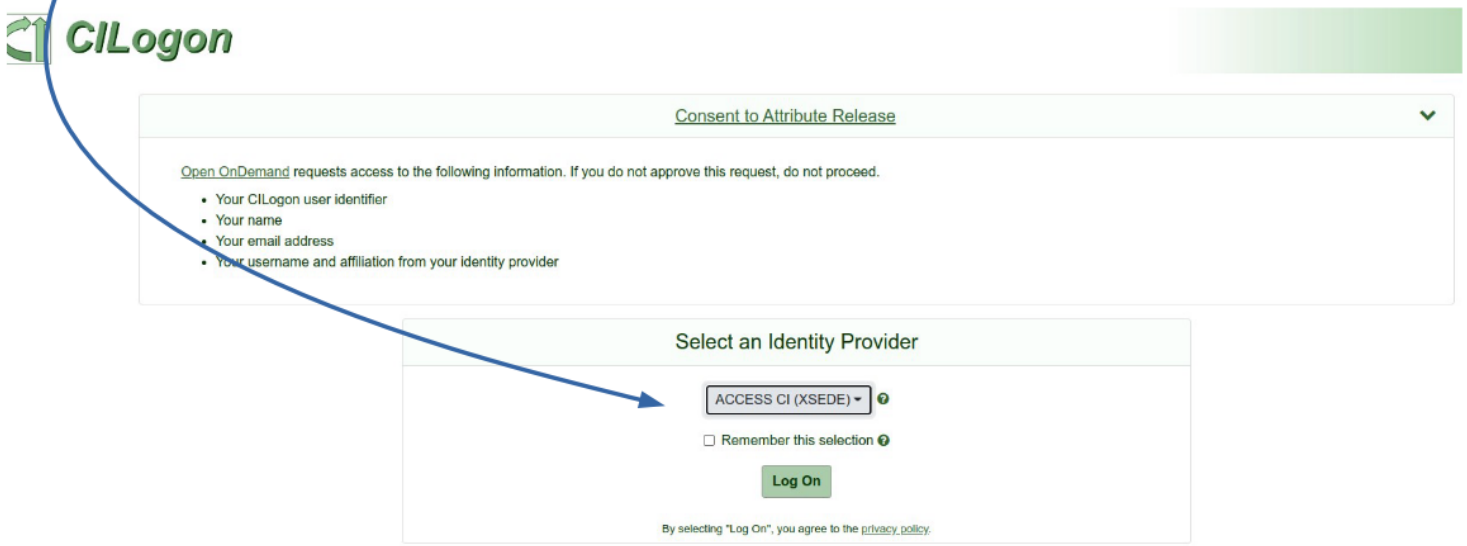
Container

How to access Rstudio?

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 provide and then press "Log On"



University of Colorado Anschutz Medical Campus

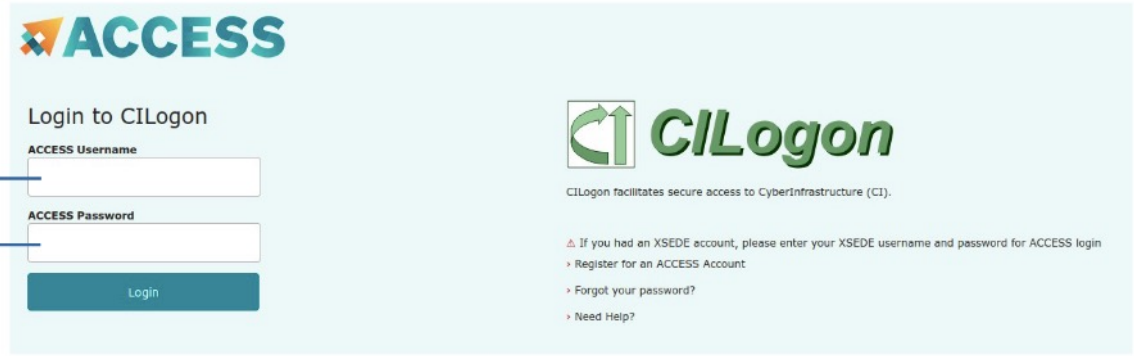
Source: https://github.com/kf-cuanschutz/CU-Anschutz-HPC-documentation/blob/main/Workshops/Alpine_Noob_Introduction_to_HPC_and_Alpine-120723.pdf

How to access Rstudio?

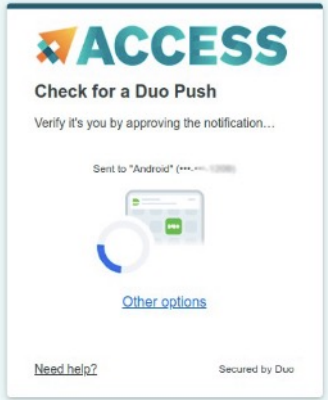
First let's login to Alpine!

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!!

ACCESS ID and ACCESS password **not** CU Anschutz credentials!!



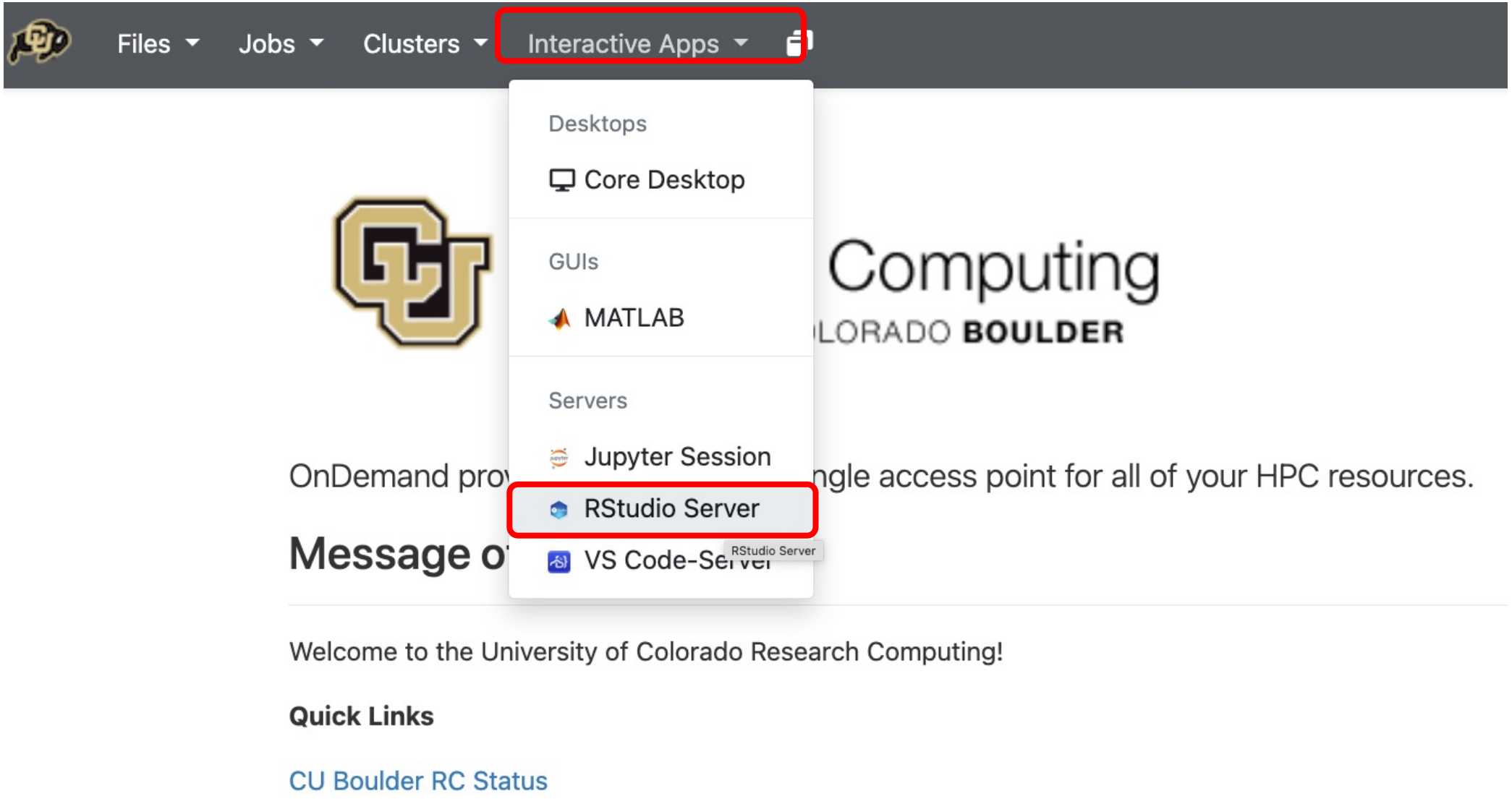
4 This will prompt a DUO MFA push to your phone or whichever way you have DUO set up on your phone to authenticate. Accept the push sent to your device.



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

How to access Rstudio?

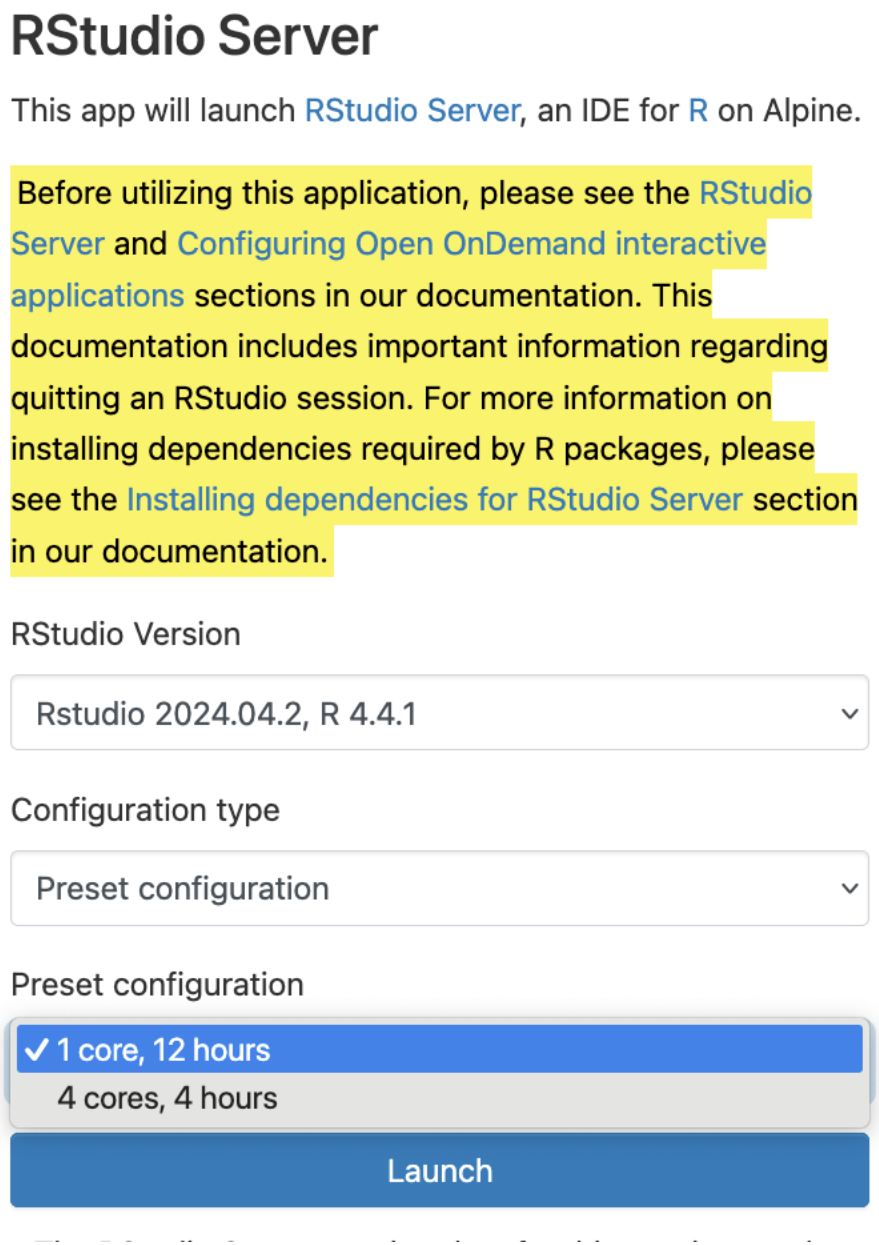


When to use R with Ondemand?

Use Rstudio on Alpine if:

- If you are used to Rstudio outside of Alpine
- If you need to display interactive plots.
- If you think that most of your pipelines will not require more than 16 cores or 60GB of RAM.

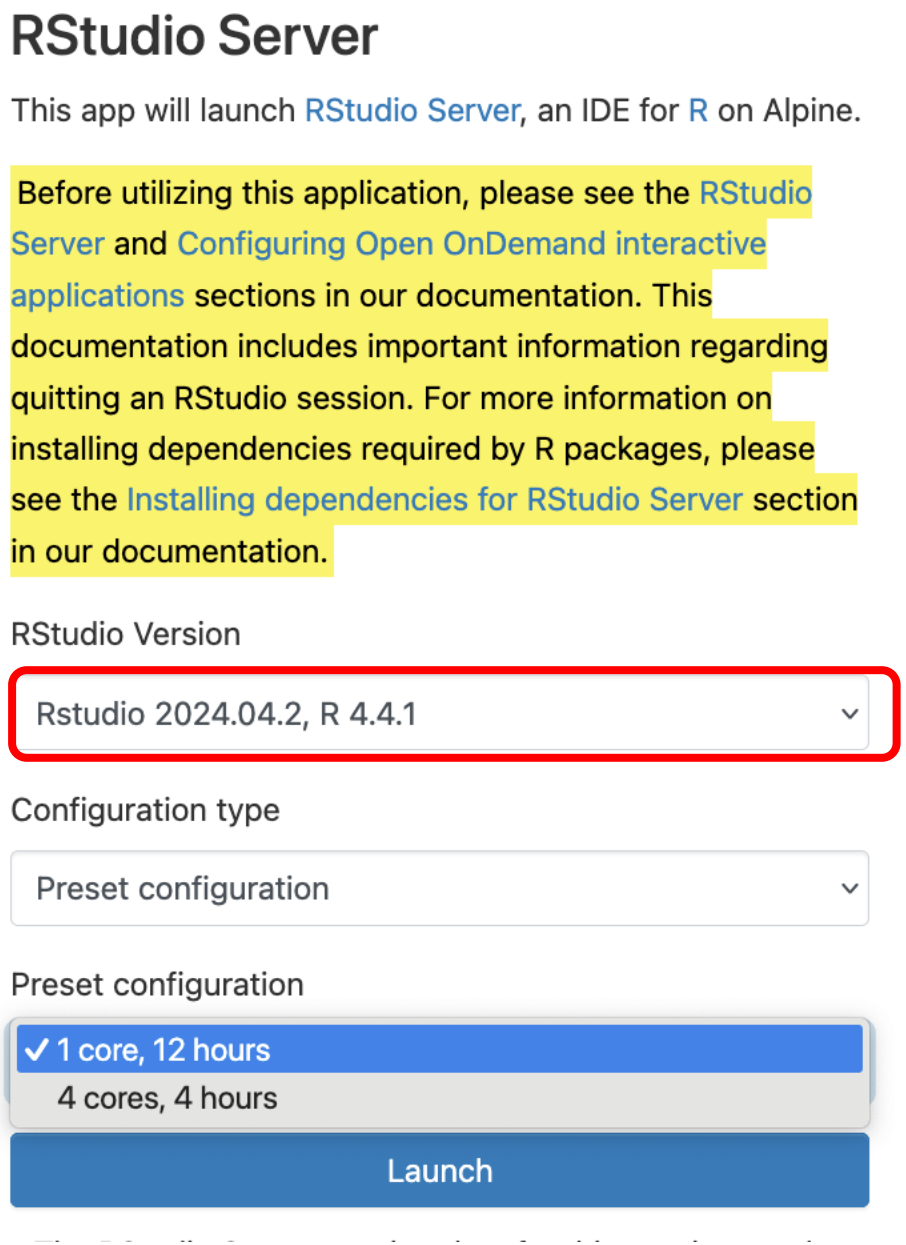
University of Colorado
Anschutz Medical Campus



When to use R with Ondemand?

2 versions of Rstudio available:

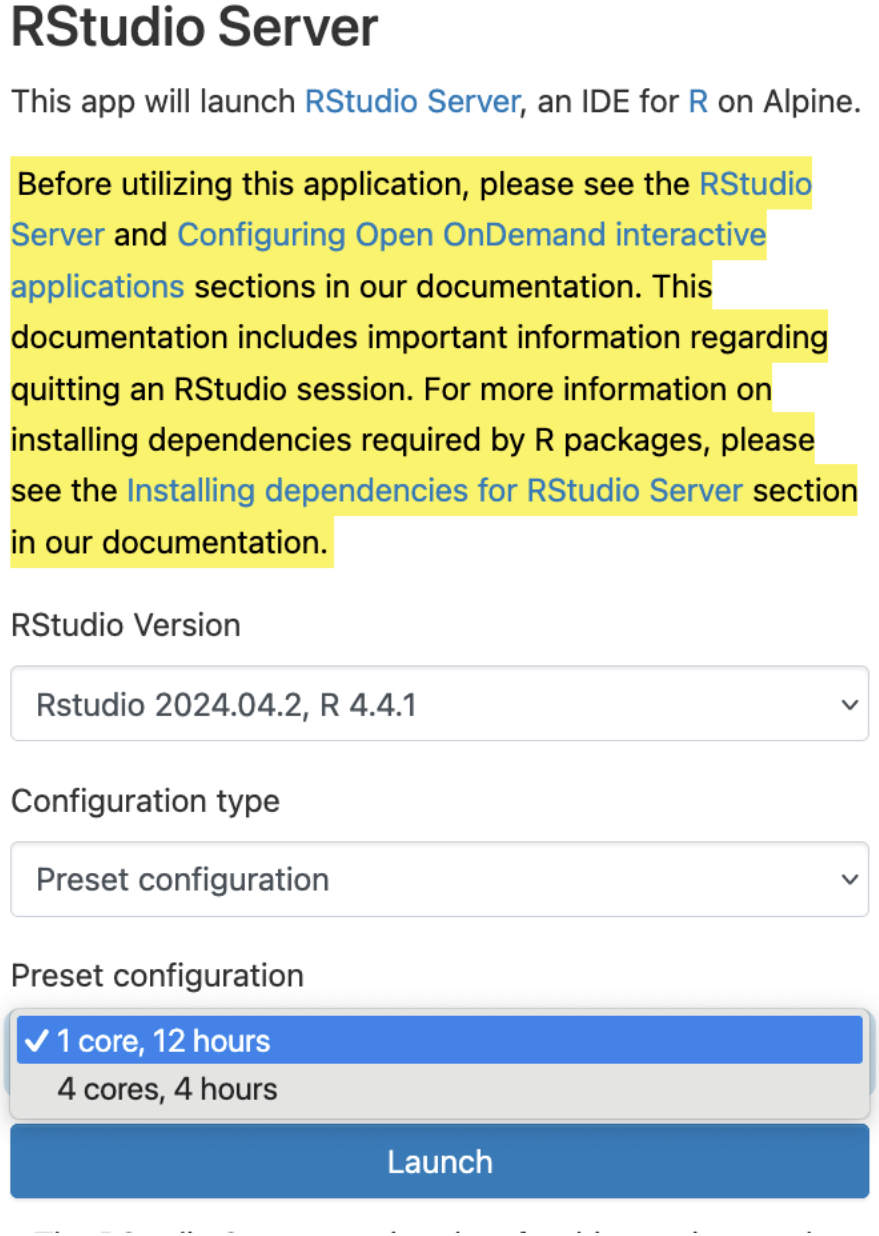
- R v4.2.2
- R v4.4.1
- We strongly recommend to use R v4.4.1!



When to use R with Ondemand?

