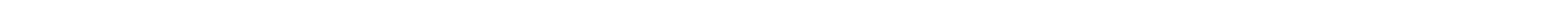


Using RStudio on Hoffman2

Hoffman2 Happy Hour

Charles Peterson





Welcome to Hoffman2 Happy Hours



Welcome to the Hoffman2 Happy Hours

- Short presentations on HPC-related topics and practical uses of Hoffman2
- Thoughts for future “Happy Hour” topics? 



cpeterson@oarc.ucla.edu





Access the Workshop Files

This presentation and accompanying materials are available on UCLA OARC GitHub Repository  <https://github.com/ucla-oarc-hpc>

You can view the slides:

-  PDF format - WS_H2HH-Rstudio.pdf
-  HTML Workshop Slides https://ucla-oarc-hpc.github.io/H2HH_rstudio
-  Recordings can be found on our [BOX account](#)

Clone the repository for workshop files:

```
1 git clone https://github.com/ucla-oarc-hpc/H2HH_rstudio
```

Clone the H2 Rstudio repository:

```
1 git clone https://github.com/ucla-oarc-hpc/H2-RStudio
```

RStudio Information



What is Rstudio

A powerful IDE for R, data visualization, and script management



But why do you want to use RStudio on Hoffman2 when you can use your own computer???

RStudio on Hoffman2 provides access to:

- Higher Memory
- multi-core processing
- GPUs
- Your Hoffman2-hosted data



RStudio Formats

There are two main (free) RStudio formats that researchers can use

RStudio Desktop

- Standalone desktop application
- Runs locally on the machine running R
- [Hoffman2 information on Desktop version](#)

RStudio Server

- Run RStudio as a server process
- Runs R on a remote server
- Connect to server and open on a web browser
- [Hoffman2 information on Server version](#)



RStudio on Hoffman2

RStudio Desktop

- Must connect to H2 via X11 forwarding or remote desktop
- Runs R and Rstudio from H2 modules
- sluggish interaction depending on connection

RStudio Server

- R/Rstudio from Docker container
 - Ran with Apptainer
 - [Workshop on using containers on Hoffman2](#)
- Isolate container OS image
 - Separate version of R that was built from Hoffman2 modules
 - **DON'T load R modules for server version

Running Rstudio Desktop



Connect via X11

Login to Hoffman2 via [X11 Forwarding](#)

- On your local machine, ensure you have an X11-enabled application installed:
 - Windows: Use MobaXterm, XMing, or Cygwin. For WSL, install an X server like VcXsrv or XMing.
 - macOS: Install and launch XQuartz.
 - Linux: Ensure X11 forwarding is enabled on your terminal.
- Use the following command to login to Hoffman2 with X11 Forwarding:

```
1 ssh -X USER@hoffman2.idre.ucla.edu
```

The -X flag enables your local computer's X11 server to display the RStudio GUI.



Connect via Remote Desktop

Connect via Remote Desktop

- Alternatively, you can use a remote desktop application:
- NoMachine or X2Go: Configure the session with the correct server address and login credentials. Set the desktop resolution and environment based on your preferences.

Information on using X11 and Remote Desktop on Hoffman2 can be found on our workshop [Visualization application on HPC](#)



Running Rstudio Desktop

Once connected to Hoffman2 via X11 or Remote Desktop,

- Start an interactive session to request resources:
- Load the necessary modules:
- Launch RStudio:

```
1 qrsh -l h_data=10G,h_rt=1:00:00
```

```
1 module gcc/10.2.0
2 module R/4.2.0
3 module load Rstudio
```

```
1 module load Rstudio
2 rstudio &
```

If everything is set up correctly, the RStudio GUI will appear on your local computer or remote desktop.

Running Rstudio Server



Starting RStudio

Get An Interactive Job

Containers cannot run on login nodes.

```
1 qrsh -l h_data=10G
```

- You **MUST** use a compute node

Modify the qrsh to meet your RStudio computing needs

- More memory and/or job time
- More cores
- Using GPUs

```
1 qrsh -l h_data=50G,h_rt=5:00:00
```

```
1 qrsh -l h_data=10G -pe shared 10
```

```
1 qrsh -l h_data=10G,gpu,V100
```



Setting up directoreis

Create Temp Directories

- Rstudio requires writable temporary directories outside of the container
- It can be anywhere you have write access
- Create directories in
\$SCRATCH/rstudiotmp

```
1 mkdir -pv $SCRATCH/rstudiotmp/{var/lib,var,
```



