# RStudio Server Professional Edition

## *Administration Guide*

### *2017-10-04*

# 1 Getting Started

## 1.1 Introduction

RStudio Server enables you to provide a browser based interface (the RStudio IDE) to a version of R running on a remote Linux server. Deploying R and RStudio on a server has a number of benefits, including:

- The ability to access R sessions from any computer in any location;
- Easy sharing of code, data, and other files with colleagues;
- Allowing multiple users to share access to the more powerful compute resources (memory, processors, etc.) available on a well equipped server; and
- Centralized installation and configuration of R, R packages, TeX, and other supporting libraries.

This manual describes *RStudio Server Professional Edition*, which adds many enhancements to the open-source version of RStudio Server, including:

- The ability to run multiple concurrent R sessions per-user.
- Flexible use of multiple versions of R on the same server.
- Project sharing for easy collaboration within workgroups.
- Load balancing for increased capacity and higher availability.
- An administrative dashboard that provides insight into active sessions, server health, and monitoring of system-wide and per-user performance and resource metrics;
- Authentication using system accounts, ActiveDirectory, LDAP, or Google Accounts;
- Full support for PAM (including PAM sessions for dynamically provisioning user resources);

- Ability to establish per-user or per-group CPU priorities and memory limits;

- HTTP enhancements including support for SSL and keep-alive for improved performance;

- Ability to restrict access to the server by IP;

- Customizable server health checks; and

- Suspend, terminate, or assume control of user sessions; Impersonate users for assistance and troubleshooting.

# 1.2 Installation

## 1.2.1 Prerequisites

RStudio Server requires a previous installation of R version 2.11.1 or higher; see below for instructions on installing R on your specific Linux distribution.

RStudio Server interacts frequently with user home directories. If you mount home directories with NFS, we recommend using the `async` mount option along with a modern, high-throughput network connection that can support many simultaneous clients. If you'd like your users to be able to share their projects with each other, see the section on Project Sharing for additional NFS requirements.

## 1.2.2 RedHat / CentOS (5+)

**Installing R**

You can install R for RedHat and CentOS using the instructions on CRAN:
https://cran.rstudio.com/bin/linux/redhat/README.

**Installation Commands**

After downloading the appropriate RedHat/CentOS package for RStudio Server Professional you should execute the following command to complete the installation:

```
sudo yum install --nogpgcheck <rstudio-server-package.rpm>
```

NOTE: If you are running on RedHat 5 you will need to enable the EPEL repository to satisfy RStudio's dependencies on the `libffi` and `rrdtool` packages (these packages are part of the base repository in RedHat 6 and 7 so EPEL is not required on those systems).

## 1.2.3 Debian (8+) / Ubuntu (12.04+)

**Installing R**

To install the latest version of R you should first add the CRAN repository to your system as described here:

- Debian: https://cran.rstudio.com/bin/linux/debian/README.html
- Ubuntu: https://cran.rstudio.com/bin/linux/ubuntu/README.html

You can then install R using the following command:

```
$ sudo apt-get install r-base
```

NOTE: If you do not add the CRAN Debian or Ubuntu repository as described above this command will install the version of R corresponding to your current system version. Since this version of R may be a year or two old it is strongly recommended that you add the CRAN repositories so you can run the most up to date version of R.

**Installation Commands**

After downloading the appropriate Debian/Ubuntu package for RStudio Server Professional you should execute the following commands to complete the installation:

```
$ sudo apt-get install gdebi-core
$ sudo gdebi <rstudio-server-package.deb>
```

## 1.2.4 openSUSE / SLES (11+)

**Installing R**

You can install R for openSUSE or SLES using the instructions on CRAN:
https://cran.rstudio.com/bin/linux/suse/.

Note that the binaries linked to from this page have one additional requirement that isn't satisfied using the default repositories. Before installing R you should install the `libgfortran43` package. This package is available from the SUSE Linux Enterprise SDK. If the SDK repository is available in your environment you can install `libgfortran43` as follows:

```
$ sudo zypper install libgfortran43
```

**Installation Commands**

After downloading the appropriate RPM package for RStudio Server Professional you should execute the following command to complete the installation:

```
$ sudo zypper install <rstudio-server-package.rpm>
```

# 1.3 Management Script

RStudio Server management tasks are performed using the `rstudio-server` utility (installed under `/usr/sbin`). This utility enables the stopping, starting, and restarting of the server, enumeration and suspension of user sessions, taking the server offline, as well as the ability to hot upgrade a running version of the server.

For example, to restart the server you can use the following command:

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

Note that on some systems (including RedHat/CentOS 5 and SLES 11) the `sudo` utility doesn't have the `/usr/sbin` directory in it's path by default. For these systems you can use a full path to the management script. For example:

```
$ sudo /usr/sbin/rstudio-server restart
```

# 1.4 Activation

After completing the installation steps described in the previous section you may need to activate the product before using it. Alternatively, if you haven't previously installed RStudio Server on a system then it will run in evaluation mode for a period of time before requiring activation. To determine the current license status of your system you can use the following command:

```
$ sudo rstudio-server license-manager status
```

To activate the product you obtain a product key and then use the following commands:

```
$ sudo rstudio-server license-manager activate <product-key>
$ sudo rstudio-server restart
```

Note that you need to restart the server in order for licensing changes to take effect.

Additional details on license management (including discussions of offline activation and activating through a proxy server) can be found in the License Management section.

# 1.5 Accessing the Server

## 1.5.1 Logging In

By default RStudio Server runs on port 8787 and accepts connections from all remote clients. After installation you should therefore be able to navigate a web browser to the following address to access the server:

```
http://<server-ip>:8787
```

RStudio will prompt for a username and password and will authenticate access using the PAM authentication scheme configured for the server. Some notes related to user authentication:

- RStudio Server will not permit logins by system users (those with ids < 100).
- By default on Debian/Ubuntu the system default PAM profile ( `/etc/pam.d/other` ) will be used (this can be customized by creating an RStudio PAM profile at `/etc/pam.d/rstudio` ).
- By default on RedHat/CentOS and SLES an RStudio PAM profile ( `/etc/pam.d/rstudio` ) that authenticates using the system username/password database will be used (this can be customized by editing the profile as appropriate).
- User credentials are encrypted using RSA as they travel over the network.

Additional details on customizing RStudio Server authentication are provided in Authenticating Users. Details on customizing the port and enabling SSL are covered in Access and Security.

# 1.5.2 Troubleshooting Problems

If you are unable to access the server after installation, you should run the `verify-installation` command to output additional diagnostics:

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

This command will start the server and run and connect to an R session. Note that this will test the correct installation of RStudio Server and ensure that it can connect to a locally installed version of R. However, it won't test whether networking or authentication problems are preventing access to the server.

If problems persist, you can also consult the system log to see if there are additional messages there. On Debian/Ubuntu systems this will typically be located at:

```
/var/log/syslog
```

On RedHat/CentOS systems this will typically be located at:

```
/var/log/messages
```