2 resource configurations possible:

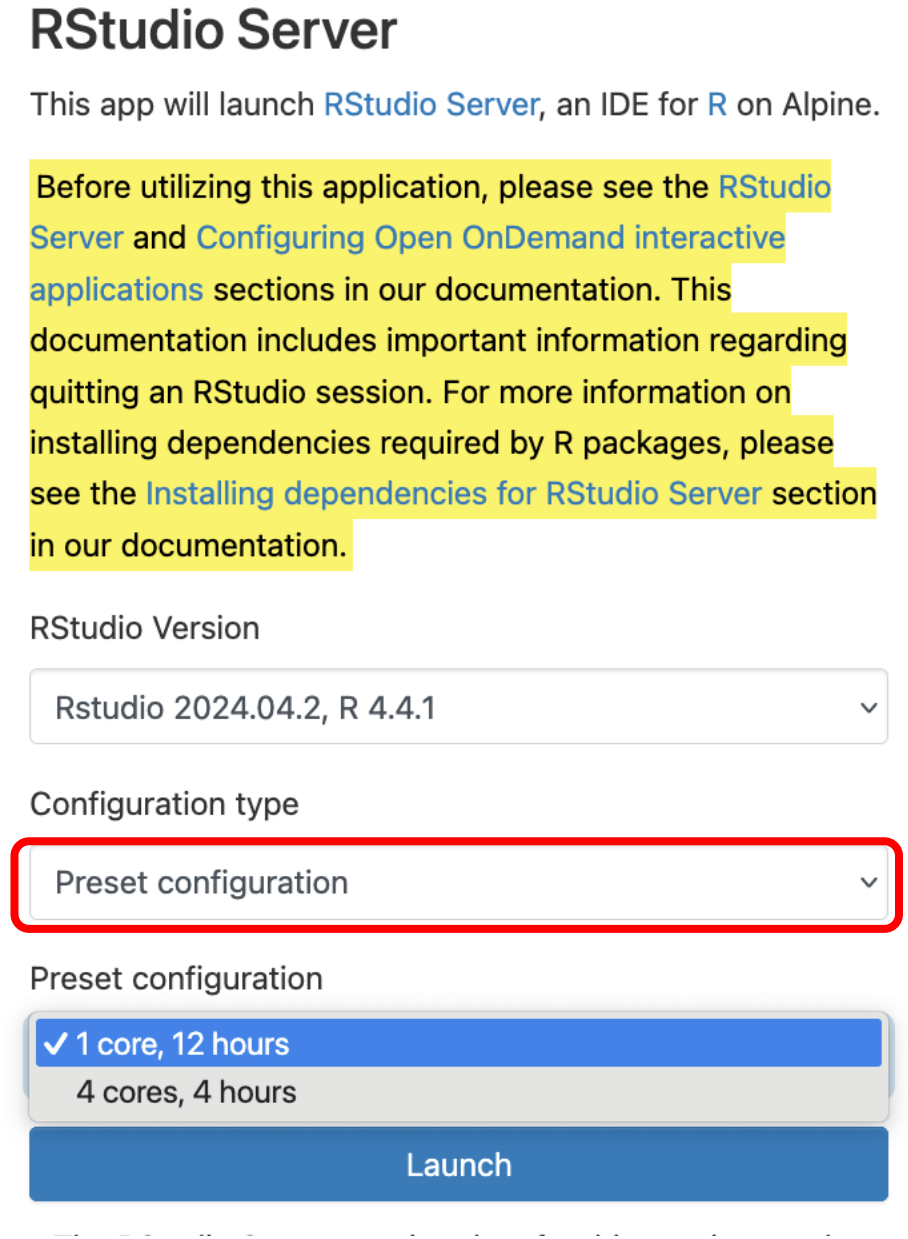
- Preset configuration
- Custom configuration



When to use R with Ondemand?

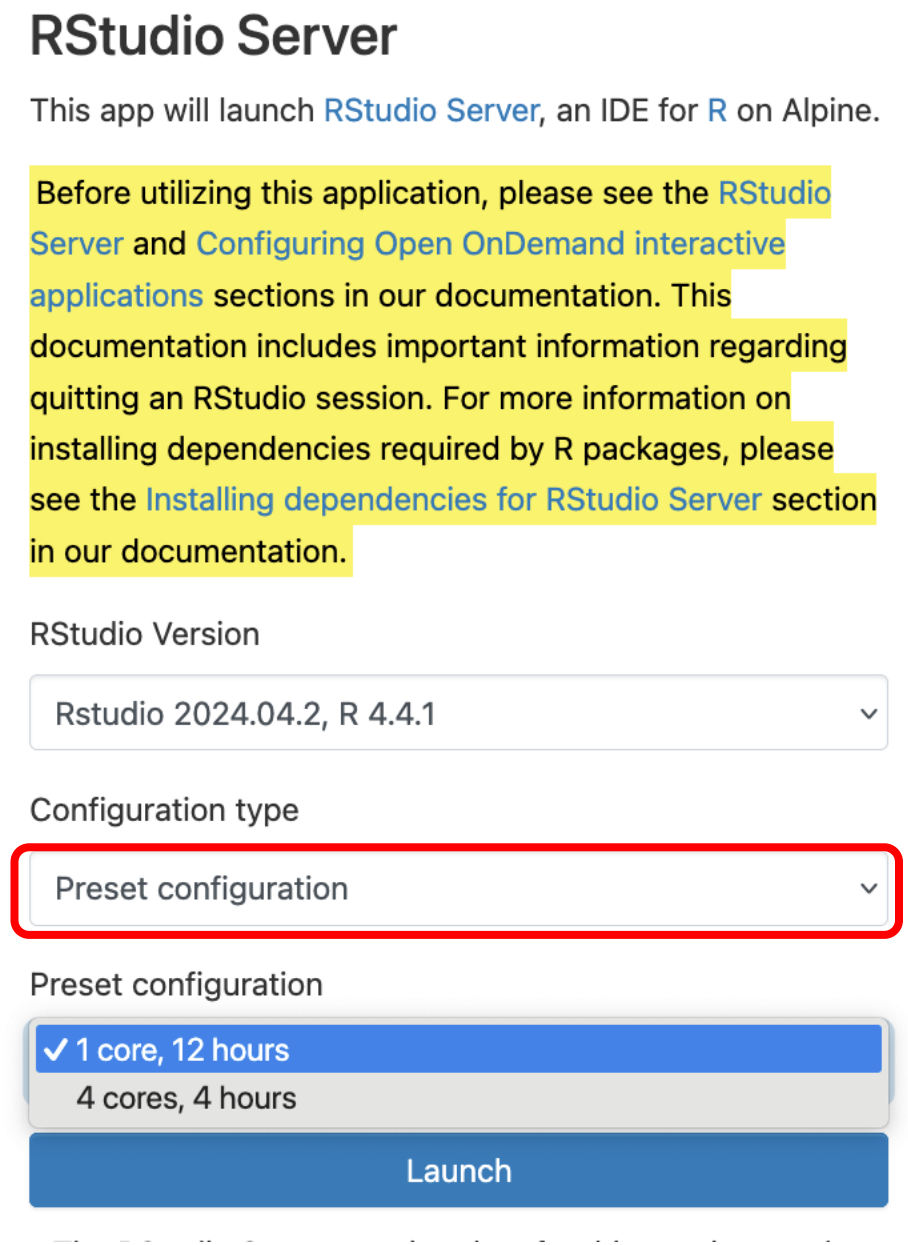
2 resource configurations possible:

- Preset configuration
- Custom configuration



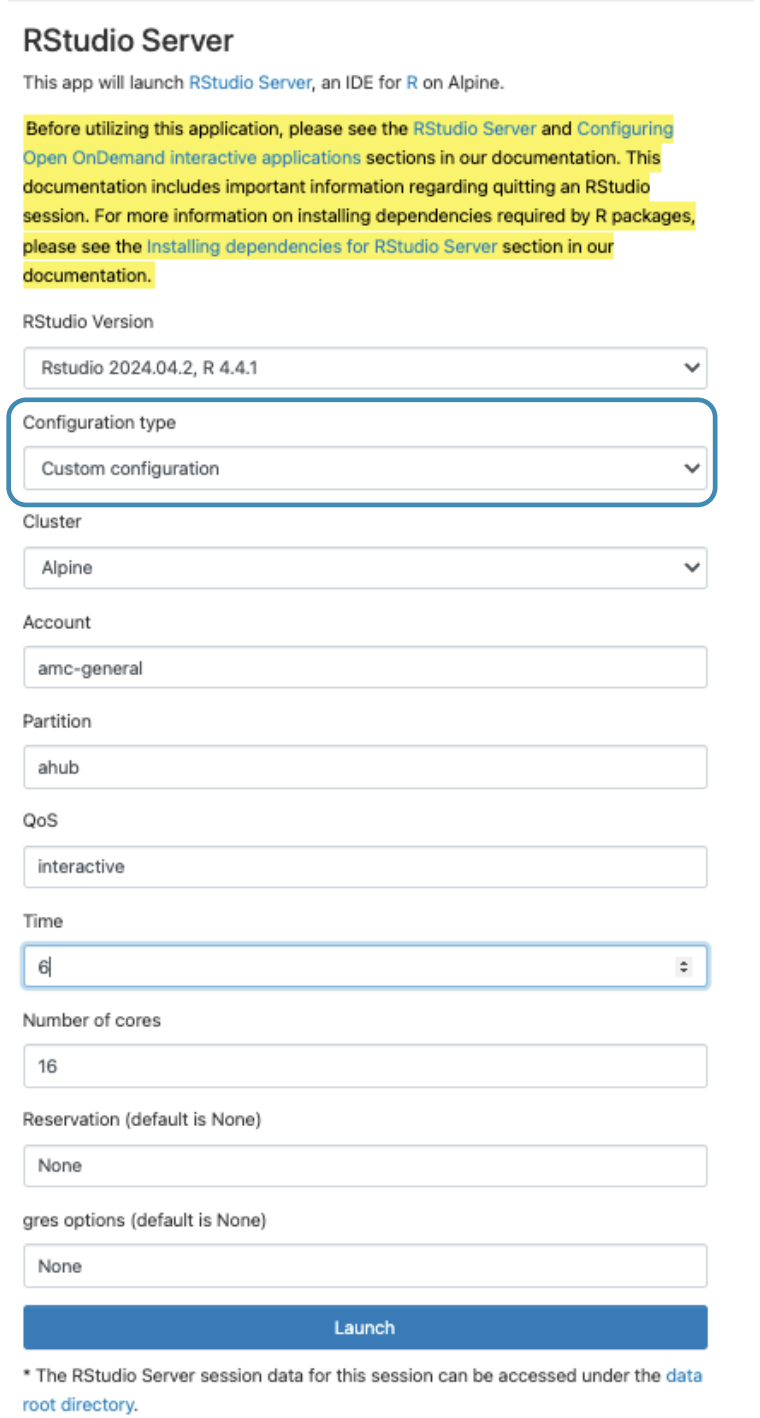
Preset configuration

- Between 1 and 4 cores.
- Up to 12 hours.
- Choose this configuration if you are still new to Alpine and you do not need more resources.



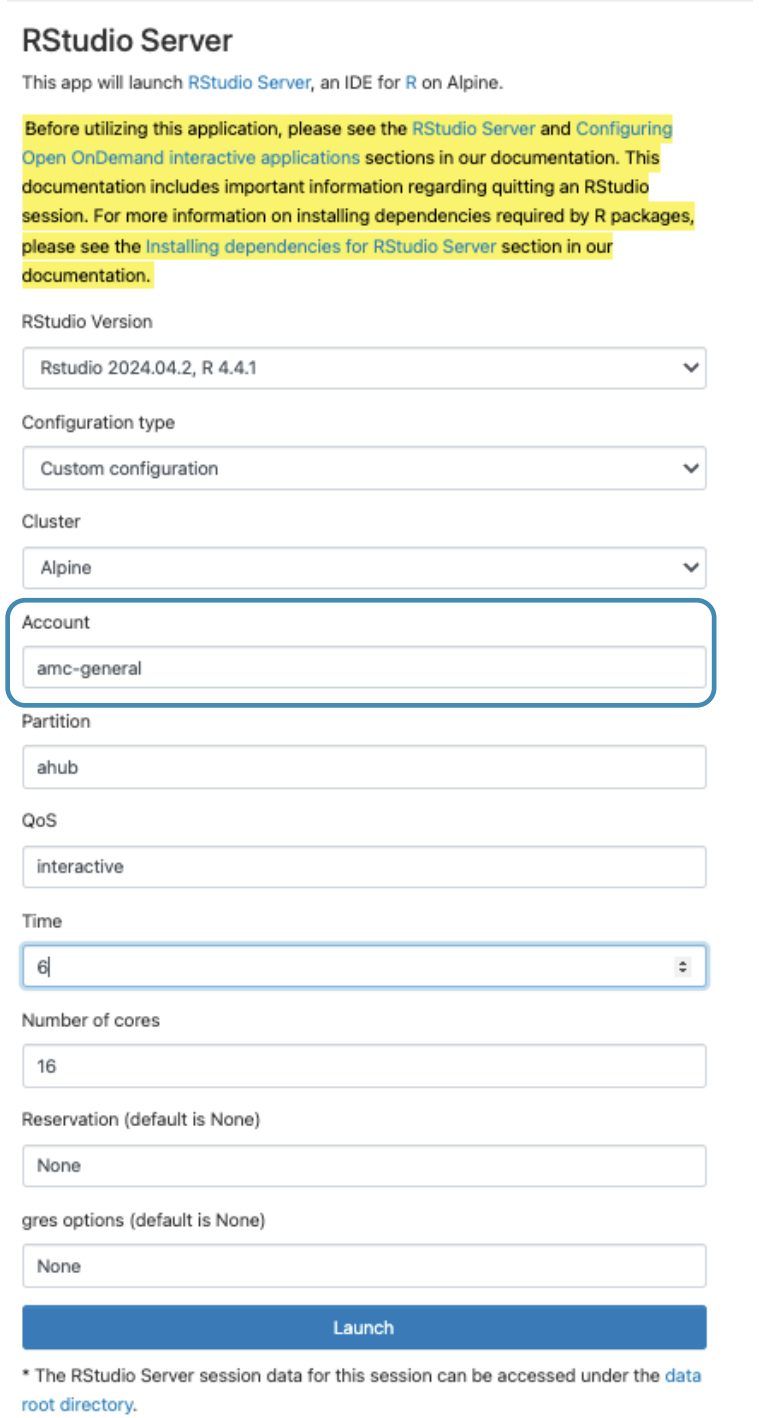
Custom configuration

To access it, select “Custom configuration” under “ Configuration type”.



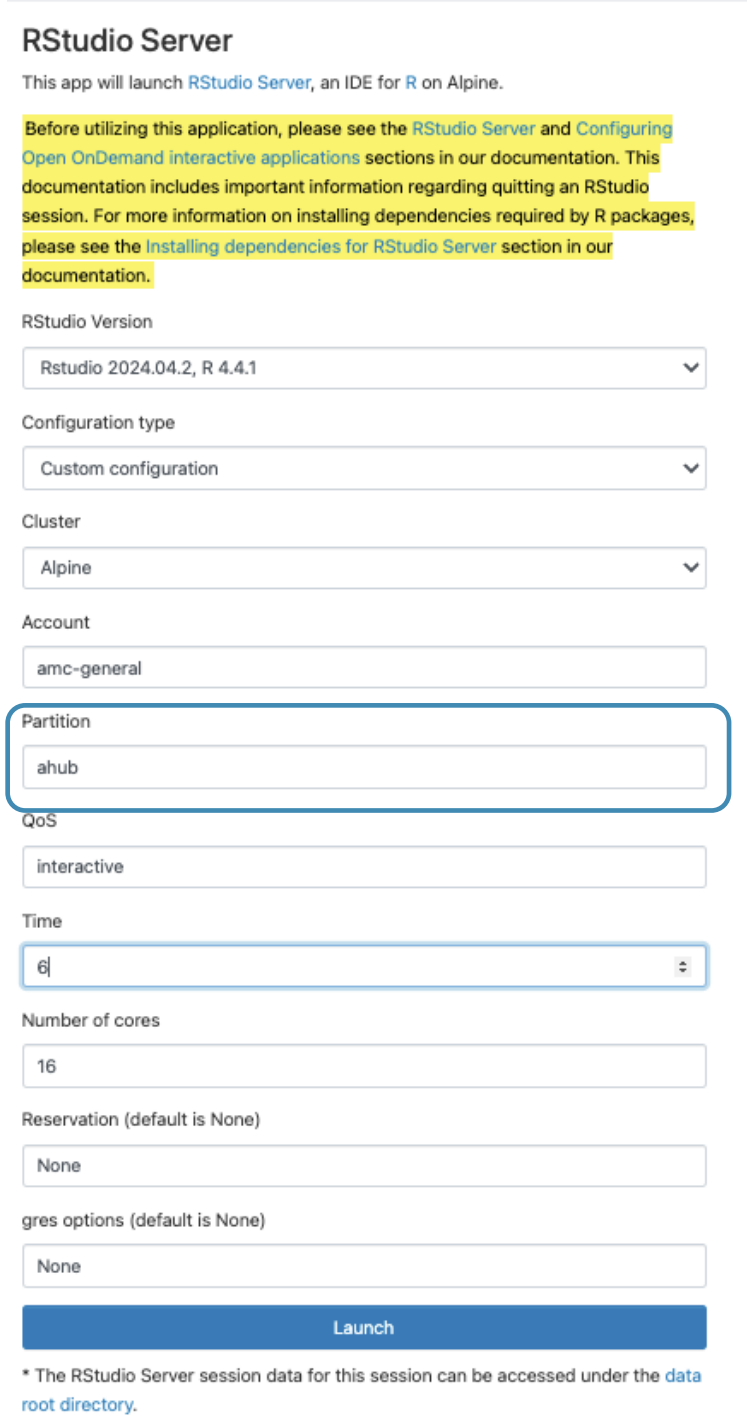
Custom configuration

- Make sure that your account is amc-general!
- If the launch fails due to an incorrect account, please email HPCSupport@cuanschutz.edu .



Custom configuration

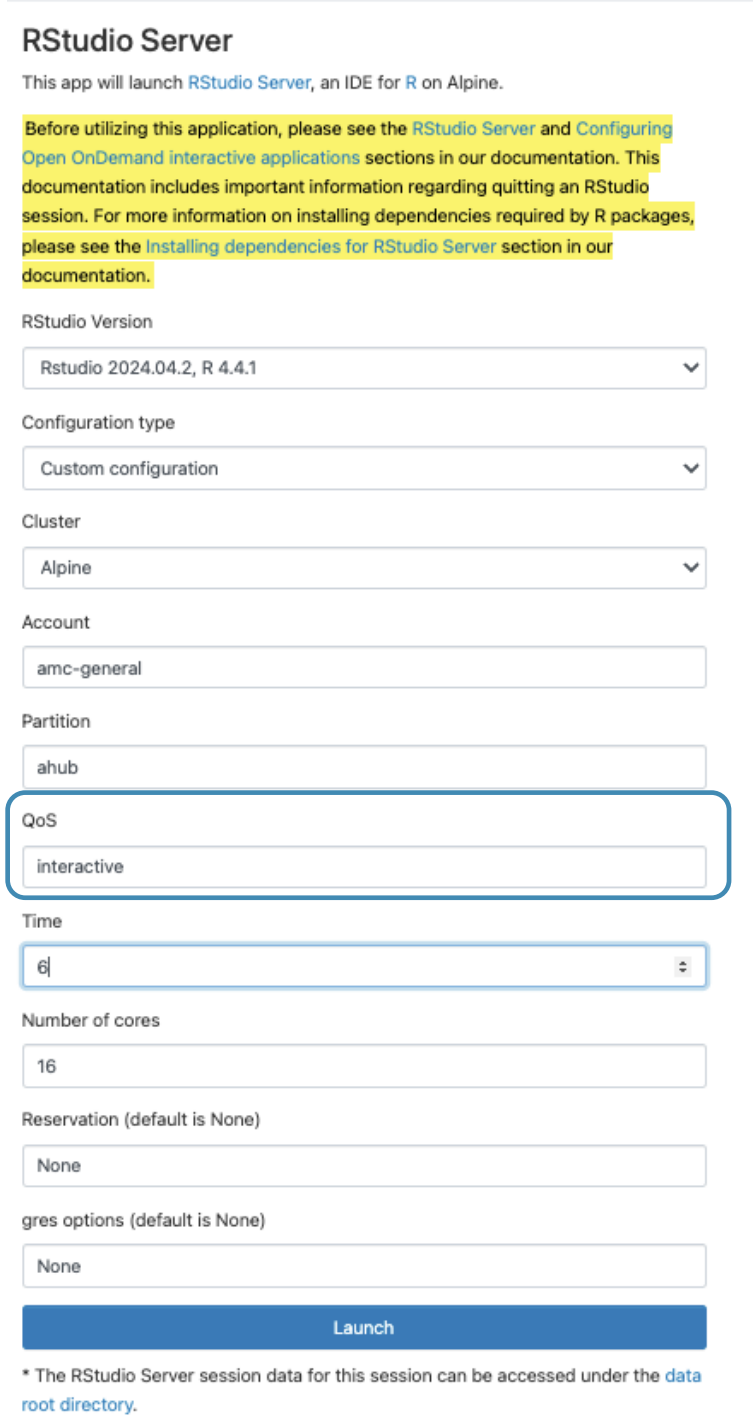
- Always select ahub, which is the correct Ondemand partition.



