Configuring the Server

RStudio Support
September 15, 2017 03:43

Follow

RStudio Server: Configuring the Server

Overview

RStudio is configured by adding entries to two configuration files (note that these files do not exist by default so you will need to create them if you wish to specify custom settings):

```
/etc/rstudio/rserver.conf
/etc/rstudio/rsession.conf
```

After editing configuration files you should perform a check to ensure that the entries you specified are valid. This can be accomplished by executing the following command:

```
$ sudo rstudio-server verify-installation
```

Note that this command is also automatically executed when starting or restarting the server (those commands will fail if the configuration is not valid).

Network Port and Address

After initial installation RStudio accepts connections on port 8787. If you wish to change to another port you should create an **`/etc/rstudio/rserver.conf`** file (if one doesn't already exist) and add a `www-port` entry corresponding to the port you want RStudio to listen on. For example:

```
www-port=80
```

By default RStudio binds to address 0.0.0.0 (accepting connections from any remote IP). You can modify this behavior using the `www-address` entry. For example:

```
www-address=127.0.0.1
```

Note that after editing the **`/etc/rstudio/rserver.conf`** file you should always restart the server to apply your changes (and validate that your configuration entries were valid). You can do this by entering the following command:

```
$ sudo rstudio-server restart
```

External Libraries

You can add elements to the default LD_LIBRARY_PATH for R sessions (as determined by the R ldpaths script) by adding an `rsession-ld-library-path` entry to the server config file. This might be

Related articles

Accessing RStudio Server Open-Source
Managing the Server
Getting Started with RStudio Server
Version Control with Git and SVN
What is my username on my RStudio Server?

useful for ensuring that packages can locate external library dependencies that aren't installed in the system standard library paths. For example:

```
rsession-ld-library-path=/opt/local/lib:/opt/local/someapp/lib
```

Specifying R Version

By default RStudio Server runs against the version of R which is found on the system PATH (using `which R`). You can override which version of R is used via the `rsession-which-r` setting in the server config file. For example, if you have two versions of R installed on the server and want to make sure the one at `/usr/local/bin/R` is used by RStudio then you would use:

```
rsession-which-r=/usr/local/bin/R
```

Note again that the server must be restarted for this setting to take effect.

Setting User Limits

There are a number of settings which place limits on which users can access RStudio and the amount of resources they can consume. This file does not exist by default so if you wish to specify any of the settings below you should create the file.

To limit the users who can login to RStudio to the members of a specific group, you use the `auth-required-user-group` setting. For example:

```
auth-required-user-group=rstudio_users
```

Additional Settings

There is a separate **`/etc/rstudio/rsession.conf`** configuration file that enables you to control various aspects of R sessions (note that as with `rserver.conf` this file does not exist by default). These settings are especially useful if you have a large number of potential users and want to make sure that resources are balanced appropriately.

Session Timeouts

By default if a user hasn't issued a command for 2 hours RStudio will suspend that user's R session to disk so they are no longer consuming server resources (the next time the user attempts to access the server their session will be restored). You can change the timeout (including disabling it by specifying a value of 0) using the `session-timeout-minutes` setting. For example:

```
session-timeout-minutes=30
```

Note that a user's session will never be suspended while it is running code (only sessions which are idle will be suspended).

As of RStudio version 1.0 session timeout settings can be customized for individual users or groups. You can add `session-timeout-minutes` to the */etc/rstudio/profiles* file:

```
[user]
session-timeout-minutes=30

[@powerusers]
session-timeout-minutes=0
```

Package Library Path

By default RStudio sets the R_LIBS_USER environment variable to ~/R/library. This ensures that packages installed by end users do not have R version numbers encoded in the path (which is the default behavior). This in turn enables administrators to upgrade the version of R on the server without reseting users installed packages (which would occur if the installed packages were in an R-version derived directory).

If you wish to override this behavior you can do so using the `r-libs-user` settings. For example:

```
r-libs-user=~/R/packages
```

CRAN Repository

Finally, you can set the default CRAN repository for the server using the `r-cran-repos` setting. For example:

```
r-cran-repos=https://mirrors.nics.utk.edu/cran/
```

Note again that the above settings should be specified in the **`/etc/rstudio/rsession.conf`** file (rather than the aforementioned `rserver.conf` file).

Related Topics

- Getting Started
- Managing the Server
- Running with a Proxy

Was this article helpful?
3 out of 3 found this helpful

Comments