Loading RStudio

Load the Apptainer Module

- Apptainer is software that will run the Rstudio container

```
1 module load apptainer
```

RStudio Server on Hoffman2 created from Docker

- Separate R from modules on Hoffman2
 - DO NOT load R modules
 - R packages may need to be reinstalled



Starting RStudio

```
1 export RSTUDIO_VERSION=4.1.0
2 apptainer run \
3     -B $SCRATCH/rstudiotmp/var/lib:/var/lib/rstudio-server
4     -B $SCRATCH/rstudiotmp/var/run:/var/run/rstudio-server
5     -B $SCRATCH/rstudiotmp/tmp:/tmp \
6         $H2_CONTAINER_LOC/h2-rstudio_${RSTUDIO_VERSION}.sif
```

- `apptainer run`
 - Starts the RStudio container
 - `-B $SCRATCH/rstudiotmp/[dir]:[/dir]`
 - Mounts tmp directories to the container
 - `$H2_CONTAINER_LOC/h2-rstudio_4.1.0.sif`
 - Location of RStudio container
 - Can be change to different RStudio versions
-
- Information will display about RStudio session
 - Note the **compute node name** and **port number**.
 - Displays `ssh -N -L ...` info to be ran
 - You will see a Rstudio Password
 - Needed to open Rstudio



Note

KEEP THIS TERMINAL OPEN UNTIL YOU JOB IS DONE



Connecting to Rstudio

Once Rstudio process is started:

- Open another terminal on your local computer
- Run the port forward command
 - Creates a connection from local computer to compute node
- Change port **8787** if needed
- **nXXX** is the compute node name
- **username** is your Hoffman2 username

```
1 ssh -N -L 8787:nXXX:8787 username@hoffman2.idr
```

This will create a remote ssh connection from your local computer to the Hoffman2 compute node running Rstudio via port **8787**



Opening RStudio

- Finally, open a web browser
 - Type URL of RStudio Server
 - Will ALWAYS be localhost
 - Change port **8787** if needed

```
1 http://localhost:8787
```



Running Rstudio - The Easy Way

- [h2_rstudio.sh](#)
 - Script that runs everything from the previous slide
 - Starts Rstudio and opens a web browser for you
 - Runs on your **local computer** (not Hoffman2)
- Download script
- To display how to use this script
- Run script
 - Replace [username](#) with Hoffman2 username



h2-studio.sh Information

Look at our [Github page](#)

```
1 wget https://raw.githubusercontent.com/ucla-oarc-hpc/H2-RStudio/main/h2_
2 chmod +x h2_rstudio.sh
```

```
1 ./h2_rstudio.sh -h
```

```
1 ./h2_rstudio.sh -u username
```