Custom configuration

- The only QOS that works with the ahub partition is “interactive”.
- For more information on QOS please refer to this workshop:

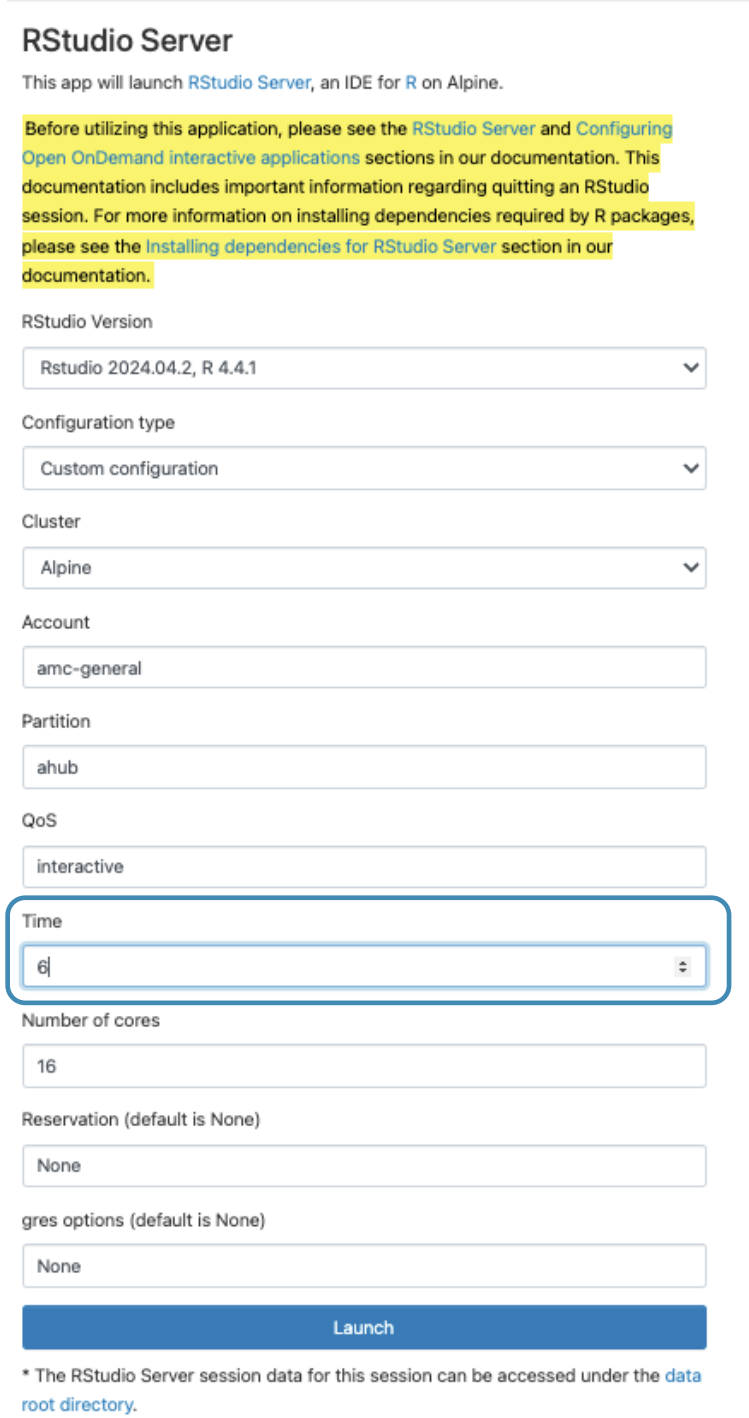
https://github.com/kf-cuanschutz/CU-Anschutz-HPC-documentation/blob/main/Workshops/Introduction_to_Alpine_workshop_09092025_v3.pdf



University of Colorado
Anschutz Medical Campus

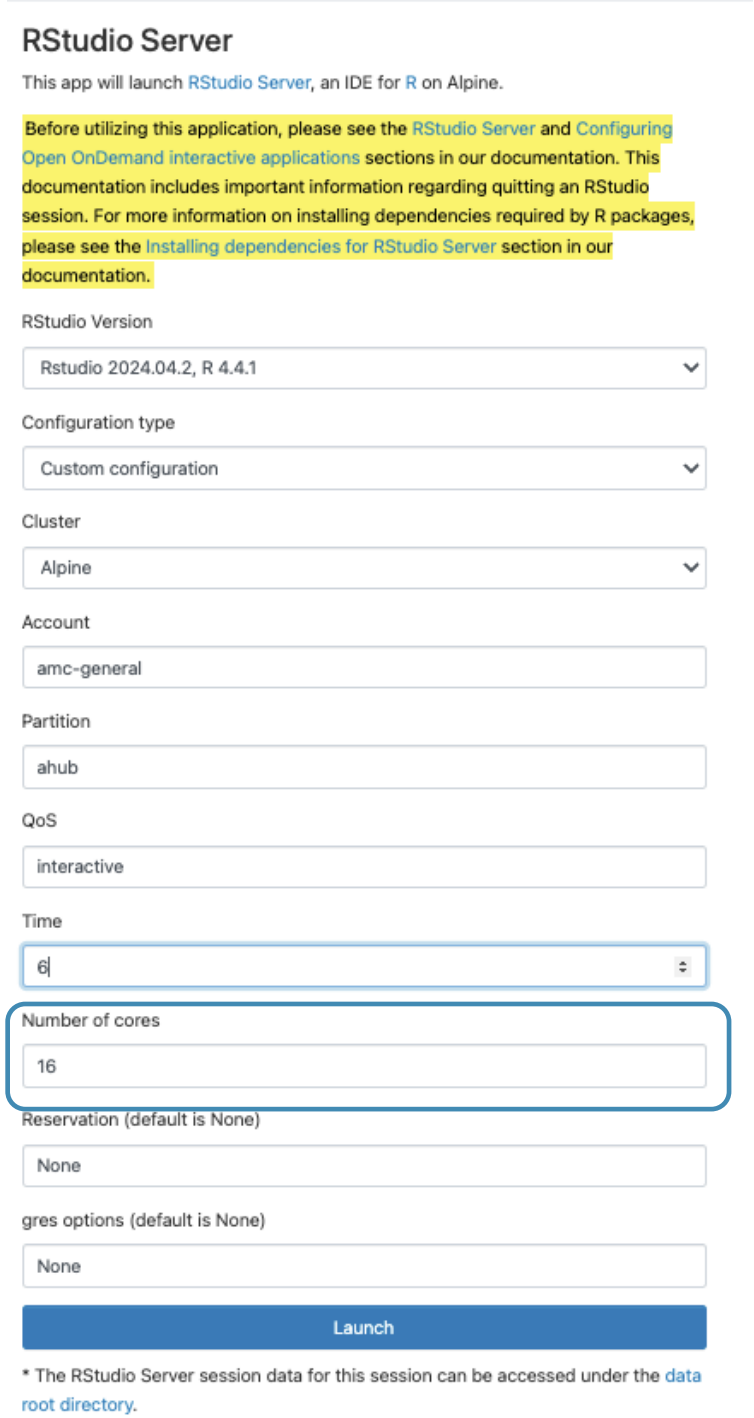
Custom configuration

- Up to 12 hours walltime.
- However, wait times should be nonexistent.



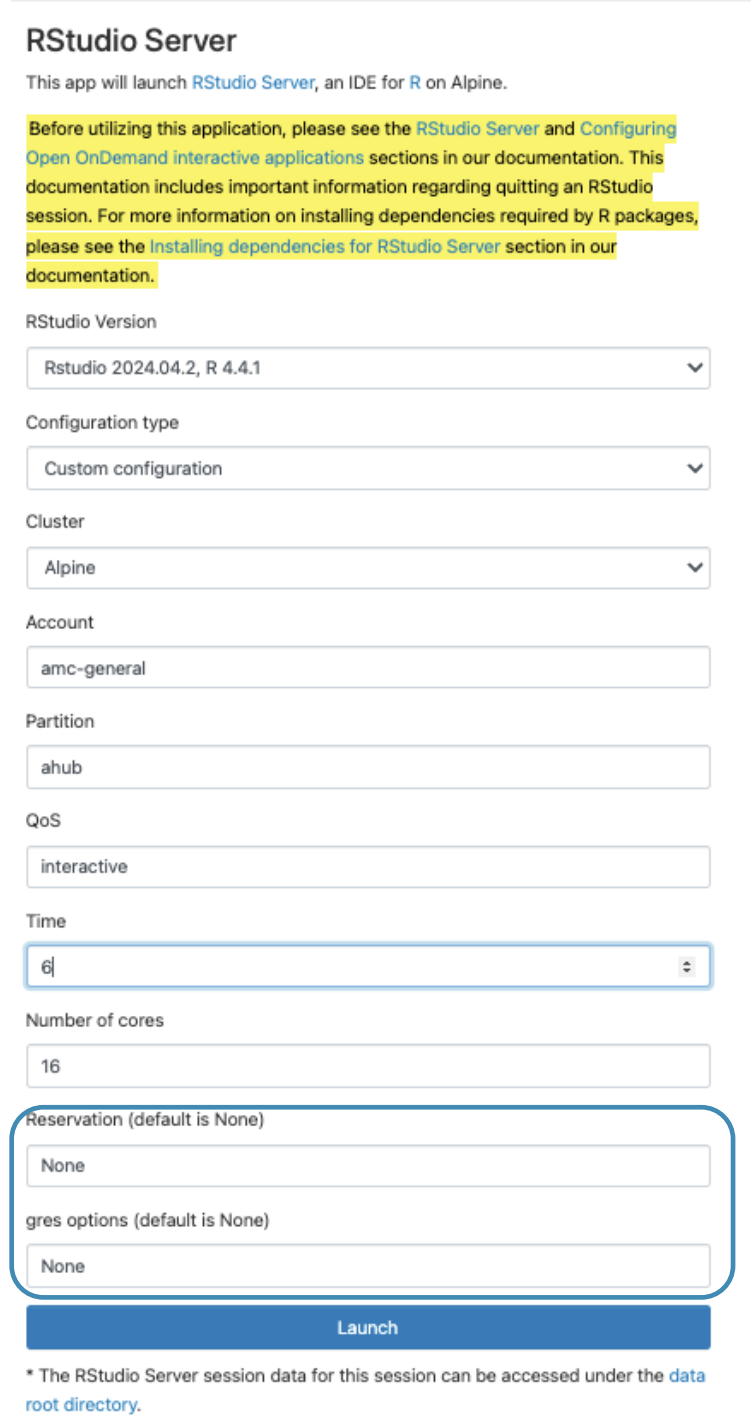
Custom configuration

- Up to 16 cores maximum.
- This will translate to roughly 60G mem
- 3.75G-per-core X 16 cores = 60 G



Custom configuration

Reservation and gres do not apply here.



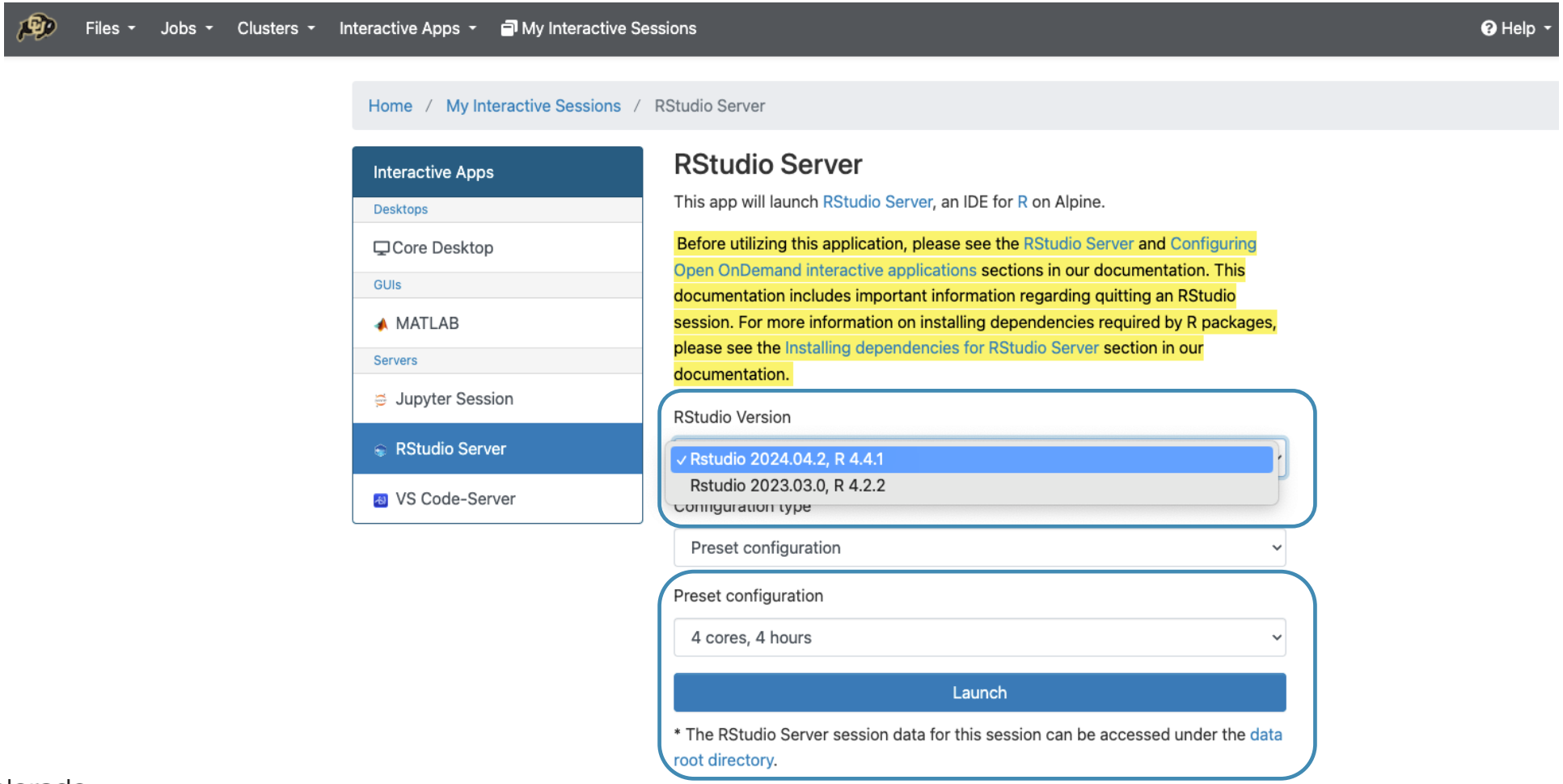
Rstudio library dependencies

- Because Rstudio on Alpine is distributed through a container, it does not contain all the dependencies.

- Let’s review how to install most library dependencies needed for an AMC user on Alpine.

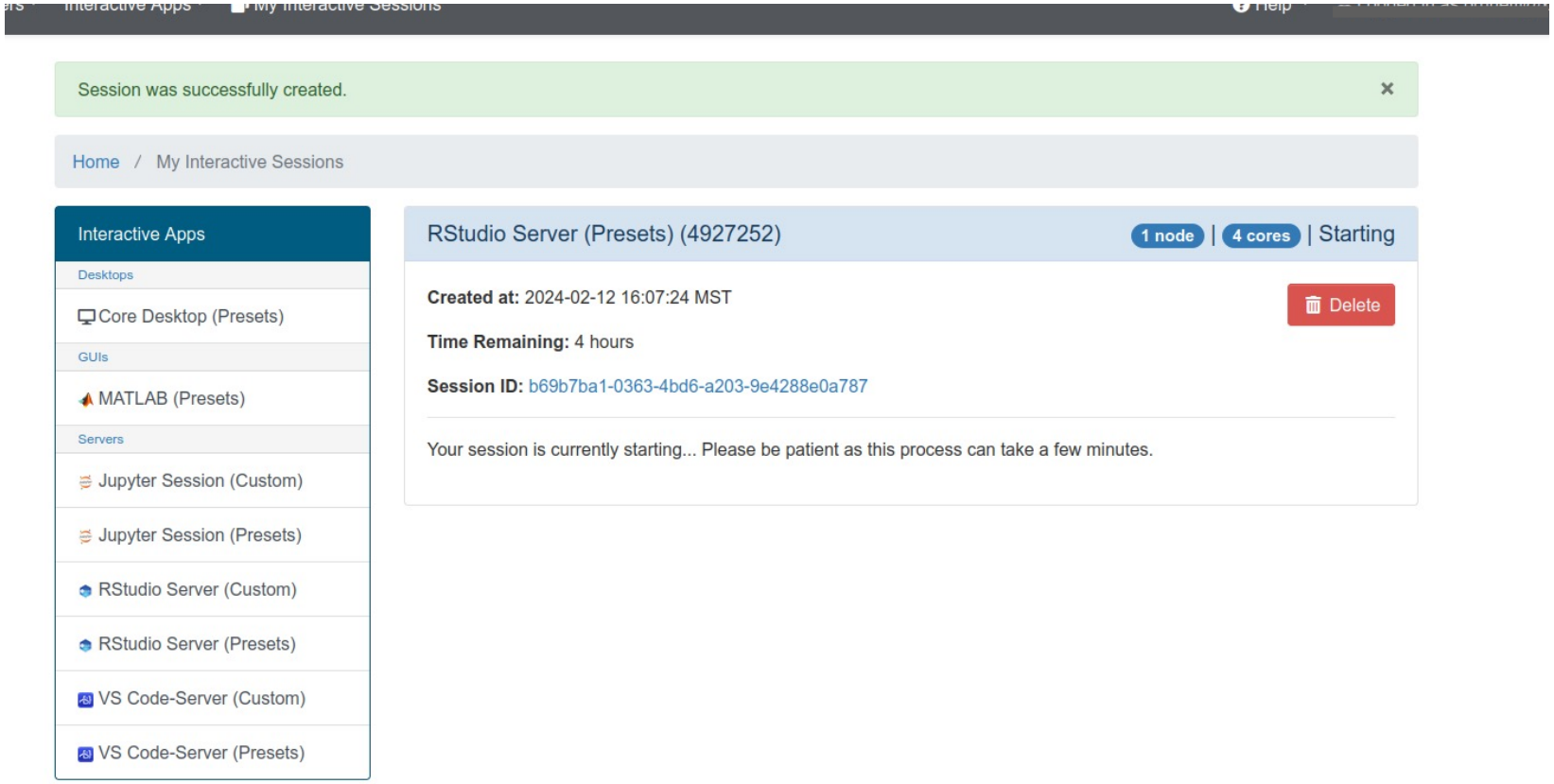
Rstudio library dependencies

- Select Rv4.4.1, 4 cores and finally launch a session.



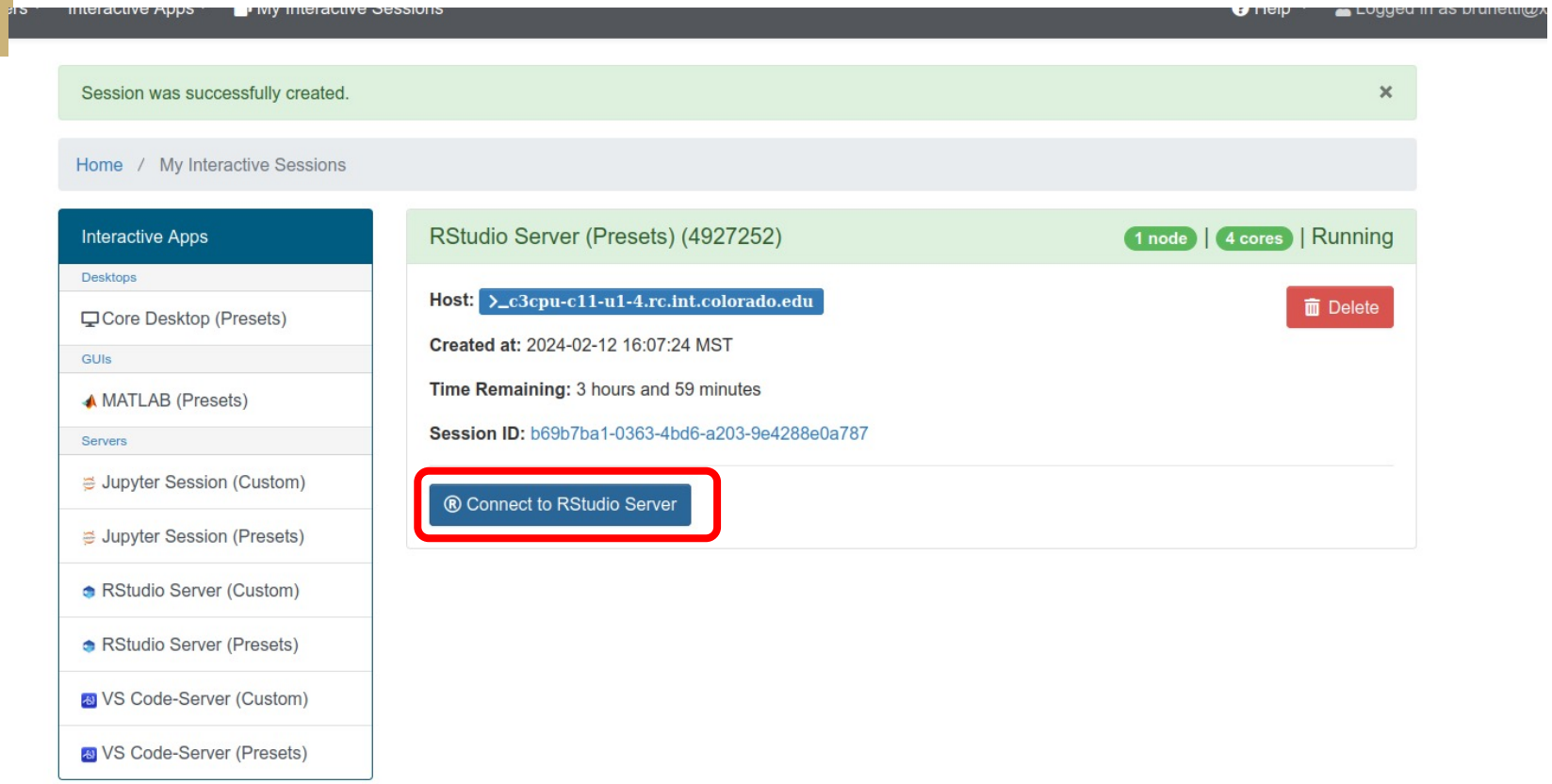
Rstudio library dependencies

- The screen below indicates that your session is starting.



