



By: Dean Attali  4  3

# How To Set Up R on Ubuntu 14.04

July 13, 2015      Applications Ubuntu

## Introduction

R is a popular open source programming language that specializes in statistical computing and graphics. It is widely used by statisticians for developing statistical software and performing data analysis. One of R's strengths is that it is highly and easily extensible by allowing users to author and submit their own packages. The R community is known to be very active and is noted for continuously adding user-generated statistical packages for specific areas of study, which makes R applicable to many fields of study.

The "Comprehensive R Archive Network" (CRAN) is a collection of sites (called *mirrors*) which carry identical material, consisting of many R packages and the R distributions themselves. You can download R and many R packages from any of the CRAN mirrors, but we will use the RStudio mirror.

In this guide, we will learn how to set up R on a DigitalOcean Droplet running Ubuntu 14.04. If your Droplet is running a different operating system, most of the instructions will still apply, but you may need to modify some of the commands. Following this guide to completion should take about 10-15 minutes.

## Prerequisites

For this tutorial, you will need:

- Ubuntu 14.04 Droplet with at least 1 GB of RAM. If your Droplet has less than 1 GB of RAM, you need to add 1GB of swap space (How To Add Swap on Ubuntu 14.04 gives you step-by-step instructions for adding swap space.)

- A non-root user with sudo privileges ([Initial Server Setup with Ubuntu 14.04](#) explains how to set this up.)

All the commands in this tutorial should be run as a non-root user. If root access is required for the command, it will be preceded by `sudo`. [Initial Server Setup with Ubuntu 14.04](#) explains how to add users and give them sudo access.

## Step 1 — Setting Up APT

To install R, we're going to use the *APT* (Advanced Packaging Tool) tool. It uses a special file that lists the sources of where packages should be downloaded from. That file is `/etc/apt/sources.list`. In order to get the most recent version of R, we need to add the correct *repository* to the list of sources by adding a line to the sources file. The exact line you need to add will vary depending on the exact Ubuntu version. For Ubuntu 14.04, run the following command to add the correct repository to `/etc/apt/sources.list`.

```
-c 'echo "deb http://cran.rstudio.com/bin/linux/ubuntu trusty/" >> /etc/apt/sour
```



If you are running a different Ubuntu version, consult [this document](#) for the correct repository to add.

To authenticate packages downloaded using APT, we have to add a public key. The Ubuntu archives on CRAN are signed with a key with ID E084DAB9. Add this key to your system.

```
$ gpg --keyserver keyserver.ubuntu.com --recv-key E084DAB9
```

Next we need to add the key to `apt`.

```
$ gpg -a --export E084DAB9 | sudo apt-key add -
```

## Step 2 — Installing R

Now that APT has been set up properly, we are ready to use it to install R.

First, we need to update the list of available packages since we updated the sources list.

```
$ sudo apt-get update
```

Now we can install R. We use the `-y` flag to automatically answer **Yes** when asked if we are sure we want to download the package.

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

At this point, you should have an installation of the latest R version on your Droplet. You can test this by running the `R` command.

```
$ R
```

You should see output similar to the following.

```
R version 3.2.1 (2015-06-18) -- "World-Famous Astronaut"
Copyright (C) 2015 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu (64-bit)
```

```
R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.
```

```
  Natural language support but running in an English locale
```

```
R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.
```

```
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.
```

```
>
```

You are now inside the R interactive shell and can run arbitrary R commands.

Quit R, and return to your Droplet with the `q()` function:

```
> q(save = "no")
```

## Step 3 — Installing R Packages from CRAN

Now that R is installed on your Droplet, any user on the Droplet can use R. When R is installed, it automatically installs a number of default packages, but in order to do anything truly meaningful in R you will probably need to install extra packages. It is important to have at least 1 GB of RAM available in order to install many packages, which is the reason that the prerequisites section instructed you to add swap space if your Droplet has less than 1 GB of RAM.

As mentioned previously, CRAN hosts not only R itself, but many R packages as well. To install new R packages that are hosted on CRAN, or to update existing ones, you use the `install.packages()` function in R. If you wanted to install package `somepackage`, you would open R and run the following R command.

```
# This is an example, do not run this
install.packages("somepackage")
```

However, any package installed by a specific user in R will only be available to that user by default. For example, if user `sammy` installs `somepackage`, then user `jessie` will not be able to use `somepackage` until they install it as well.

