

Troubleshooting for RStudio: run RStudio in your own R environment

RStudio is now available in three lab servers:

- <https://XX1.mdanderson.edu/rstudio/>
- <https://XX2.mdanderson.edu/rstudio/>
- <https://XX3.mdanderson.edu/rstudio/>

Path of the default R binary file is `/opt/R/4.1.2/bin/R` Paths of default gcc and g++ are: `/usr/bin/gcc` and `/usr/bin/g++`, version: 4.8.5.

Problems encountered

1. You may need to choose a newer gcc/g++ to compile some R packages which depend on new C++ features (e.g., C++14 and C++17 standards).
2. You may need to specify a shared library path to R when compiling an R package, if it is not available in the default search paths, such as `/usr/lib/`, `/usr/local/lib`.
3. You successfully installed a R package (such as `Rcpp`), when you load it into the RStudio, you may still encounter the following error:

```
/lib64/libstdc++.so.6: version `GLIBCXX_3.4.20' not found
```

The reason for this error is that when you install/build the package, you used a different version of `libstdc++.so` from that when you load it: when you load it, `libstdc++.so` is searched for in the `LD_LIBRARY_PATH`.

Our suggestion to resolve these issues

We suggest that you run a customized instance of RStudio Server in each of the three machines with a different port (not 8787), so that the new RStudio Server is running within your own environment. You will then have permissions to change gcc/g++ and to install required dependencies.

- Log in to any of the three machines, create a conda environment following this tutorial [Create a conda environment](#).
- Under your home directory, clone this github repository: `git clone https://github.com/grst/rstudio-server-conda`
- Then `cd rstudio-server-conda/local` and edit `start_rstudio_server.sh` and add the IP (XX.XX.XX.XX) of your server machine:

```
/usr/lib/rstudio-server/bin/rsserver --server-daemonize=0 \  
  
  --www-address=XX.XX.XX.XX \  
  
  --www-port=$1 \  
  
  --secure-cookie-key-file=$COOKIE_KEY_PATH \  
  
  --server-pid-file="$CWD/rstudio-server.pid" \  
  
  --server-data-dir="$CWD/rstudio-server" \  
  
  --rsession-which-r=$(which R) \  
  
  --rsession-ld-library-path=$CONDA_PREFIX/lib \  
  
  --rsession-path="$CWD/rsession.sh" \  
  
  --server-user $USER \  
  
  --database-config-file "$CWD/database.conf" \  
  
  $REVOCATION_LIST_PAR
```

- Run a new RStudio server instance on your machine with port 8600

```
nohup ./start_rstudio_server.sh 8600 > logs.txt &
```

Please check if the port is occupied before running the code above. You can run it without nohup to see whether it starts or fails. Choose another port if it is occupied. Check the running status of the RStudio server `ps -ef | grep rstudio-server | grep "$USER"`.

The RStudio server will be running even if you log out.

You can kill the session by `kill the_pid`.