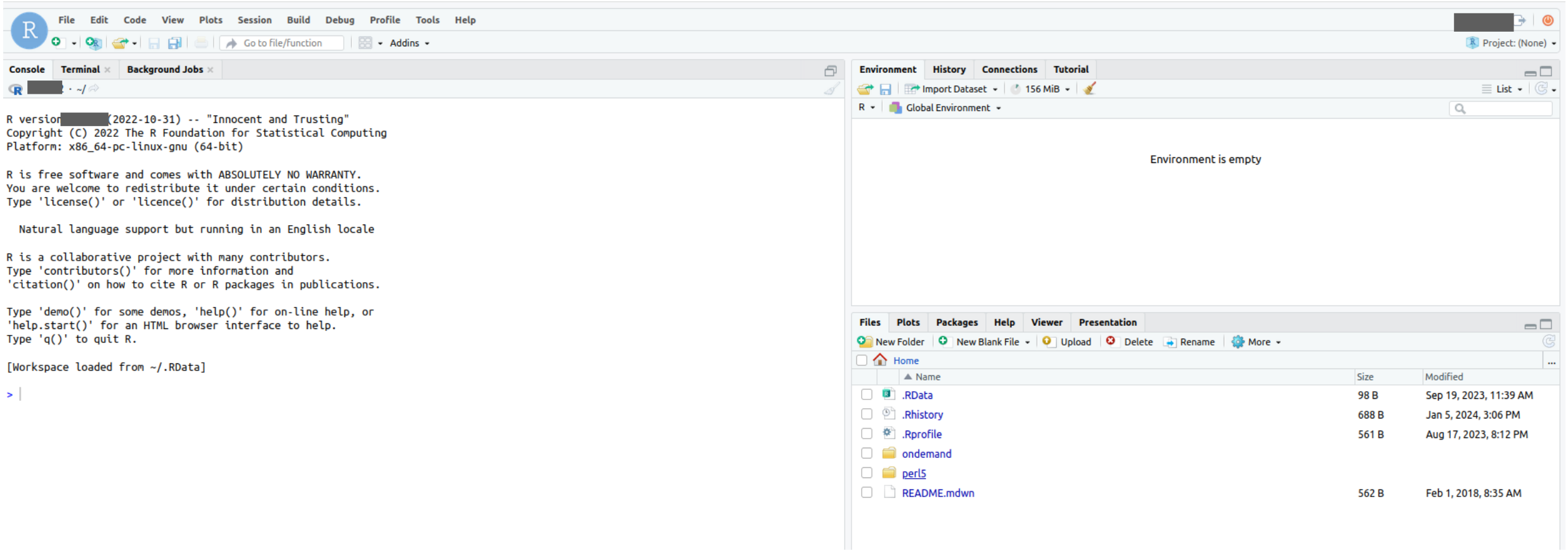
Rstudio library dependencies

- Select “Connect to Rstudio Server”.



Rstudio library dependencies

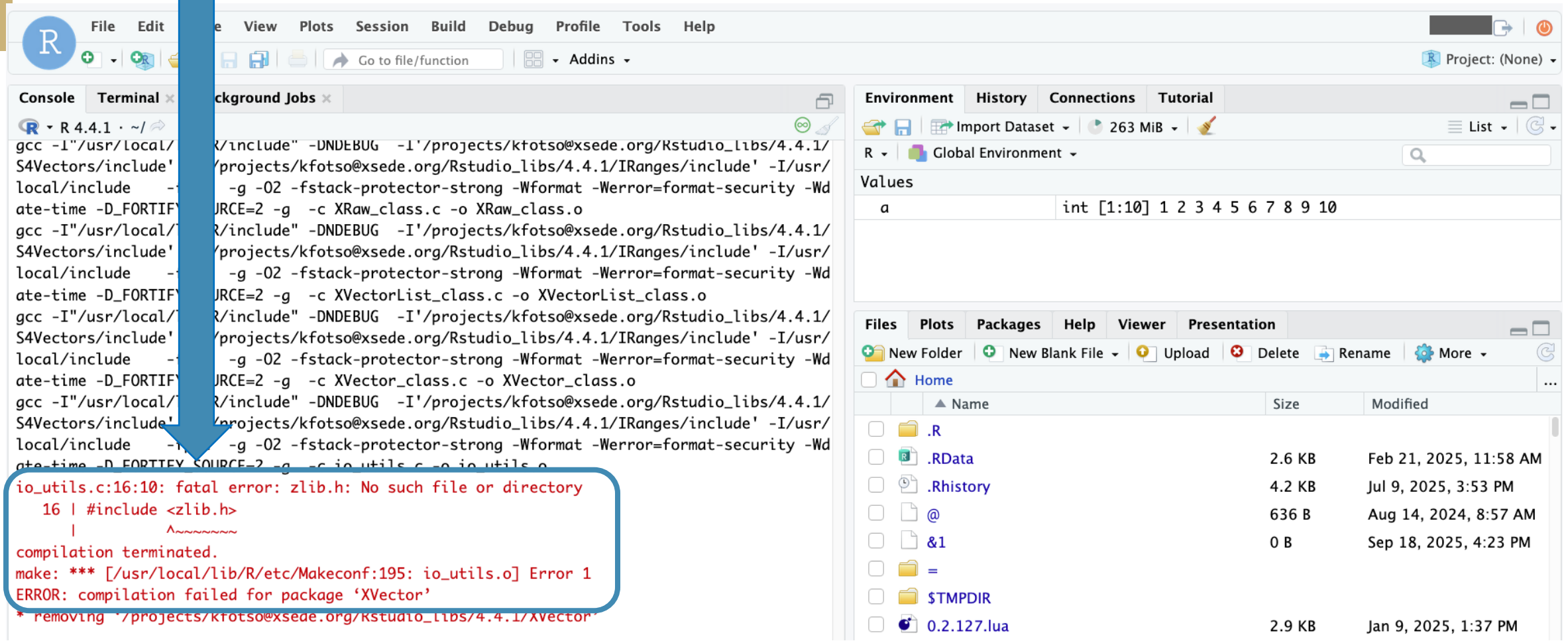
- You are now logged in!
- You have a functionial Rstudio session.



University of Colorado
Anschutz Medical Campus

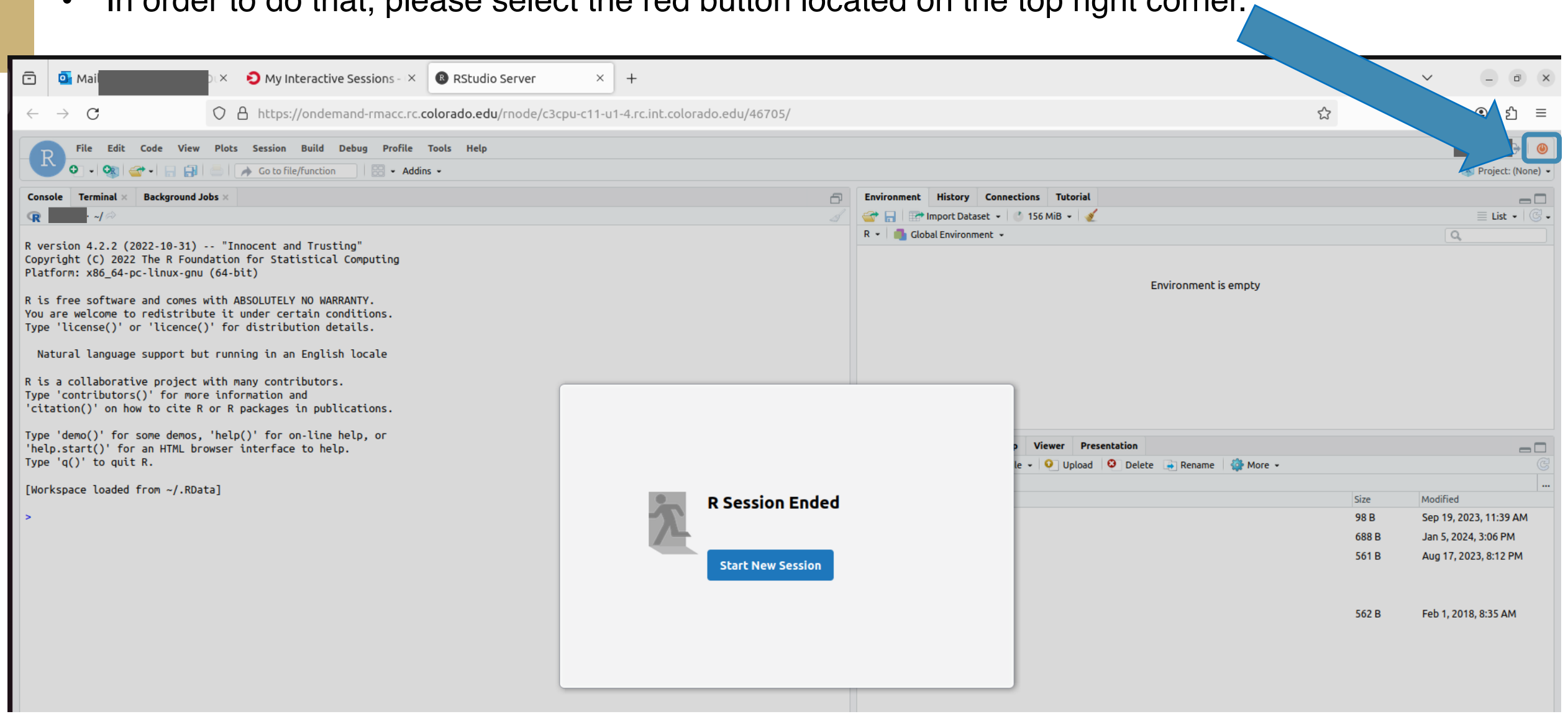
Rstudio library dependencies

- However, all the library dependencies are not installed yet (e.g. zlib.h).
- In order to achieve that, we need to install them directly inside the container!



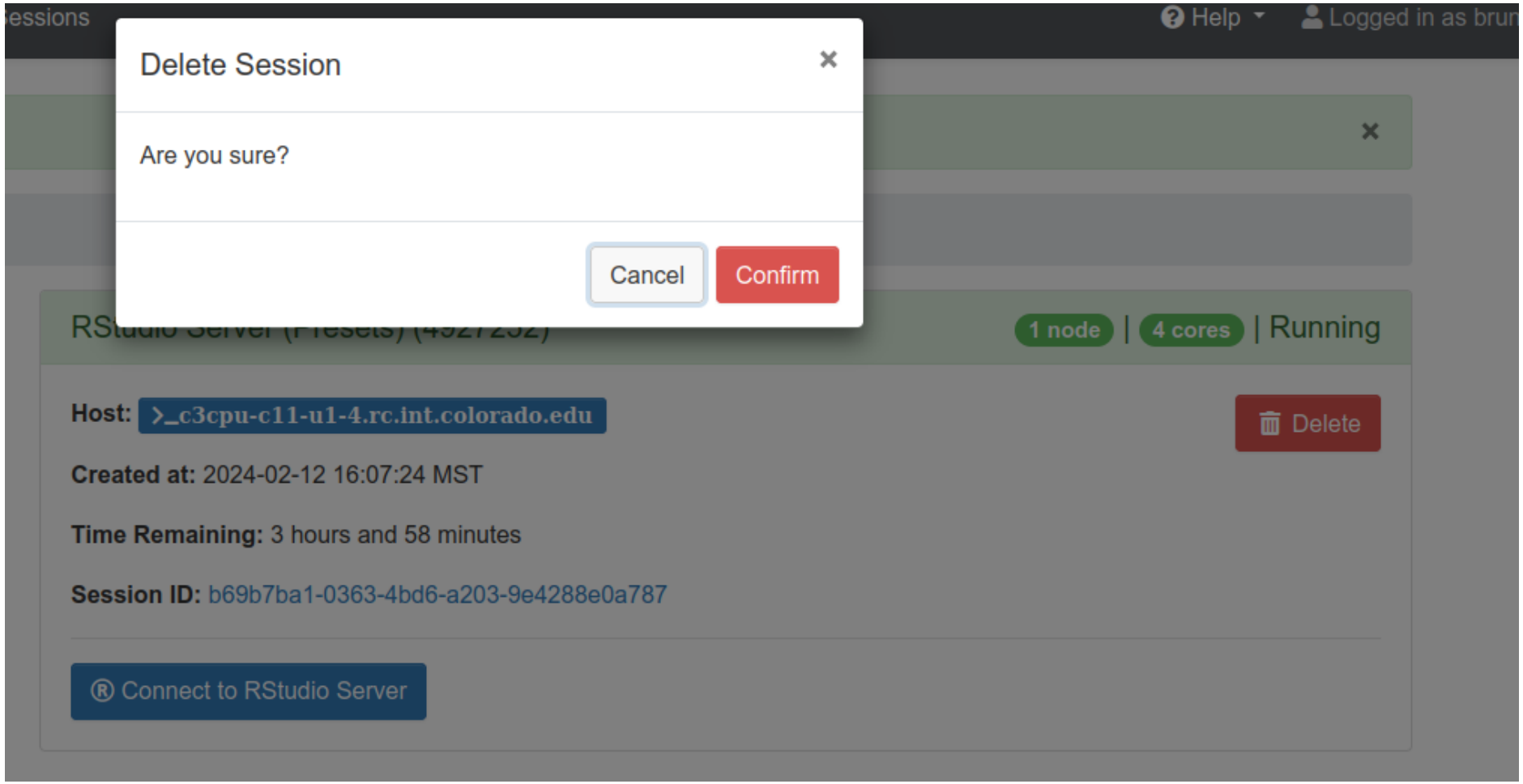
Rstudio library dependencies

- You need to first exit the Rstudio session.
- In order to do that, please select the red button located on the top right corner.



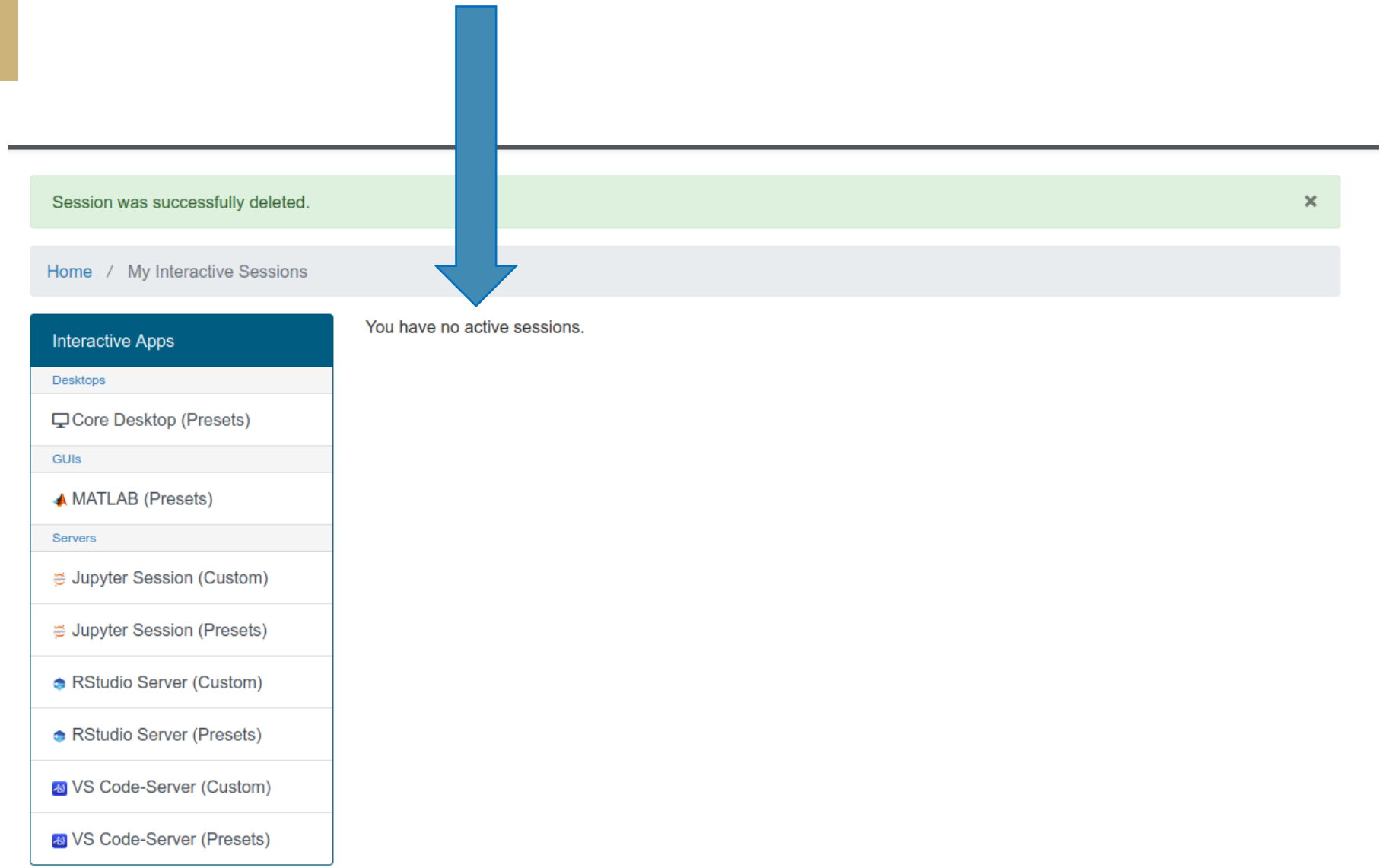
Rstudio library dependencies

- Now go back to your interactive session tab browser.
- Make sure to press the red Delete button to end your rstudio session request.
- When asked to confirm, hit the red "Confirm" button.



Rstudio library dependencies

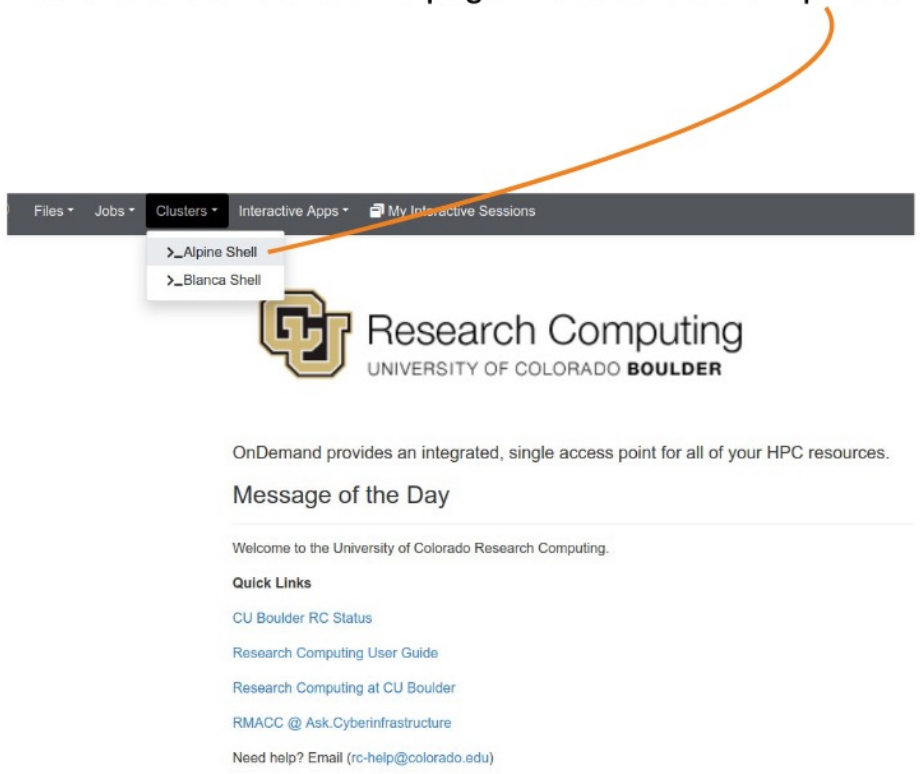
- You can confirm that you are logged out of Rstudio and have killed your RStudio session by seeing the message below.