It is possible to install an R package in a way that makes it available to all users on the Droplet by installing it as `root`. As an example, let's install the `shiny` package, which is a very popular package used to create web applications from R code. One way to install the package as `root` would be to log in as `root`, run R, and run the `install.packages()` command. However, it's recommended not to log in as `root`, so instead we can just run the R command as `root`. We'll also specify the `repos` parameter so that the package is downloaded from the RStudio CRAN repository, the same one we used when downloading R itself.

```
$ sudo su - -c "R -e \"install.packages('shiny', repos = 'http://cran.rstudio
```

By installing a package this way rather than opening R and running the `install.packages()` command, the `shiny` package is made available to all users on the Droplet.

Let's verify that `shiny` was installed correctly by trying to load it. Start an R session.

```
$ R
```

In R, try loading the `shiny` package.

```
> library(shiny)
```

Running the previous command should result in no errors. Now quit R.

```
> q(save = "no")
```

## Step 4 — Installing `devtools` Package

While many R packages are hosted on CRAN and can be installed using the built-in `install.packages()` function, there are many more packages that are hosted on GitHub but are not on CRAN. To install R packages from GitHub, we need to use the `devtools` R package, so let's install it.

The `devtools` R package requires three system packages to be installed on the Droplet, namely `libcurl4-gnutls-dev`, `libxml2-dev`, and `libssl-dev`. Install these three packages:

```
$ sudo apt-get -y install libcurl4-gnutls-dev libxml2-dev libssl-dev
```

Now the `devtools` R package can be installed. Remember that we want to install it using the same method as described SCROLL TO TOP rather than install it within an R session, because `devtools` should be available to all users.

```
$ sudo su - -c "R -e \"install.packages('devtools', repos='http://cran.rstudio.com/)'"
```

The above command to install `devtools` could take several minutes to complete.

## Step 5 — Installing R Packages from GitHub

Now that we have `devtools` installed, we can install any R package that is on GitHub using the `install_github()` function. Just like with CRAN packages, when installing GitHub packages you need to run the command from the system shell to make the package available to all users. Let's try installing the `shinyjs` GitHub package, which adds functionality to the `shiny` package. A GitHub package is defined by its author (`daattali`) and its name (`shinyjs`).

```
$ sudo su - -c "R -e \"devtools::install_github('daattali/shinyjs')\""
```

Let's verify that `shinyjs` was installed correctly by trying to load it. Start an R session.

```
$ R
```

In R, try loading the `shinyjs` package.

```
> library(shinyjs)
```

Running the previous command could result in some messages, but no error messages. Now quit R.

```
> q(save = "no")
```

## Next Steps

You now have a working R installation on your Droplet.

To learn more about R, visit [the official R website](#), or try learning R hands-on and interactively with the [swirl](#) package.

For more information on CRAN and what it offers, visit [the official CRAN website](#).

For a better experience writing R code on your Droplet, you may want to install an RStudio Server using [this tutorial](#).

If you want to host any of your Shiny code on your Droplet, you may want to install a Shiny Server using [this tutorial](#).

## Conclusion

In this guide, we went through the steps required to set up R on an Ubuntu 14.04 Droplet. We also learned the difference between installing R packages from GitHub vs CRAN and how to ensure that these packages are made available for all users on the Droplet.

Heart | 4

Share

Subscribe



Author:  
Dean Attali



Editor:  
Tammy Fox

## Spin up an SSD cloud server in under a minute.

Simple setup. Full root access.  
Straightforward pricing.

DEPLOY SERVER

## Related Tutorials

How to Install and Configure Magento on Ubuntu 14.04

How To Set Up VNC Server on Debian 8

How to Manage Front-End JavaScript and CSS Dependencies with Bower on Ubuntu 14.04

How To Install Mailpile on Ubuntu 14.04

How To Set Up Shiny Server on Ubuntu 14.04

## 3 Comments

Leave a comment...

Log In to Comment

chrisfs59 *August 8, 2015*

Thanks a bunch for this! Quite useful



daattali *August 8, 2015*

Glad to know if helps, thanks for taking the time to say that :)



drexlar *October 11, 2015*

Congratulations Dean. Excellent tutorial. I followed your tutorial to the letter and I could install R and the packages you mentioned in my Ubuntu 14.04.3. Thank you for sharing this with the community.





This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.



Copyright © 2015 DigitalOcean™ Inc.

[Community](#) [Tutorials](#) [Questions](#) [Projects](#) [Tags](#) [RSS](#) 

---

[Terms, Privacy, & Copyright](#) [Security](#) [Report a Bug](#) [Get Paid to Write](#)