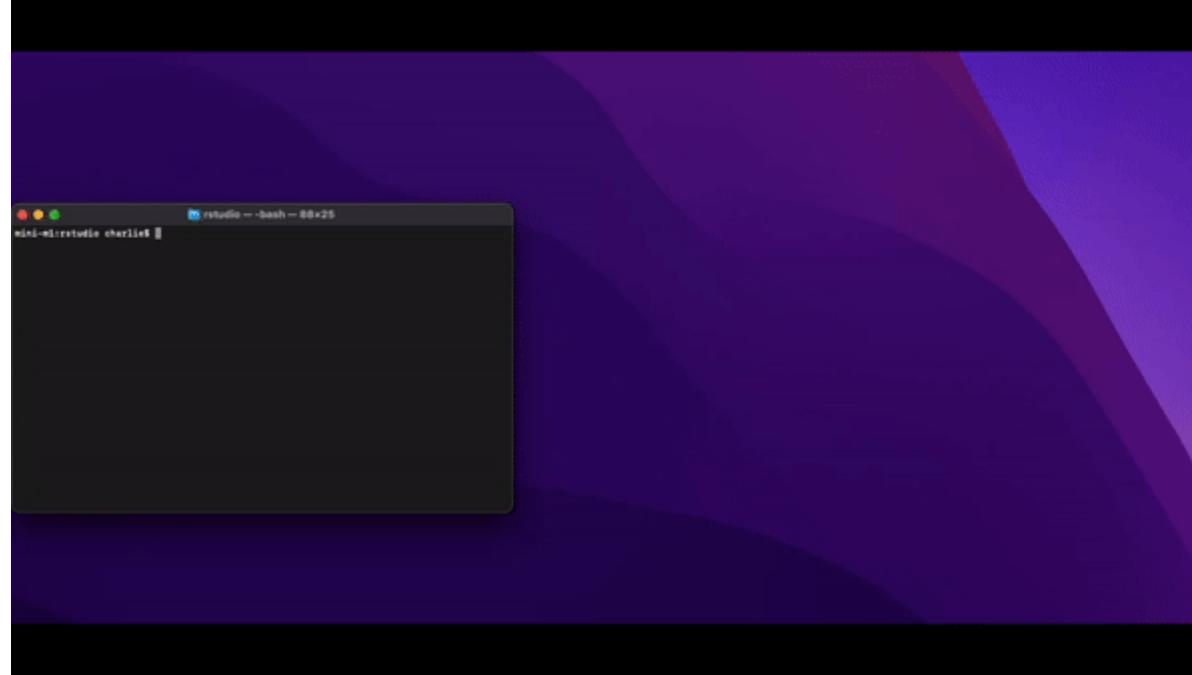


Tested Platforms

- Mac's terminal app
- Window's WSL2
- MoboXterm
- GitBash



RStudio Script



This RStudio Script is currently on our [GitHub page](#)

```
1 This script will create an RStudio session on a compute node on Hoffman2.  
2  
3 REQUIRED OPTIONS:  
4 -u [username] Your Hoffman2 username (mandatory)  
5  
6 OPTIONAL PARAMETERS:  
7 -m [MEMORY] Memory requirements in GB (default: 10 GB)  
8 -t [TIME] Time of RStudio job in HH:MM:SS (default: 2:00:00)
```

```
9  -v [VERSION]    RStudio version (default: 4.1.0)
10 -p               Request high-priority queue (highp)
11 -g [GPUTYPE]    Request GPU resources, where GPUTYPE can be 'V100', 'A100', A6000, etc.
12
13 HELP:
14 -h               Show this usage message
```



Info on RStudio Container

- Rstudio container was built using Docker
 - Based on RStudio images from the [Rocker Project](#)
 - Hoffman2 containers located at `$H2_CONTAINER_LOC`
 - RStudio containers are named:
 - `h2-rstudio_X.Y.Z.sif`
 - Where `X.Y.Z` is the R version
- View all available RStudio containers by running

```
1 module load apptainer
2 ls $H2_CONTAINER_LOC/h2-rstudio*sif
```



User Packages

- Separate build of R and
- R packages installed in unique directory
 - `~/R/APPTAINER/h2-rstudio_4.1.0` (for `h2_rstudio-4.1.0.sif`)
- [HPC Container files](#)
 - Docker and definition files for Hoffman2 containers
 - RStudio Dockerfiles have all you need to build RStudio



R Package Installs

- Some R packages require extra libraries or software in the container
- Contact us to update this container
 - OR you can modify the Dockerfile for your own container



Error fixing

- If Rstudio does not at start up
 - Possibly due to previous RStudio not shutdown correctly
- Clear out any tmp directories files

```
1 rm -rf $SCRATCH/rstudiotmp  
2 mkdir -pv $SCRATCH/rstudiotmp/{var/lib,var/run,tmp}
```

- Clear out RStudio config files

```
1 rm -rf $HOME/.config/rstudio  
2 rm -rf $HOME/.local/share/rstudio
```



Terminal Access



Using Batch R

- Instead of interactive RStudio, you can run R as a non-interactive batch job
 - Use R from inside RStudio container as a qsub job
- Create job script

```
1 #!/bin/bash
2 #\$ -cwd
3 #\$ -o rstudio_batch.out.\$JOB_ID
4 #\$ -j y
5 #\$ -l h_rt=3:00:00,h_data=10G
6 #\$ -pe shared 1
7
8 # Load the apptainer module
9 . /u/local/Modules/default/init/modules.sh
10 module load apptainer
11
12 # R Version
13 export RSTUDIO_VER=4.4.0
14
15 # Use the RStudio container to run R code
16 apptainer run $H2_CONTAINER_LOC/h2-rstudio_{RSTUDIO_VER}.sif R CMD BATCH myRtest.R
```

- Then run this job script

```
1 qsub rstudio_batch.job
```



Summary

- RStudio Desktop on Hoffman2
 - Connect via X11/Remote Desktop
 - Runs R from H2 modules
- RStudio Server on Hoffman2
 - Access through on your web browser
 - Runs R from inside containers



Thanks and Happy Computing!

Questions? Comments?

- cpeterson@oarc.ucla.edu
- Look at for more [Hoffman2 workshops](#) and other OARC workshops