Open an Alpine shell

First let's login to Alpine!

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

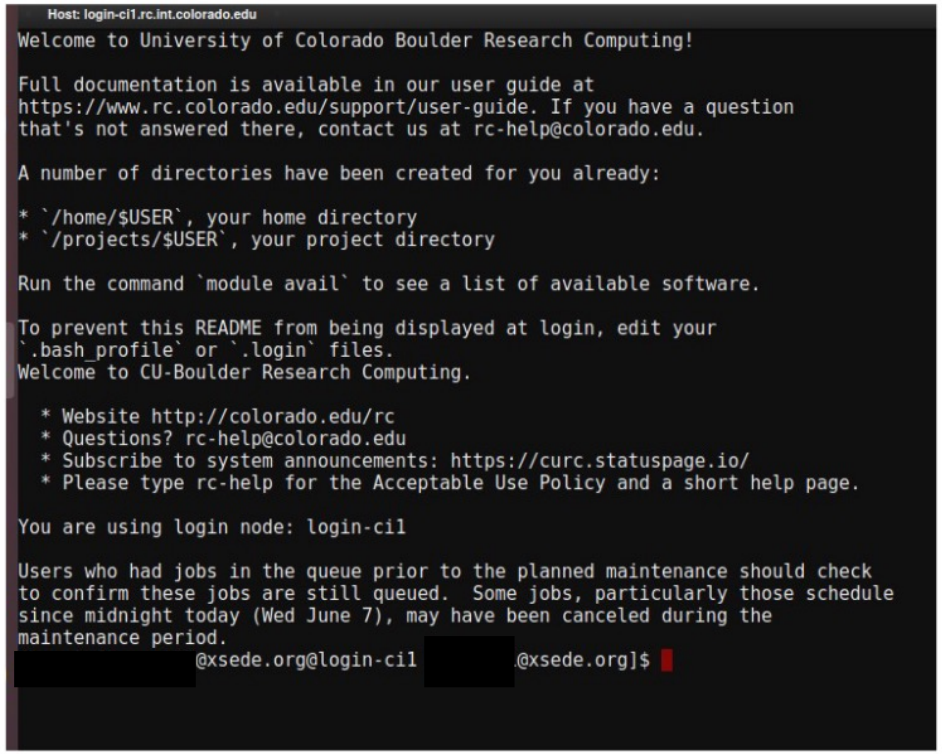


Source: https://github.com/kf-cuanschutz/CU-Anschutz-HPC-documentation/blob/main/Workshops/Alpine_Noob_Introduction_to_HPC_and_Alpine-120723.pdf

Open an Alpine shell

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.



Source: https://github.com/kf-cuanschutz/CU-Anschutz-HPC-documentation/blob/main/Workshops/Alpine_Noob_Introduction_to_HPC_and_Alpine-120723.pdf

Access to 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 access the container.

Rstudio library dependencies

- Change directories into /scratch/alpine/$USER

```
|               @c3cpu-a2-u32-3 ~]$ cd /scratch/alpine/$USER
|               @c3cpu-a2-u32-3 [redacted]$ git clone https://github.com/kf-cuanschutz/Rstudio_repo_related_files.git
Cloning into 'Rstudio_repo_related_files'...
remote: Enumerating objects: 8, done.
remote: Counting objects: 100% (8/8), done.
remote: Compressing objects: 100% (7/7), done.
remote: Total 8 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (8/8), 2.33 KiB | 477.00 KiB/s, done.
|               @c3cpu-a2-u32-3 [redacted]@xsede.org]$ cd Rstudio_repo_related_files/
|               @c3cpu-a2-u32-3 Rstudio_repo_related_files]$ ls -la
total 68
drwxr-sr-x   3                                    4096 Sep 21 17:22 .
drwxrws--- 484                                   32768 Sep 21 17:22 ..
drwxr-sr-x   8                                    4096 Sep 21 17:22 .git
-rw-r--r--   1                                      91 Sep 21 17:22 README.md
-rw-r--r--   1                                     508 Sep 21 17:22 step0_modify_overlay.sh
-rw-r--r--   1                                     509 Sep 21 17:22 step0_r_v4_2_2_modify_overlay.sh
-rw-r--r--   1                                     550 Sep 21 17:22 step1_install_os_deps.sh
[redacted]@xsede.org@c3cpu-a2-u32-3 Rstudio_repo_related_files]$
```

University of Colorado
Anschutz Medical Campus

Rstudio library dependencies

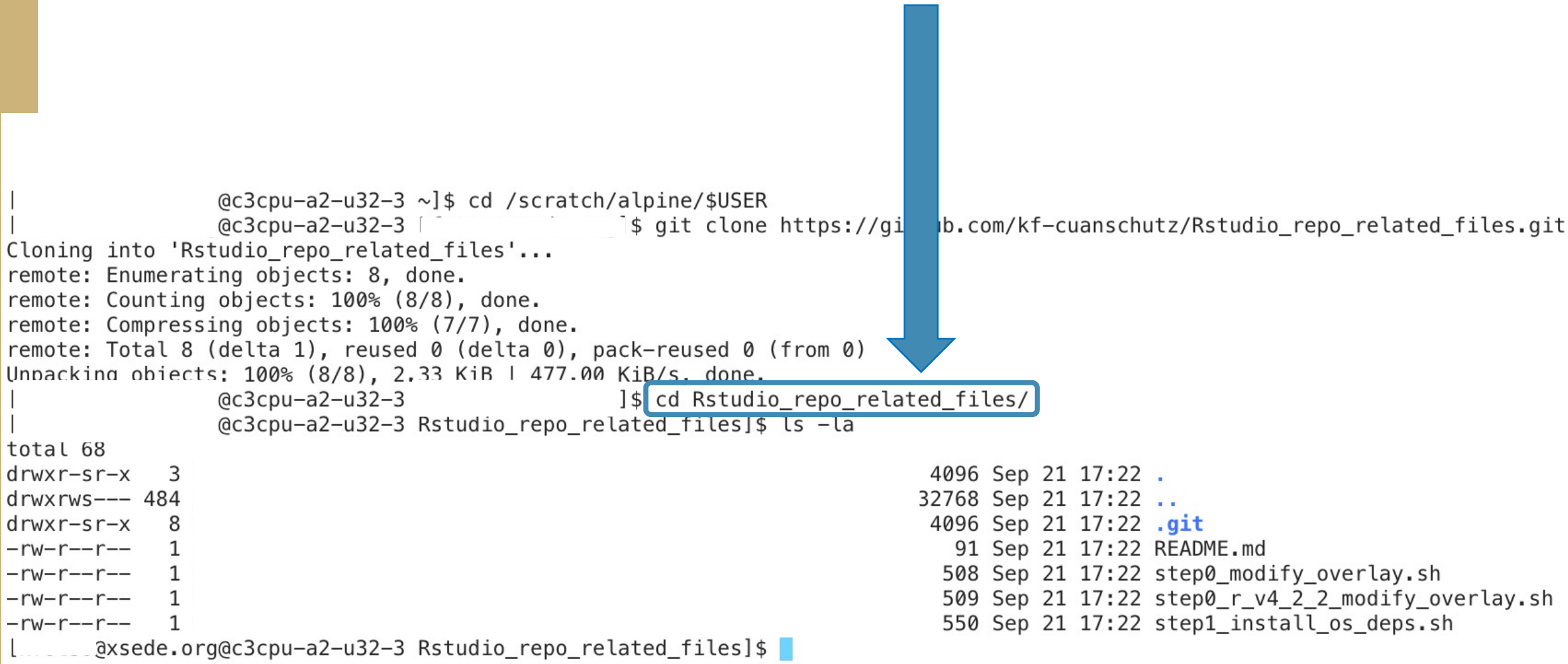
- Clone the repository below with:

git clone https://github.com/kf-cuanschutz/Rstudio_repo_related_files.git

```
|              @c3cpu-a2-u32-3 ~]$ cd /scratch/alpine/$USER
|              @c3cpu-a2-u32-3 [redacted]$ git clone https://github.com/kf-cuanschutz/Rstudio_repo_related_files.git
Cloning into 'Rstudio_repo_related_files'...
remote: Enumerating objects: 8, done.
remote: Counting objects: 100% (8/8), done.
remote: Compressing objects: 100% (7/7), done.
remote: Total 8 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (8/8), 2.33 KiB | 477.00 KiB/s, done.
|              @c3cpu-a2-u32-3                ]$ cd Rstudio_repo_related_files/
|              @c3cpu-a2-u32-3 Rstudio_repo_related_files]$ ls -la
total 68
drwxr-sr-x   3                                    4096 Sep 21 17:22 .
drwxrws--- 484                                   32768 Sep 21 17:22 ..
drwxr-sr-x   8                                    4096 Sep 21 17:22 .git
-rw-r--r--   1                                      91 Sep 21 17:22 README.md
-rw-r--r--   1                                     508 Sep 21 17:22 step0_modify_overlay.sh
-rw-r--r--   1                                     509 Sep 21 17:22 step0_r_v4_2_2_modify_overlay.sh
-rw-r--r--   1                                     550 Sep 21 17:22 step1_install_os_deps.sh
[      @xsede.org@c3cpu-a2-u32-3 Rstudio_repo_related_files]$
```

Rstudio library dependencies

- Then change directories into /scratch/alpine/$USER/Rstudio_repo_related_files



Rstudio library dependencies

- The step0 files will enable to launch the containers.
- Step0_modify_overlay.sh corresponds to the Rstudio container 4.4.1
- Step0_r_v4_2_2_modify_overlay is a legacy version.

```
|               @c3cpu-a2-u32-3 ~]$ cd /scratch/alpine/$USER
|               @c3cpu-a2-u32-3 [redacted]$ git clone https://github.com/kf-cuansch[redacted]/Rstudio_repo_related_files.git
Cloning into 'Rstudio_repo_related_files'...
remote: Enumerating objects: 8, done.
remote: Counting objects: 100% (8/8), done.
remote: Compressing objects: 100% (7/7), done.
remote: Total 8 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (8/8), 2.33 KiB | 477.00 KiB/s, done.
|               @c3cpu-a2-u32-3                ]$ cd Rstudio_repo_related_files/
|               @c3cpu-a2-u32-3 Rstudio_repo_related_files]$ ls -la
total 68
drwxr-sr-x   3                                                4096 Sep 21 17:22
drwxrws--- 484                                               32768 Sep 21 17:22
drwxr-sr-x   8                                                4096 Sep 21 17:22 .git
-rw-r--r--   1                                                  91 Sep 21 17:22 README.md
-rw-r--r--   1                                                 508 Sep 21 17:22 step0_modify_overlay.sh
-rw-r--r--   1                                                 509 Sep 21 17:22 step0_r_v4_2_2_modify_overlay.sh
-rw-r--r--   1                                                 550 Sep 21 17:22 step1_install_os_deps.sh
|       @xsede.org@c3cpu-a2-u32-3 Rstudio_repo_related_files]$
```

Rstudio library dependencies

- step1_install_os_deps.sh is the main program that will install all the necessary library dependencies.

```
|               @c3cpu-a2-u32-3 ~]$ cd /scratch/alpine/$USER
|               @c3cpu-a2-u32-3 [                    ]$ git clone https://github.com/kf-cuansch    /Rstudio_repo_related_files.git
Cloning into 'Rstudio_repo_related_files'...
remote: Enumerating objects: 8, done.
remote: Counting objects: 100% (8/8), done.
remote: Compressing objects: 100% (7/7), done.
remote: Total 8 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (8/8), 2.33 KiB | 477.00 KiB/s, done.
|               @c3cpu-a2-u32-3                  ]$ cd Rstudio_repo_related_files/
|               @c3cpu-a2-u32-3 Rstudio_repo_related_files]$ ls -la
total 68
drwxr-sr-x   3                                                4096 Sep 21 17:22
drwxrws--- 484                                               32768 Sep 21 17:22
drwxr-sr-x   8                                                4096 Sep 21 17:22 .git
-rw-r--r--   1                                                  91 Sep 21 17:22 README.md
-rw-r--r--   1                                                 508 Sep 21 17:22 step0_modify_overlay.sh
-rw-r--r--   1                                                 509 Sep 21 17:22 step0_r_v4_2_2_modify_overlay.sh
-rw-r--r--   1                                                 550 Sep 21 17:22 step1_install_os_deps.sh
[      @xsede.org@c3cpu-a2-u32-3 Rstudio_repo_related_files]$
```

Rstudio library dependencies

- Let's make both step0 and step1 scripts executable.

```
[                @c3cpu-a2-u32-3 Rstudio_repo_related_files]$ chmod u+x step0_modify_overlay.sh
[                @c3cpu-a2-u32-3 Rstudio_repo_related_files]$ chmod u+x step1_install_os_deps.sh
[                @c3cpu-a2-u32-3 Rstudio_repo_related_files]$ ls
README.md  step0_modify_overlay.sh  step0_r_v4_2_2_modify_overlay.sh  step1_install_os_deps.sh
[      @xsede.org@c3cpu-a2-u32-3 Rstudio_repo_related_files]$
```

Rstudio library dependencies

- Notice that now both scripts are green.

```
[                @c3cpu-a2-u32-3 Rstudio_repo_related_files]$ chmod u+x step0_modify_overlay.sh
[                @c3cpu-a2-u32-3 Rstudio_repo_related_files]$ chmod u+x step1_install_os_deps.sh
[                @c3cpu-a2-u32-3 Rstudio_repo_related_files]$ ls
README.md  step0_modify_overlay.sh  step0_r_v4_2_2_modify_overlay.sh  step1_install_os_deps.sh
[      @xsede.org@c3cpu-a2-u32-3 Rstudio_repo_related_files]$
```

Rstudio library dependencies

- Launch the container with “./step0_modify_overlay.sh”

```
@xsede.org@c3cpu-a2-u32-3 Rstudio_repo_related_files]$ ./step0_modify_overlay.sh
DEBUG   [U=2006460,P=1948245]setValue()                 Updated flag 'tmpdir' value to: /gpfs
/tmp
DEBUG   [U=2006460,P=1948245]persistentPreRun()         Apptainer version: 1.3.6-1.el8
DEBUG   [U=2006460,P=1948245]persistentPreRun()         Parsing configuration file /etc/appta
DEBUG   [U=2006460,P=1948245]SetBinaryPath()            Setting binary path to /usr/libexec/a
r/lpp/mmfs/bin:/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/usr/local/sbin:/usr/local/bin:
DEBUG   [U=2006460,P=1948245]SetBinaryPath()            Using that path for all binaries
DEBUG   [U=2006460,P=1948245]handleConfDir()            /home/kfotso@xsede.org/.apptainer alr
INFO    [U=2006460,P=1948245]Exec()                     User not listed in /etc/subuid, tryin
DEBUG   [U=2006460,P=1948245]UnshareRootMapped()        Executing apptainer in root-mapped un
DEBUG   [U=0,P=1948253]    setValue()                 Updated flag 'tmpdir' value to: /gpfs/a
mp
DEBUG   [U=0,P=1948253]    persistentPreRun()         Apptainer version: 1.3.6-1.el8
DEBUG   [U=0,P=1948253]    persistentPreRun()         Parsing configuration file /etc/apptain
DEBUG   [U=0,P=1948253]    SetBinaryPath()            Setting binary path to /usr/libexec/app
lpp/mmfs/bin:/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/usr/local/sbin:/usr/local/bin:/u
DEBUG   [U=0,P=1948253]    SetBinaryPath()            Using that path for all binaries
DEBUG   [U=0,P=1948253]    handleConfDir()            /home/kfotso@xsede.org/.apptainer alrea
DEBUG   [U=0,P=1948253]    Exec()                     running root-mapped unprivileged
DEBUG   [U=0,P=1948253]    FindFake()                 looking for the fakeroot-sysv command
DEBUG   [U=0,P=1948253]    findOnPath()               Found "fakeroot-sysv" at "/usr/bin/fake
DEBUG   [U=0,P=1948253]    FindFake()                 fakeroot-sysv found at /usr/bin/fakeroo
INFO    [U=0,P=1948253]    Exec()                     Using fakeroot command combined with ro
DEBUG   [U=0,P=1948253]    setUmask()                 Saving umask 0022 for propagation into
DEBUG   [U=0,P=1948253]    checkEncryptionKey()       Checking for encrypted system partition
DEBUG   [U=0,P=1948253]    Init()                     Image format detection
DEBUG   [U=0,P=1948253]    Init()                     Check for sandbox image format
```

Rstudio library dependencies

- Now you can see that you are in the container, based on the prompt.

```
Apptainer> ./step1_install_os_deps.sh
Hit:1 http://archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Get:3 http://archive.ubuntu.com/ubuntu jammy-updates InRelease [128 kB]
Get:4 http://archive.ubuntu.com/ubuntu jammy-backports InRelease [127 kB]
Get:5 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 Packages [71.0 kB]
Get:6 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [3,642 kB]
Get:7 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 Packages [5,496 kB]
Get:8 http://archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [5,691 kB]
Get:9 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [3,327 kB]
Get:10 http://archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [1,581 kB]
Get:11 http://archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 Packages [69.1 kB]
Get:12 http://archive.ubuntu.com/ubuntu jammy-backports/main amd64 Packages [83.2 kB]
Get:13 http://archive.ubuntu.com/ubuntu jammy-backports/universe amd64 Packages [35.2 kB]
Get:14 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [1,274 kB]
Fetched 21.7 MB in 4s (5,430 kB/s)
Reading package lists... Done
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  zlib1g-dev
0 upgraded, 1 newly installed, 0 to remove and 110 not upgraded.
```

Rstudio library dependencies

- Install the library dependencies with ".step1_install_os_deps.sh"

```
Apptainer> ./step1_install_os_deps.sh
Hit:1 http://archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Get:3 http://archive.ubuntu.com/ubuntu jammy-updates InRelease [128 kB]
Get:4 http://archive.ubuntu.com/ubuntu jammy-backports InRelease [127 kB]
Get:5 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 Packages [71.0 kB]
Get:6 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [3,642 kB]
Get:7 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 Packages [5,496 kB]
Get:8 http://archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [5,691 kB]
Get:9 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [3,327 kB]
Get:10 http://archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [1,581 kB]
Get:11 http://archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 Packages [69.1 kB]
Get:12 http://archive.ubuntu.com/ubuntu jammy-backports/main amd64 Packages [83.2 kB]
Get:13 http://archive.ubuntu.com/ubuntu jammy-backports/universe amd64 Packages [35.2 kB]
Get:14 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [1,274 kB]
Fetched 21.7 MB in 4s (5,430 kB/s)
Reading package lists... Done
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  zlib1g-dev
0 upgraded, 1 newly installed, 0 to remove and 110 not upgraded.
```

step1_install_os.deps.sh script

You can see the list of dependencies it installs below.

main CU-Anschutz-HPC-documentation / Rstudio_related_scripts / step1_install_os_deps.sh

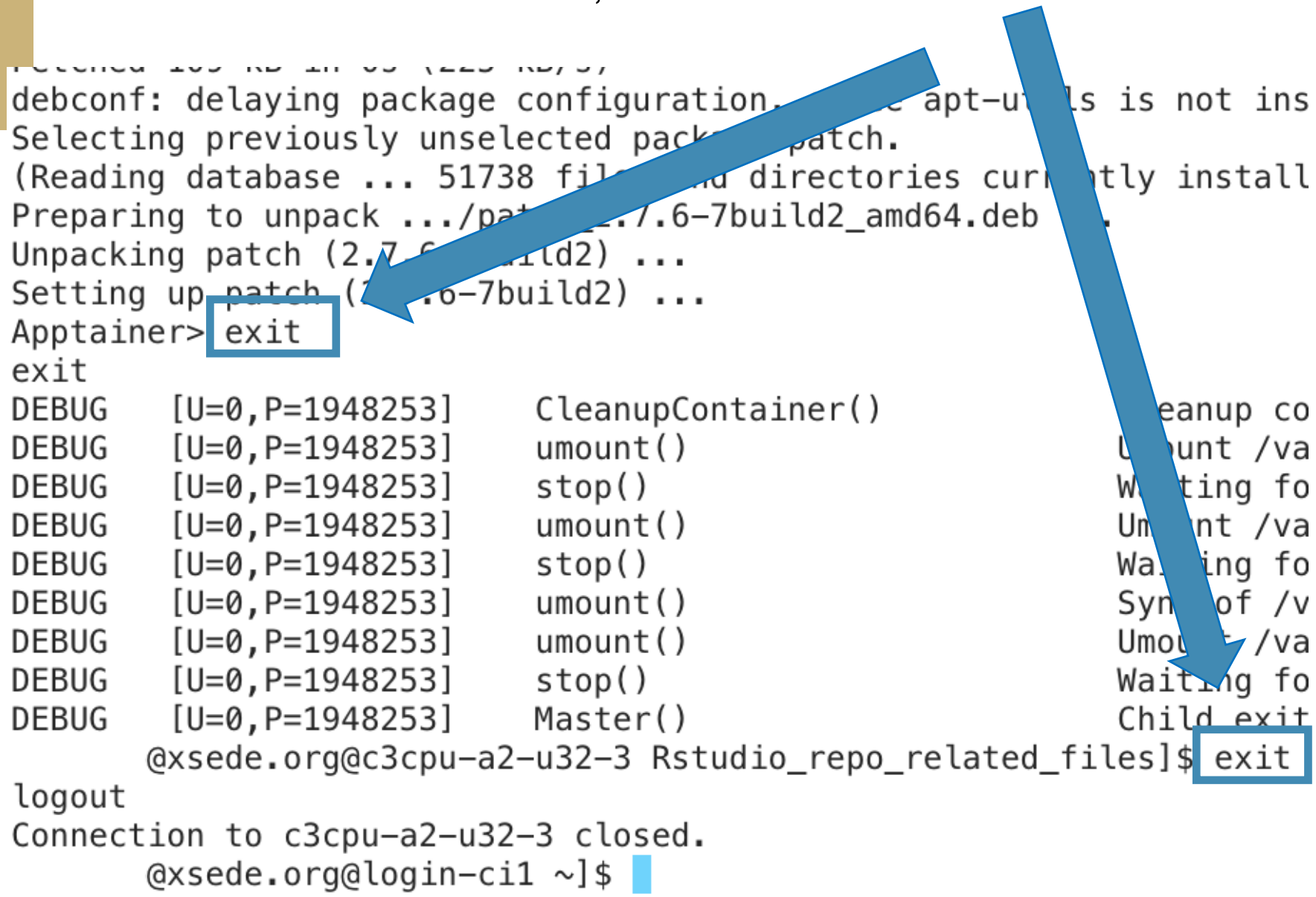
kf-cuanschutz Update step1_install_os_deps.sh

Code Blame Executable File · 20 lines (20 loc) · 550 Bytes

```
1   apt-get -y update
2   apt install -y zlib1g-dev
3   apt install -y libglpk-dev
4   apt install -y glpk-utils
5   apt install -y glpk-doc
6   apt install -y libfftw3-3
7   apt install -y libcairo2-dev
8   apt install -y libxt-dev
9   apt install -y liblzma-dev
10  apt install -y libharfbuzz-dev
11  apt install -y libtiff-dev
12  apt install -y libpng-dev
13  apt install -y libfribidi-dev
14  apt install -y libcurl4-openssl-dev
15  apt install -y libcurl4-nss-dev
16  apt install -y libcurl4-gnutls-dev
17  apt install -y libfreetype-dev
18  apt install -y libbz2-dev
19  apt install -y libhdf5-dev
20  apt install -y patch
```

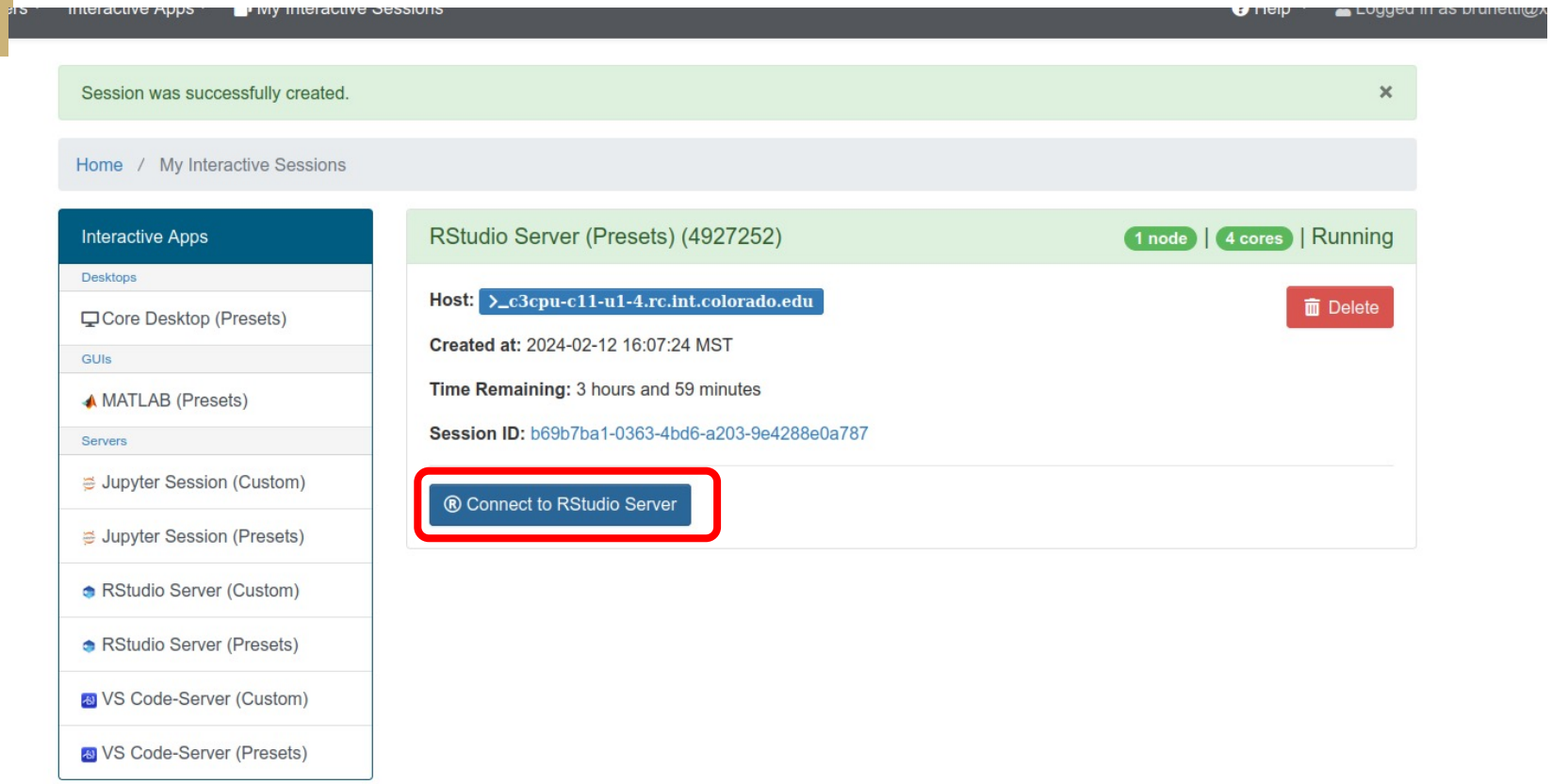
step1_install_os.deps.sh script

- Once the installation is done, make sure to exit the container and the interactive session!



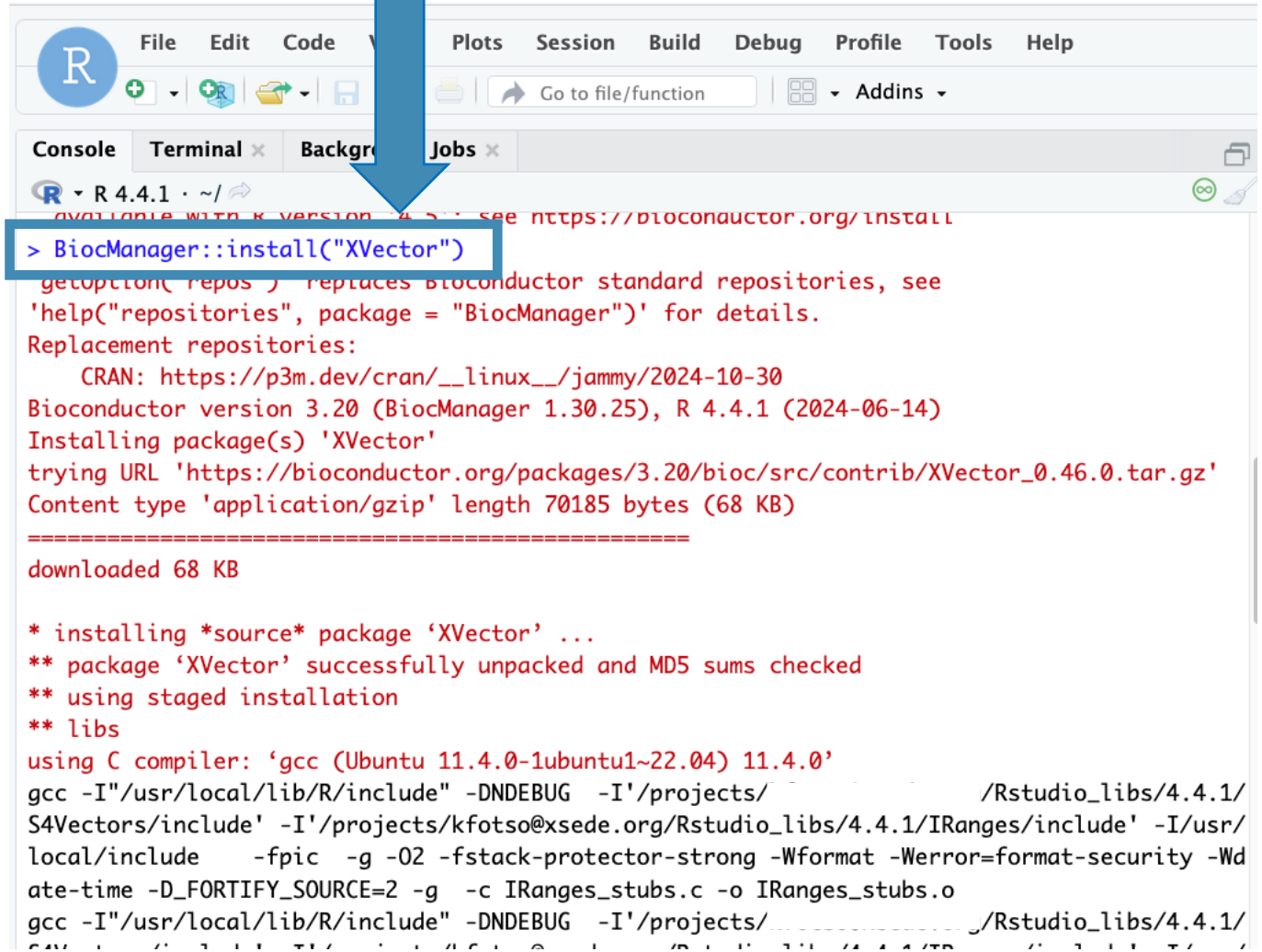
Install a package

- We go back to Ondemand and start a new Rstudio session



Install a package

- Now we can install ”Xvector”!



Install a package

- Now we can install ”Xvector”!

```
vector_copy.o vi_summarization_methods.o -lz -L/usr/local/l
installing to /projects/    @xsede.org/Rstudio_libs/4.4.1/0
s
** R
** inst
** byte-compile and prepare package for lazy loading
** help
*** installing help indices
** building package indices
** testing if installed package can be loaded from temporary
** checking absolute paths in shared objects and dynamic libr
** testing if installed package can be loaded from final loca
** testing if installed package keeps a record of temporary i
* DONE (XVector)

The downloaded source packages are in
        '/projects/      @xsede.org/.rstudioserver/rstudio-4.
ded_packages'
Old packages: 'boot', 'foreign', 'MASS', 'Matrix', 'nlme', 's
Update all/some/none? [a/s/n]:
```

Install a package

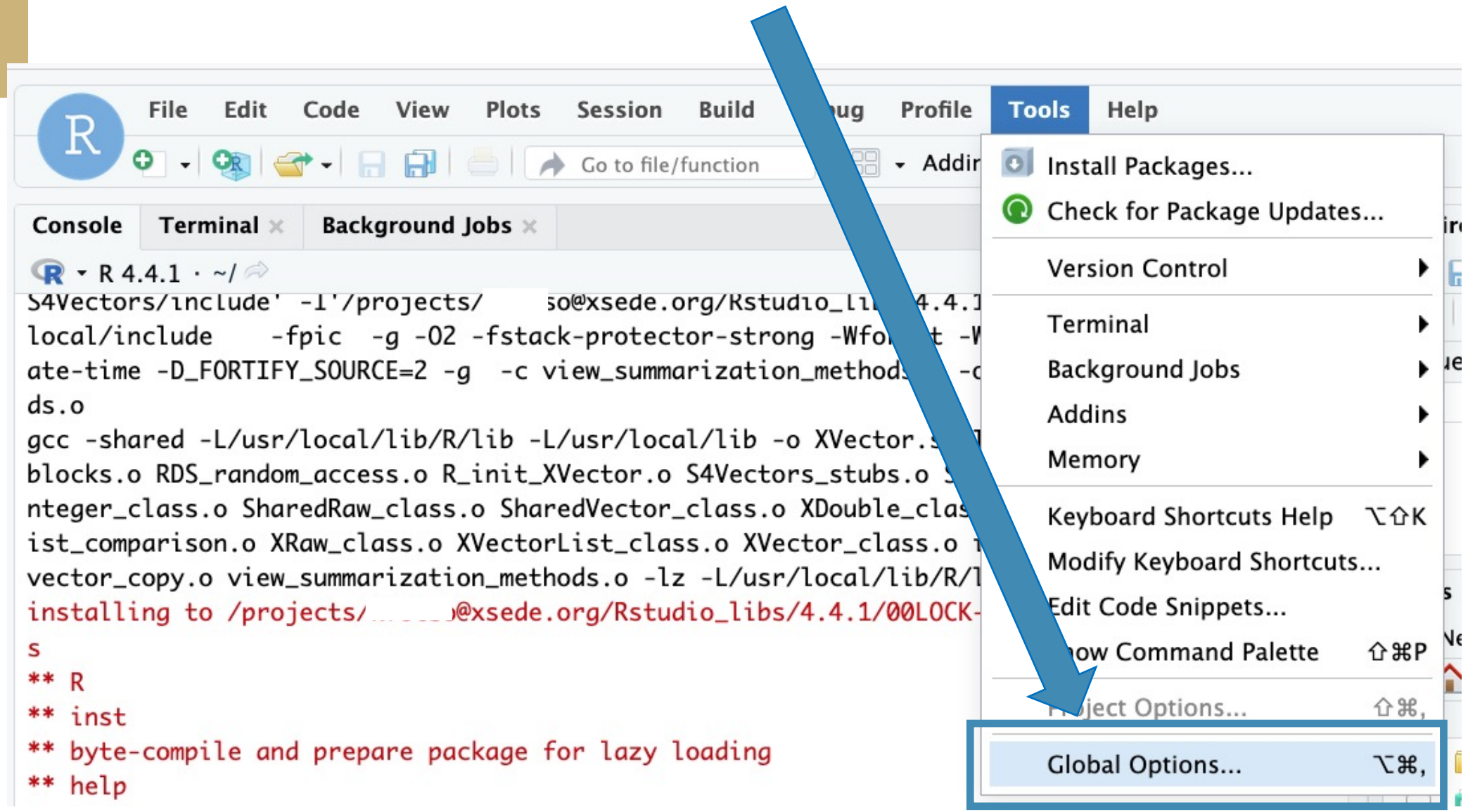
Please refer to this guide for more details:

https://github.com/kf-cuanschutz/CU-Anschutz-HPC-documentation/tree/main/Rstudio_related_scripts

```
vector_copy.o view_summarization_methods.o -lz -L/usr/local/l
installing to /projects/      @xsede.org/Rstudio_libs/4.4.1/0
s
** R
** inst
** byte-compile and prepare package for lazy loading
** help
*** installing help indices
** building package indices
** testing if installed package can be loaded from temporary
** checking absolute paths in shared objects and dynamic libr
** testing if installed package can be loaded from final loca
** testing if installed package keeps a record of temporary i
* DONE (XVector)

The downloaded source packages are in
        '/projects/      )@xsede.org/.rstudioserver/rstudio-4.
ded_packages'
Old packages: 'boot', 'foreign', 'MASS', 'Matrix', 'nlme', 's
Update all/some/none? [a/s/n]:
```

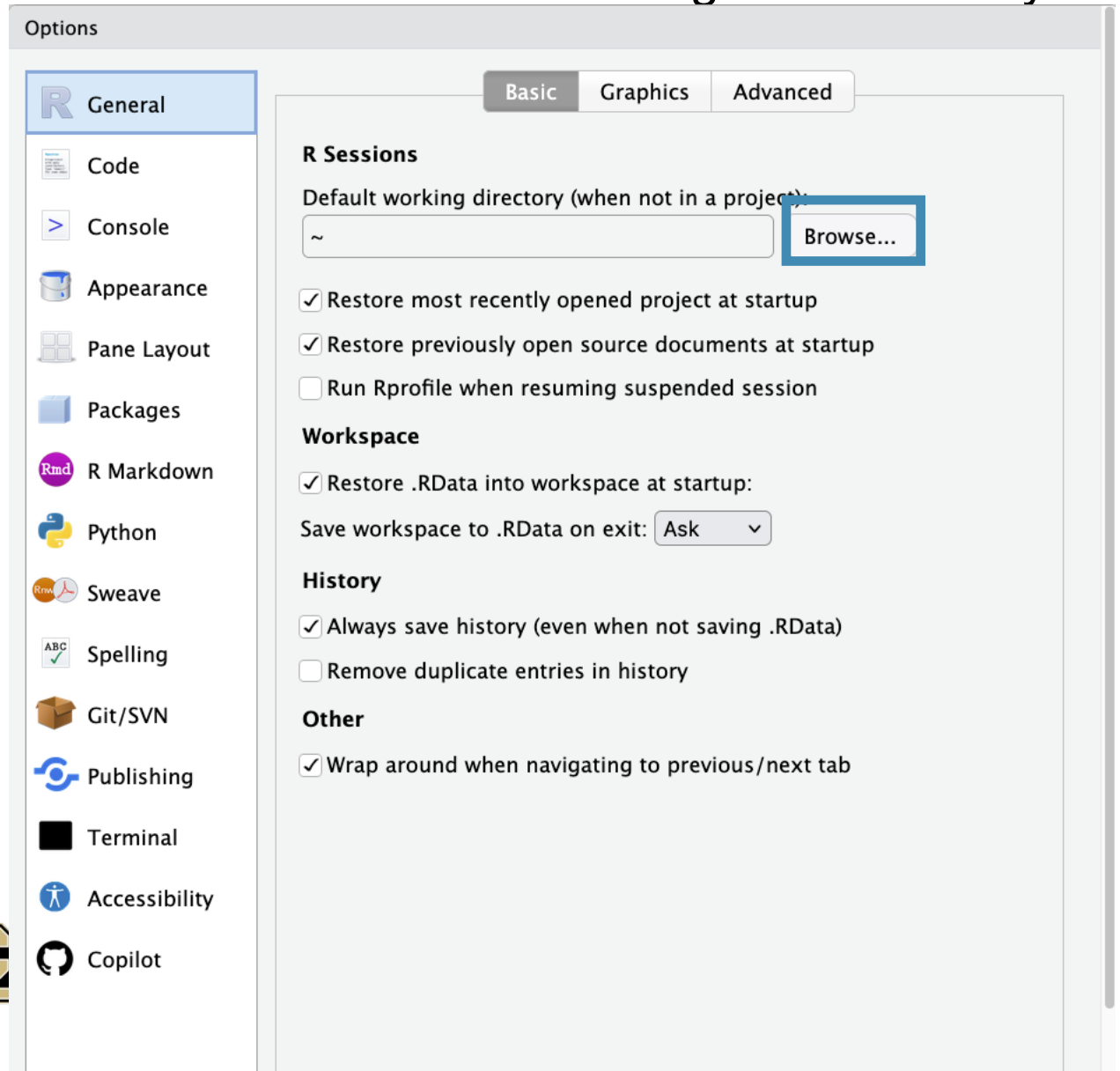
Change my working directory

- Select Tools -> Global Options



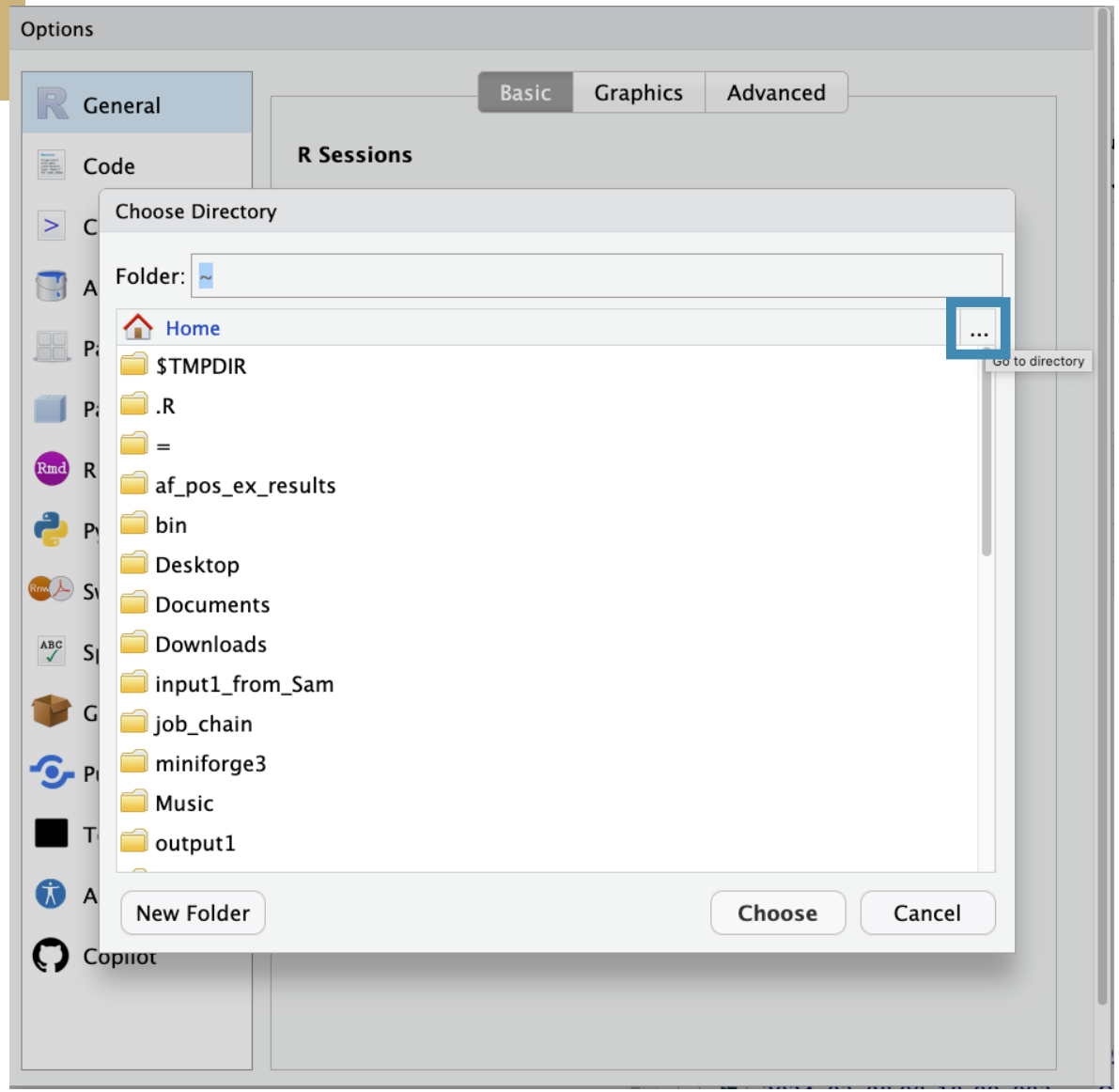
Change my working directory

- Select “Browse” to change the directory.



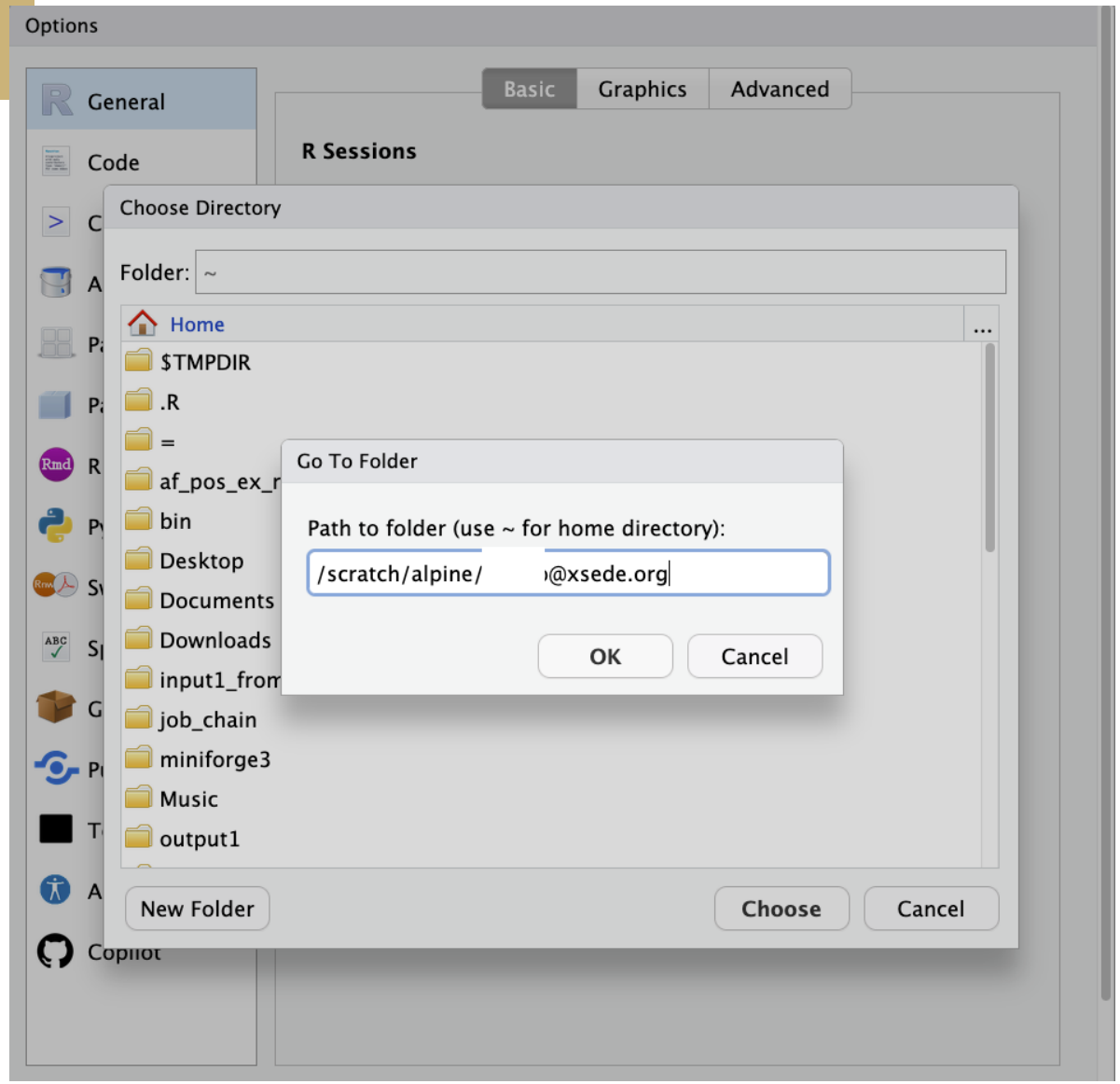
Change my working directory

- Select the three dots on the right to enter your directory manually.



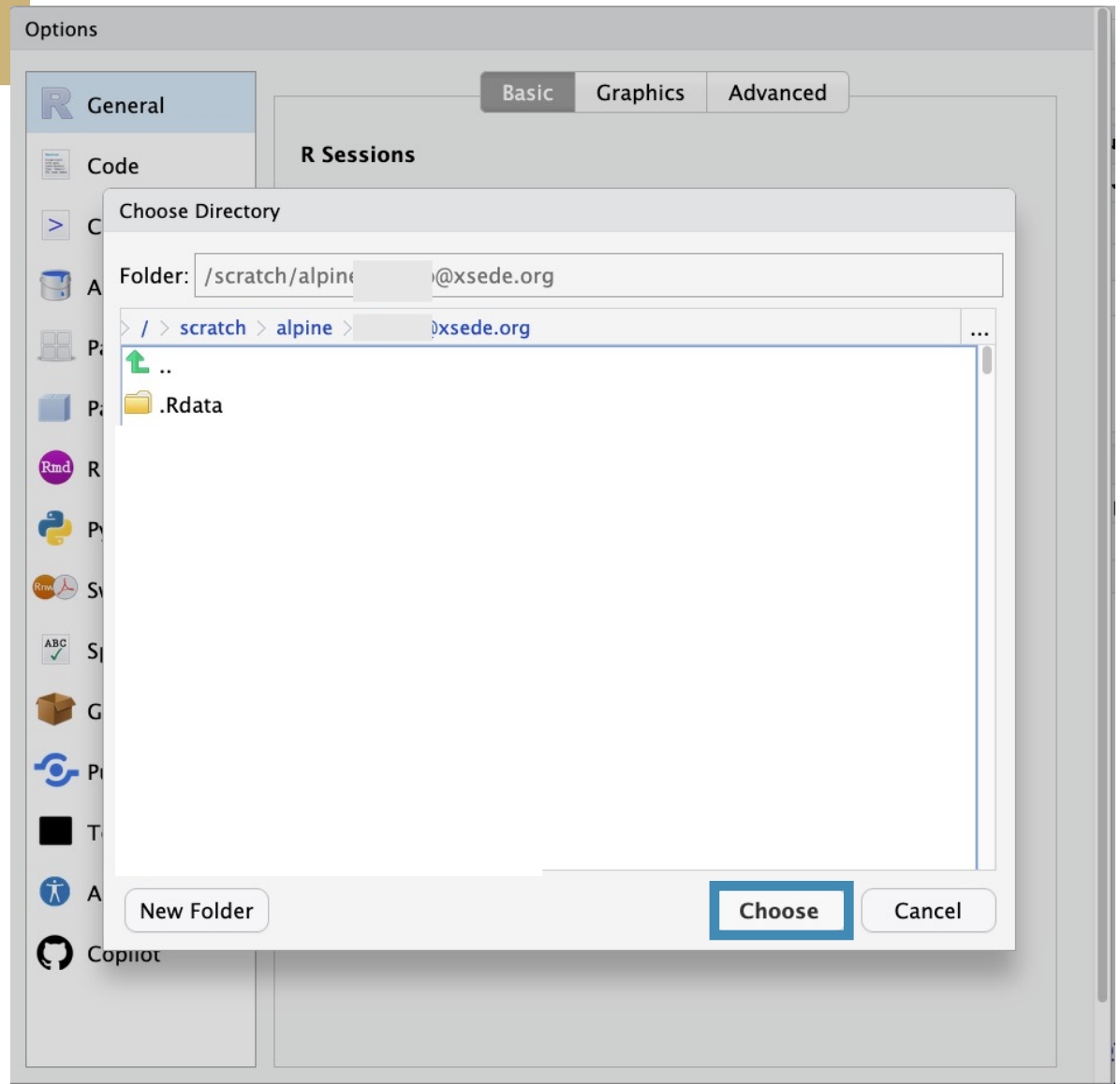
Change my working directory

- Here, I enter my scratch directory as my working directory.



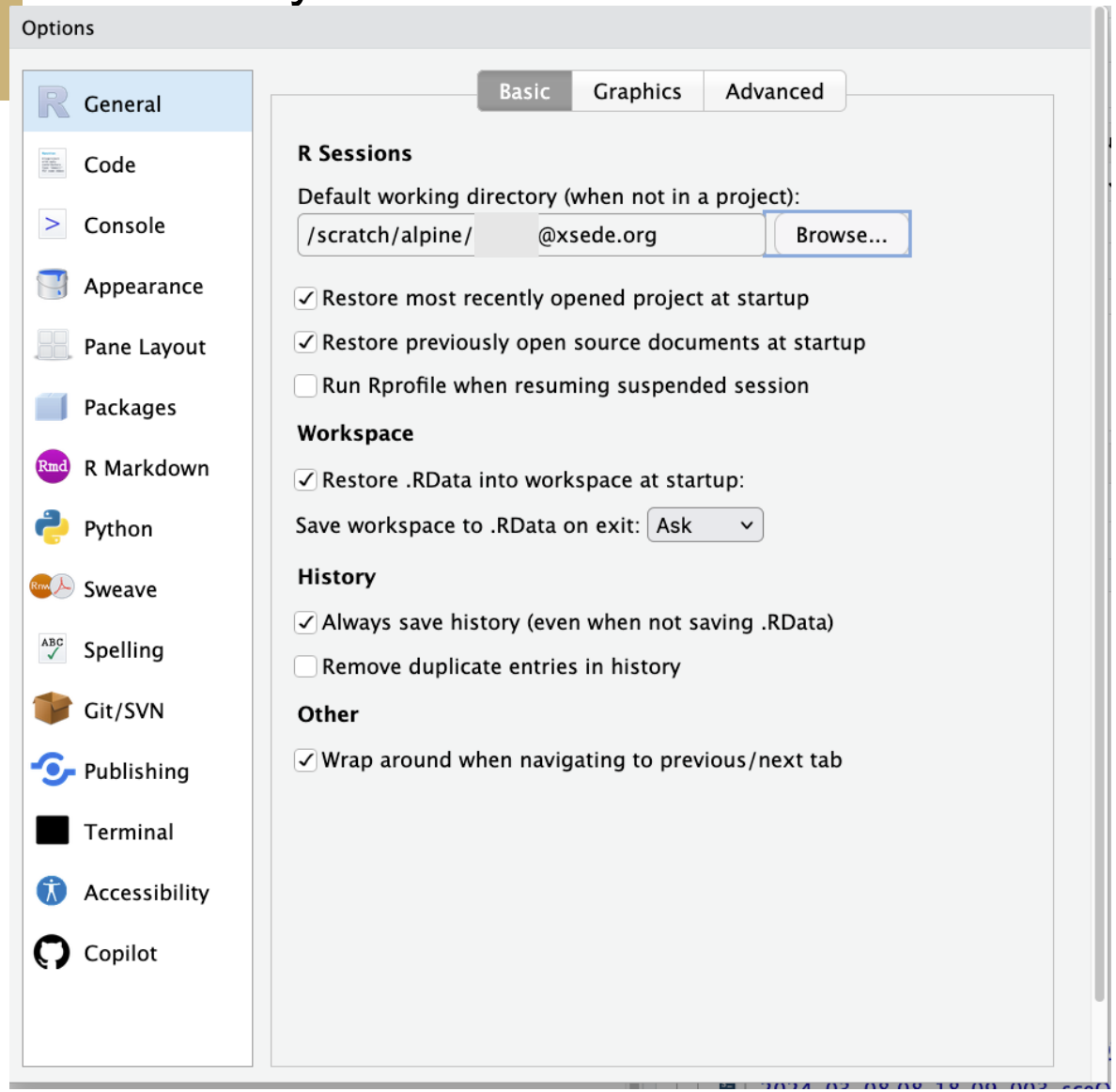
Change my working directory

- Select “Choose”.



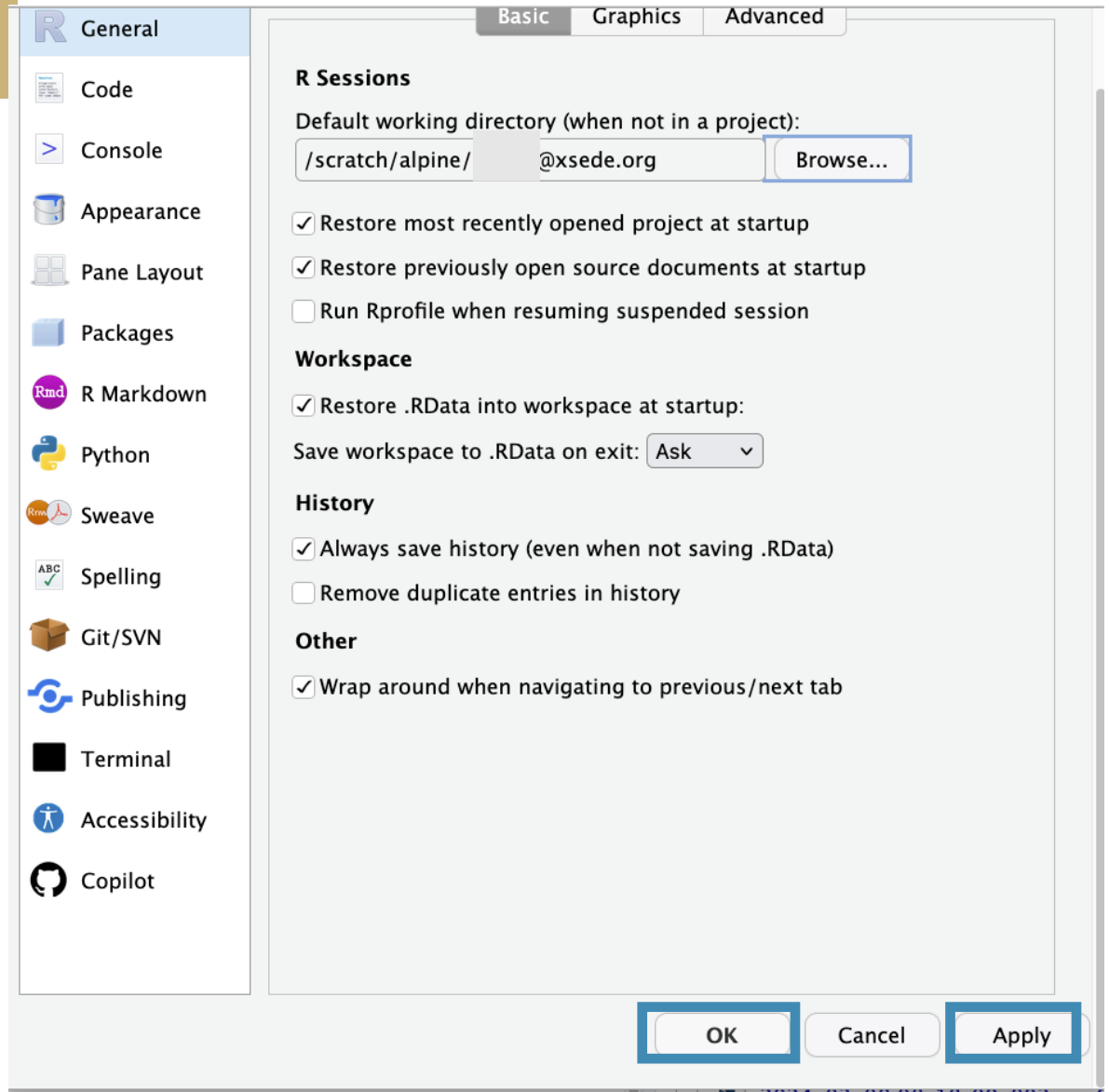
Change my working directory

- The screen below confirms that we've selected the scratch directory as the working directory.



Change my working directory

- Select “Apply” and then “OK” to save the change.



Slurm script with Rstudio environment

- If you need more RAM (>60G).
- If you need more cores (>16).
- If you need to reuse your R environment for longer walltime (>12 hours).

Slurm script with Rstudio environment

```
#!/bin/bash
#SBATCH --nodes=1
#SBATCH --time=00:00:05
#SBATCH --qos=normal
#SBATCH --partition=amilan
#SBATCH --ntasks=1
#SBATCH --account=amc-general
#SBATCH --job-name=R_job
#SBATCH --output=Rjob.%j.out
#SBATCH --error=Rjob.%j.err
#SBATCH --mail-user=foo@cuanschutz.edu
#SBATCH --mail-type=BEGIN,END,FAIL


# Temporary directories
export ALPINE_SCRATCH=/gpfs/alpine1/scratch/$USER
export APPTAINER_TMPDIR=$ALPINE_SCRATCH/apptainer/tmp
export APPTAINER_CACHEDIR=$ALPINE_SCRATCH/apptainer/cache
mkdir -pv $APPTAINER_CACHEDIR $APPTAINER_TMPDIR
```

Number of nodes. Always use 1 unless you are using MPI, gnu parallel or sparkcluster!

Slurm script with Rstudio environment

```
#!/bin/bash
#SBATCH --nodes=1
#SBATCH --time=00:00:05
#SBATCH --qos=normal
#SBATCH --partition=amilan
#SBATCH --ntasks=1
#SBATCH --account=amc-general
#SBATCH --job-name=R_job
#SBATCH --output=Rjob.%j.out
#SBATCH --error=Rjob.%j.err
#SBATCH --mail-user=foo@cuanschutz.edu
#SBATCH --mail-type=BEGIN,END,FAIL


# Temporar directories
export ALPINE_SCRATCH=/gpfs/alpine1/scratch/$USER
export APPTAINER_TMPDIR=$ALPINE_SCRATCH/apptainer/tmp
export APPTAINER_CACHEDIR=$ALPINE_SCRATCH/apptainer/cache
mkdir -pv $APPTAINER_CACHEDIR $APPTAINER_TMPDIR
```

Duration of the slurm job.

Slurm script with Rstudio environment

```
#!/bin/bash
#SBATCH --nodes=1
#SBATCH --time=00:00:05
#SBATCH --qos=normal
#SBATCH --partition=amilan
#SBATCH --ntasks=1
#SBATCH --account=amc-general
#SBATCH --job-name=R_job
#SBATCH --output=Rjob.%j.out
#SBATCH --error=Rjob.%j.err
#SBATCH --mail-user=foo@cuanschutz.edu
#SBATCH --mail-type=BEGIN,END,FAIL


# Temporary directories
export ALPINE_SCRATCH=/gpfs/alpine1/scratch/$USER
export APPTAINER_TMPDIR=$ALPINE_SCRATCH/apptainer/tmp
export APPTAINER_CACHEDIR=$ALPINE_SCRATCH/apptainer/cache
mkdir -pv $APPTAINER_CACHEDIR $APPTAINER_TMPDIR
```

Quality of service

Slurm Quality of service (qos)

- Used to modify or constrain characteristics that a job can have.
- **--qos=normal** corresponds to a walltime of 24 hours and is the default.
- **--qos=long** corresponds to a walltime of up to 7 days
- **--qos=mem** corresponds to high memory jobs only (up to 2TB)

Slurm Quality of service (qos)

- Used to modify or constrain characteristics that a job can have.
- **--qos=normal** corresponds to a walltime of 24 hours and is the default.
- **--qos=long** corresponds to a walltime of up to 7 days
- **--qos=mem** corresponds to high memory jobs only (up to 2TB)
- It is now mandatory to use #SBATCH --qos when interacting with Slurm!

Slurm script with Rstudio environment

```
#!/bin/bash
#SBATCH --nodes=1
#SBATCH --time=00:00:05
#SBATCH --qos=normal
#SBATCH --partition=amilan
#SBATCH --ntasks=1
#SBATCH --account=amc-general
#SBATCH --job-name=R_job
#SBATCH --output=Rjob.%j.out
#SBATCH --error=Rjob.%j.err
#SBATCH --mail-user=foo@cuanschutz.edu
#SBATCH --mail-type=BEGIN,END,FAIL


# Temporar directories
export ALPINE_SCRATCH=/gpfs/alpine1/scratch/$USER
export APPTAINER_TMPDIR=$ALPINE_SCRATCH/apptainer/tmp
export APPTAINER_CACHEDIR=$ALPINE_SCRATCH/apptainer/cache
mkdir -pv $APPTAINER_CACHEDIR $APPTAINER_TMPDIR
```

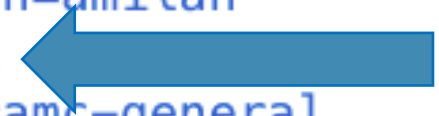
Main CPU partition name

Partitions on Alpine

- amilan -> CPU
- amem -> High memory
- aa100 -> NVIDIA GPU partition
- al40 -> NVIDIA GPU partition
- ami100 -> AMD GPU partition

Slurm script with Rstudio environment

```
#!/bin/bash
#SBATCH --nodes=1
#SBATCH --time=00:00:05
#SBATCH --qos=normal
#SBATCH --partition=amilan
#SBATCH --ntasks=1
#SBATCH --account=amc-general
#SBATCH --job-name=R_job
#SBATCH --output=Rjob.%j.out
#SBATCH --error=Rjob.%j.err
#SBATCH --mail-user=foo@cuanschutz.edu
#SBATCH --mail-type=BEGIN,END,FAIL


# Temporary directories
export ALPINE_SCRATCH=/gpfs/alpine1/scratch/$USER
export APPTAINER_TMPDIR=$ALPINE_SCRATCH/apptainer/tmp
export APPTAINER_CACHEDIR=$ALPINE_SCRATCH/apptainer/cache
mkdir -pv $APPTAINER_CACHEDIR $APPTAINER_TMPDIR
```



Number of CPU cores.

Slurm script with Rstudio environment

```
#!/bin/bash
#SBATCH --nodes=1
#SBATCH --time=00:00:05
#SBATCH --qos=normal
#SBATCH --partition=amilan
#SBATCH --ntasks=1
#SBATCH --account=amc-general
#SBATCH --job-name=R_job
#SBATCH --output=Rjob.%j.out
#SBATCH --error=Rjob.%j.err
#SBATCH --mail-user=foo@cuanschutz.edu
#SBATCH --mail-type=BEGIN,END,FAIL


# Temporary directories
export ALPINE_SCRATCH=/gpfs/alpine1/scratch/$USER
export APPTAINER_TMPDIR=$ALPINE_SCRATCH/apptainer/tmp
export APPTAINER_CACHEDIR=$ALPINE_SCRATCH/apptainer/cache
mkdir -pv $APPTAINER_CACHEDIR $APPTAINER_TMPDIR
```

Account name. All AMC users should use the amc-general account!

Slurm script with Rstudio environment

```
#!/bin/bash
#SBATCH --nodes=1
#SBATCH --time=00:00:05
#SBATCH --qos=normal
#SBATCH --partition=amilan
#SBATCH --ntasks=1
#SBATCH --account=amc-general
#SBATCH --job-name=R_job
#SBATCH --output=Rjob.%j.out
#SBATCH --error=Rjob.%j.err
#SBATCH --mail-user=foo@cuanschutz.edu
#SBATCH --mail-type=BEGIN,END,FAIL


# Temporary directories
export ALPINE_SCRATCH=/gpfs/alpine1/scratch/$USER
export APPTAINER_TMPDIR=$ALPINE_SCRATCH/apptainer/tmp
export APPTAINER_CACHEDIR=$ALPINE_SCRATCH/apptainer/cache
mkdir -pv $APPTAINER_CACHEDIR $APPTAINER_TMPDIR
```

Job name

Slurm script with Rstudio environment

```
#!/bin/bash
#SBATCH --nodes=1
#SBATCH --time=00:00:05
#SBATCH --qos=normal
#SBATCH --partition=amilan
#SBATCH --ntasks=1
#SBATCH --account=amc-general
#SBATCH --job-name=R_job
#SBATCH --output=Rjob.%j.out
#SBATCH --error=Rjob.%j.err
#SBATCH --mail-user=foo@cuanschutz.edu
#SBATCH --mail-type=BEGIN,END,FAIL


# Temporary directories
export ALPINE_SCRATCH=/gpfs/alpine1/scratch/$USER
export APPTAINER_TMPDIR=$ALPINE_SCRATCH/apptainer/tmp
export APPTAINER_CACHEDIR=$ALPINE_SCRATCH/apptainer/cache
mkdir -pv $APPTAINER_CACHEDIR $APPTAINER_TMPDIR
```

- Job output file.
- Similar to stdout in linux.

Slurm script with Rstudio environment

```
#!/bin/bash
#SBATCH --nodes=1
#SBATCH --time=00:00:05
#SBATCH --qos=normal
#SBATCH --partition=amilan
#SBATCH --ntasks=1
#SBATCH --account=amc-general
#SBATCH --job-name=R_job
#SBATCH --output=Rjob.%j.out
#SBATCH --error=Rjob.%j.err
#SBATCH --mail-user=foo@cuanschutz.edu
#SBATCH --mail-type=BEGIN,END,FAIL


# Temporary directories
export ALPINE_SCRATCH=/gpfs/alpine1/scratch/$USER
export APPTAINER_TMPDIR=$ALPINE_SCRATCH/apptainer/tmp
export APPTAINER_CACHEDIR=$ALPINE_SCRATCH/apptainer/cache
mkdir -pv $APPTAINER_CACHEDIR $APPTAINER_TMPDIR
```

%j for the jobID number

Slurm script with Rstudio environment

```
#!/bin/bash
#SBATCH --nodes=1
#SBATCH --time=00:00:05
#SBATCH --qos=normal
#SBATCH --partition=amilan
#SBATCH --ntasks=1
#SBATCH --account=amc-general
#SBATCH --job-name=R_job
#SBATCH --output=Rjob.%j.out
#SBATCH --error=Rjob.%j.err
#SBATCH --mail-user=foo@cuanschutz.edu
#SBATCH --mail-type=BEGIN,END,FAIL


# Temporary directories
export ALPINE_SCRATCH=/gpfs/alpine1/scratch/$USER
export APPTAINER_TMPDIR=$ALPINE_SCRATCH/apptainer/tmp
export APPTAINER_CACHEDIR=$ALPINE_SCRATCH/apptainer/cache
mkdir -pv $APPTAINER_CACHEDIR $APPTAINER_TMPDIR
```

Job error file.

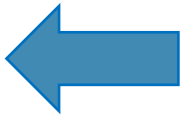
Slurm script with Rstudio environment

```
#!/bin/bash
#SBATCH --nodes=1
#SBATCH --time=00:00:05
#SBATCH --qos=normal
#SBATCH --partition=amilan
#SBATCH --ntasks=1
#SBATCH --account=amc-general
#SBATCH --job-name=R_job
#SBATCH --output=Rjob.%j.out
#SBATCH --error=Rjob.%j.err
#SBATCH --mail-user=foo@cuanschutz.edu
#SBATCH --mail-type=BEGIN,END,FAIL


# Temporary directories
export ALPINE_SCRATCH=/gpfs/alpine1/scratch/$USER
export APPTAINER_TMPDIR=$ALPINE_SCRATCH/apptainer/tmp
export APPTAINER_CACHEDIR=$ALPINE_SCRATCH/apptainer/cache
mkdir -pv $APPTAINER_CACHEDIR $APPTAINER_TMPDIR
```

Email address and request to receive a notification.

Slurm script with Rstudio environment

```
#!/bin/bash
#SBATCH --nodes=1
#SBATCH --time=00:00:05
#SBATCH --qos=normal
#SBATCH --partition=amilan
#SBATCH --ntasks=1
#SBATCH --account=amc-general
#SBATCH --job-name=R_job
#SBATCH --output=Rjob.%j.out
#SBATCH --error=Rjob.%j.err
#SBATCH --mail-user=foo@cuanschutz.edu
#SBATCH --mail-type=BEGIN,END,FAIL


# Temporary directories
export ALPINE_SCRATCH=/gpfs/alpine1/scratch/$USER
export APPTAINER_TMPDIR=$ALPINE_SCRATCH/apptainer/tmp
export APPTAINER_CACHEDIR=$ALPINE_SCRATCH/apptainer/cache
mkdir -pv $APPTAINER_CACHEDIR $APPTAINER_TMPDIR
```



We need to redirect the temporary files to the scratch filesystem.

Slurm script with Rstudio environment

"${r_app_version}.sif" is the name of the container image.

```
# Current R version is 4.4.1
export r_app_version="4.4.1"

apptainer exec --bind /projects,/scratch/alpine,$CURC_CONTAINER_DIR_OOD \
        --fakeroot \
        --overlay \
        /projects/$USER/.rstudioserver/rstudio-${r_app_version}/rstudio-server-${r_app_version}_overlay.img:ro \
        /curc/sw/containers/open_ondemand/rstudio-server-${r_app_version}.sif \
        Rscript R_test.R
```

Slurm script with Rstudio environment

- “exec” is going to run the container.

```
# Curren  R version is 4.4.1
export r pp_version="4.4.1"

apptainer exec --bind /projects,/scratch/alpine,$CURC_CONTAINER_DIR_OOD \
        --fakeroot \
        --overlay \
        /projects/$USER/.rstudioserver/rstudio-${r_app_version}/rstudio-server-${r_app_version}_overlay.img:ro \
        /curc/sw/containers/open_ondemand/rstudio-server-${r_app_version}.sif \
        Rscript R_test.R
```

Slurm script with Rstudio environment

- "--bind" will bring Alpine directories into the container.

```
# Current R version is 4.4.1
export r_app_version="4.4.1"

apptainer exec --bind /projects,/scratch/alpine,$CURC_CONTAINER_DIR_OOD \
        --fakeroot \
        --overlay \
        /projects/$USER/.rstudioserver/rstudio-${r_app_version}/rstudio-server-${r_app_version}_overlay.img:ro \
        /curc/sw/containers/open_ondemand/rstudio-server-${r_app_version}.sif \
        Rscript R_test.R
```

Slurm script with Rstudio environment

"${r_app_version}.sif" is the name of the container image (v.4.4.1).

```
# Current R version is 4.4.1
export r_app_version="4.4.1"

apptainer exec --bind /projects,/scratch/alpine,$CURC_CONTAINER_DIR_OOD \
        --fakeroot \
        --overlay
        /projects/$USER/.rstudioserver/rstudio-${r_app_version}/rstudio-server-${r_app_version}_overlay.img:ro \
        /curc/sw/containers/open_ondemand/rstudio-server-${r_app_version}.sif \
        Rscript R_test.R
```

Slurm script with Rstudio environment

- We call R with Rscript.
- R_test.R is the name of the R script. This is the only variable that you will have to worry about.

```
# Current R version is 4.4.1
export r_app_version="4.4.1"

apptainer exec --bind /projects,/scratch/alpine,$CURC_CONTAINER_DIR_OOD \
        --fakeroot \
        --overlay \
        /projects/$USER/.rstudioserver/rstudio-${r_app_version}/rstudio-server-${r_app_version}_overlay.img:ro \
        /curc/sw/containers/open_ondemand/rstudio-server-${r_app_version}.sif \
        Rscript R_test.R
```

Slurm script with Rstudio environment

Please refer to this guide for more details:

https://github.com/kf-cuanschutz/CU-Anschutz-HPC-documentation/blob/main/Rstudio_on_Slurm_.md

```
# Current R version is 4.4.1
export r_app_version="4.4.1"

apptainer exec --bind /projects,/scratch/alpine,$CURC_CONTAINER_DIR_OOD \
        --fakeroot \
        --overlay \
        /projects/$USER/.rstudioserver/rstudio-${r_app_version}/rstudio-server-${r_app_version}_overlay.img:ro \
        /curc/sw/containers/open_ondemand/rstudio-server-${r_app_version}.sif \
        Rscript R_test.R
```

Part II- LMOD R

LMOD R

What is LMOD R?

- It is accessible through the use of **LMOD**, the modern environment module system for HPC.
- You need to be on a compute node interactively or you need to submit a slurm batch script.
- We covered LMOD during the previous workshop last week:

  https://github.com/kf-cuanschutz/CU-Anschutz-HPC-documentation/blob/main/Workshops/LMOD_and_anaconda_v2.pdf

LMOD R

- Please use only the version **4.4.0** to install your packages!

- Note that some advance package installation might require some dependencies. Please open a ticket at hpcsupport@cuanschutz.edu if you are stuck!

LMOD R

Use LMOD R if:

- You will run heavy job on Alpine, such as high memory computation jobs.
- You don’t need to use a GUI interactively.
- You need to run distributed computations, such as with gnu parallel or MPI.

Slurm script

- We use “module load” as seen last week.

```
#!/bin/bash

#SBATCH --partition=amilan                #
#SBATCH --job-name=Rscript_test
#SBATCH --output=Rscript_test.%j.out    #
#SBATCH --error=Rscript_test.%j.err     #
#SBATCH --account=amc-general  # Account
#SBATCH --qos=normal    # Quality of serv
#SBATCH --nodes=1     # Total number of n
#SBATCH --ntasks=4    # Total number of c
#SBATCH --mail-type=ALL  # Begin, End an
#SBATCH --mail-user=kevin.fotso@cuanschu
#SBATCH --time=00:00:01


# Load the compiler that R  depends on.
module load R/4.4.0

# Run the R command
Rscript R_test.
```

Slurm script

- Call “Rscript” to run your R software.

```
#!/bin/bash

#SBATCH --partition=amilan                #
#SBATCH --job-name=Rscript_test
#SBATCH --output=Rscript_test.%j.out    #
#SBATCH --error=Rscript_test.%j.err     #
#SBATCH --account=amc-general  # Account
#SBATCH --qos=normal   # Quality of serv
#SBATCH --nodes=1    # Total number of n
#SBATCH --ntasks=4   # Total number of c
#SBATCH --mail-type=ALL  # Begin, End an
#SBATCH --mail-user=kevin.fotso@cuanschu
#SBATCH --time=00:00:01


# Load the compiler that R  depends on.
module load R/4.4.0

# Run the R command
Rscript R_test.
```


University of Colorado Anschutz Medical Campus
Part III- Miniforge

Miniforge R

What is miniforge/anaconda R?

- It gives you the possibility to install many R related packages through miniforge.

- Note that your miniforge R located in your conda environment will be very distinct from LMOD R!

Miniforge R

Use miniforge/anaconda R if:

- You are a frequent anaconda user.

- If you have followed this guide already:

  https://github.com/kf-cuanschutz/CU-Anschutz-HPC-documentation/blob/main/miniforge_migration.md

- You can also refer to our last week's workshop on how to use miniforge.

Miniforge R

Use miniforge/anaconda R if:

- If you plan to share the environment you are using with the rest of your lab on Alpine. For example, one can export a miniforge environment as yml.

- Consider this r-irkernel guide if you plan on using Jupyterlab:

https://curc.readthedocs.io/en/latest/open_ondemand/jupyter_session.html#creating-your-own-custom-jupyter-kernel

R installation

We first access a compute node with acompile.

```
[      @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  --mer
=acompile --bell --oversubscribe srun --pty /bin/bash
salloc: Granted job allocation 17847185
salloc: Nodes c3cpu-a2-u32-3 are ready for job
[      @xsede.org@c3cpu-a2-u32-3 ~]$ module load miniforge
(base) [      @xsede.org@c3cpu-a2-u32-3 ~]$ conda create --name R_workshop r-base r-essentials -y
Channels:
 - conda-forge
Platform: linux-64
Collecting package metadata (repodata.json): done
Solving environment: done


==> WARNING: A newer version of conda exists. <==
    current version: 24.11.3
    latest version: 25.7.0
```

R installation

We load miniforge.

```
[      @xsede.org@login-ci1 ~]$ acompile      tasks=4 --time=12:00:00
acompile: submitting job... salloc --node     --partition=acompile --ntasks=4  --time=12:00:00  --mem
=acompile --bell --oversubscribe srun --p     /bin/bash
salloc: Granted job allocation 17847185
salloc: Nodes c3cpu-a2-u32-3 are ready for job
[      @xsede.org@c3cpu-a2-u32-3 ~]$ module load miniforge
(base) [      @xsede.org@c3cpu-a2-u32-3 ~]$ conda create --name R_workshop r-base r-essentials -y
Channels:
 - conda-forge
Platform: linux-64
Collecting package metadata (repodata.json): done
Solving environment: done


==> WARNING: A newer version of conda exists. <==
    current version: 24.11.3
    latest version: 25.7.0
```

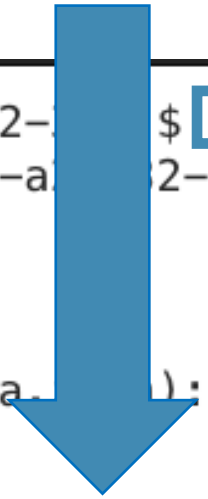
R installation

We create a new R environment with R v.4.4.3

```
[      @xsede.org@login-ci1 ~]$ acompile      tasks=4 --time=12:00:00
acompile: submitting job... salloc --node     --partition=acompile --ntasks=4  --time=12:00:00  --mem
=acompile --bell --oversubscribe srun --p     /bin/bash
salloc: Granted job allocation 17847185
salloc: Nodes c3cpu-a2-u32-3 are ready f      b
[      @xsede.org@c3cpu-a2-u32-3 ~]$ module load miniforge
(base) [      @xsede.org@c3cpu-a2-u32-3 ~]$ conda create --name R_workshop r-base r-essentials -y
Channels:
 - conda-forge
Platform: linux-64
Collecting package metadata (repodata.json): done
Solving environment: done


==> WARNING: A newer version of conda exists. <==
    current version: 24.11.3
    latest version: 25.7.0
```

R installation

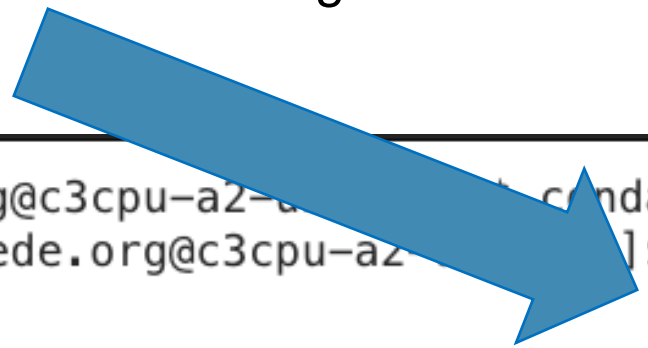
We activate the ENV.



```
(base) [      @xsede.org@c3cpu-a2-u32-      $ conda activate R_workshop
(R_workshop) [      @xsede.org@c3cpu-a      2-3 ~]$  conda install conda-forge::r-seurat
Channels:
 - conda-forge
Platform: linux-64
Collecting package metadata (repodata.      ): done
Solving environment: done


==> WARNING: A newer version of conda exists. <==
    current version: 24.11.3
    latest version: 25.7.0

Please update conda by running

    $ conda update -n base -c conda-forge conda
```

Let install Seurat!

We install Seurat using the conda-forge channel.



```
(base) [      @xsede.org@c3cpu-a2-u          conda activate R_workshop
(R_workshop) [      @xsede.org@c3cpu-a2-          ]$ conda install conda-forge::r-seurat
Channels:
 - conda-forge
Platform: linux-64
Collecting package metadata (repodata.json): done
Solving environment: done


==> WARNING: A newer version of conda exists. <==
    current version: 24.11.3
    latest version: 25.7.0

Please update conda by running

    $ conda update -n base -c conda-forge conda
```


University of Colorado Anschutz Medical Campus
THANK YOU