

Bare Metal Cloud now available at **special prices!**

[SEE DISCOUNTS](#)



How To Install Spark On Ubuntu

Posted April 13, 2020

[Home / DevOps and Development / How to Install Spark on Ubuntu](#)

Introduction

Apache Spark is a framework used in cluster computing environments for **analyzing big data**. This platform became widely popular due to its ease of use and the improved data processing speeds over **Hadoop**.

Apache Spark is able to distribute a workload across a group of computers in a cluster to more effectively process large sets of data. This **open-source engine** supports a wide array of programming languages. This includes Java, Scala, Python, and R.

In this tutorial, you will learn **how to install Spark on an Ubuntu machine**. The guide will show you how to start a master and slave server and how to load Scala and Python shells. It also provides the most important Spark



≡ Contents

- 1 Install Packages Required for Spark
- 2 Download and Set Up Spark on Ubuntu
- 3 Configure Spark Environment
- 4 Start Standalone Spark Master Server
- 5 Start Spark Slave Server (Start a Worker Process)
 - 5.1 Specify Resource Allocation for Workers
- 6 Test Spark Shell
- 7 Test Python in Spark
- 8 Basic Commands to Start and Stop Master Server and Workers

Bare Metal Cloud now available at **special prices!**

[SEE DISCOUNTS](#)



How to Install Spark on Ubuntu

A large white arrow points upwards from the title towards the content area.

The Apache Spark logo is displayed below the title.

The background of the slide is blue.

Prerequisites

- An Ubuntu system.
- Access to a terminal or command line.
- A user with sudo or root permissions.

Install Packages Required for Spark

Contents

- 1 Install Packages Required for Spark
- 2 Download and Set Up Spark on Ubuntu
- 3 Configure Spark Environment
- 4 Start Standalone Spark Master Server
- 5 Start Spark Slave Server (Start a Worker Process)
 - 5.1 Specify Resource Allocation for Workers
- 6 Test Spark Shell
- 7 Test Python in Spark
- 8 Basic Commands to Start and Stop Master Server and Workers



Bare Metal Cloud now available at **special prices!**

SEE DISCOUNTS



- JDK
- Scala
- Git

Open a terminal window and run the following command to install all three packages at once:

```
sudo apt install default-jdk scala git -y
```

You will see which packages will be installed.

≡ Contents	
1	Install Packages Required for Spark
2	Download and Set Up Spark on Ubuntu
3	Configure Spark Environment
4	Start Standalone Spark Master Server
5	Start Spark Slave Server (Start a Worker Process)
5.1	Specify Resource Allocation for Workers
6	Test Spark Shell
7	Test Python in Spark
8	Basic Commands to Start and Stop Master Server and Workers





SEE DISCOUNTS

Bare Metal Cloud now available at special prices!

```
test@ubuntu1:~$ sudo apt install default-jdk scala git -y
[sudo] password for test:
Reading package lists... Done
Building dependency tree
Reading state information... Done
git is already the newest version (1:2.17.1-1ubuntu0.5).
The following packages were automatically installed and are no longer required:
liballegro4.4 libdevilic2 libeigen2 libmodplug1 libopenal-data libopenal1
libluajit-5.1-common libmng2 libmodplug1 libopenal-data libopenal1
libphysfs1 libSDL1.2debian libSDL2-2.0-0 vim-runtime
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed: →
ca-certificates-java default-jdk-headless default-jre default-jre-headless
fonts-dejavu-extra java-common libatk-wrapper-java libatk-wrapper-java-jni
libhawtjni-runtime-jar libice-dev libjansi-jar libjansi-native-jar
libjline2-jar libpthread-stubs0-dev libsm-dev libxi11-dev libx11-doc
libxau-dev libxcb1-dev libxdmcp-dev libxt-dev openjdk-11-jdk
openjdk-11-jdk-headless openjdk-11-jre openjdk-11-jre-headless
scala-library scala-parser-combinators scala-xml x11proto-core-dev
x11proto-dev xorg-sgml-doctools xtrans-dev
```

Once the process completes, verify the **installed dependencies** by running these commands:

- 1 Install Packages Required for Spark
- 2 Download and Set Up Spark on Ubuntu
- 3 Configure Spark Environment
- 4 Start Standalone Spark Master Server
- 5 Start Spark Slave Server (Start a Worker Process)
- 5.1 Specify Resource Allocation for Workers
- 6 Test Spark Shell
- 7 Test Python in Spark
- 8 Basic Commands to Start and Stop Master Server and Workers

```
test@ubuntu1:~$ java -version; javac -version; scala -version; git --version
openjdk version "11.0.6" 2020-01-14
OpenJDK Runtime Environment (build 11.0.6+10-post-Ubuntu-1ubuntu18.04.1)
OpenJDK 64-Bit Server VM (build 11.0.6+10-post-Ubuntu-1ubuntu18.04.1, mixed mode, sharing)
javac 11.0.6
Scala code runner version 2.11.12 -- Copyright 2002-2017, LAMP/EPFL
git version 2.17.1
```

Bare Metal Cloud now available at **special prices!**

[SEE DISCOUNTS](#)



Download and Set Up Spark on Ubuntu

Now, you **need to download the version of Spark you want** from their website. We will go for *Spark 3.0.1 with Hadoop 2.7* as it is the latest version at the time of writing this article.

Use the `wget` command and the direct link to download the Spark archive:

```
wget https://downloads.apache.org/spark/spark-3.0.1-bin-hadoop2.7.tgz
```

When the download completes, you will see the saved message.

```
goran@goran-test:~$ wget https://downloads.apache.org/spark/spark-3.0.1-bin-hadoop2.7.tgz
2020-09-14 19:21:23.448995  Resolving downloads.apache.org (downloads.apache.org)... 88.99.95.219
2020-09-14 19:21:23.450000  Connecting to downloads.apache.org (downloads.apache.org)|88.99.95.219|:443... connected
2020-09-14 19:21:23.450000  HTTP request sent, awaiting response... 200 OK
Length: 219929956 (210M) [application/x-gzip]
Saving to: 'spark-3.0.1-bin-hadoop2.7.tgz.1' [=====
2020-09-14 19:22:16 (3.96 MB/s) - 'spark-3.0.1-bin-hadoop2.7.tgz.1' saved [219929956/219929956]
```

Contents	
1	Install Packages Required for Spark
2	Download and Set Up Spark on Ubuntu
3	Configure Spark Environment
4	Start Standalone Spark Master Server
5	Start Spark Slave Server (Start a Worker Process)
5.1	Specify Resource Allocation for Workers
6	Test Spark Shell
7	Test Python in Spark
8	Basic Commands to Start and Stop Master Server and Workers



Bare Metal Cloud now available at **special prices!**

SEE DISCOUNTS



version. Remember to replace the Spark version number in the subsequent commands if you change the download URL.

Now, extract the saved archive using the `tar` command:

```
tar xvf spark-*
```

Let the process complete. The output shows the files that are being unpacked from the archive.

Finally, move the unpacked directory `spark-3.0.1-bin-hadoop2.7` to the ***opt/spark*** directory.

Use the `mv` command to do so:

```
sudo mv spark-3.0.1-bin-hadoop2.7 /opt/spark
```

The terminal returns no response if it successfully moves the directory. If you mistype the name, you will get a message similar to:

```
mv: cannot stat 'spark-3.0.1-bin-hadoop2.7': No such file or directory.
```

Contents

- 1 Install Packages Required for Spark
- 2 Download and Set Up Spark on Ubuntu
- 3 Configure Spark Environment
- 4 Start Standalone Spark Master Server
- 5 Start Spark Slave Server (Start a Worker Process)
 - 5.1 Specify Resource Allocation for Workers
- 6 Test Spark Shell
- 7 Test Python in Spark
- 8 Basic Commands to Start and Stop Master Server and Workers



Bare Metal Cloud now available at **special prices!**

[SEE DISCOUNTS](#)



Before starting a master server, you need to configure environment variables. There are a few Spark home paths you need to add to the user profile.

Use the `echo` command to add these three lines to `.profile`:

```
echo "export SPARK_HOME=/opt/spark" >> ~/.profile
echo "export PATH=$PATH:$SPARK_HOME/bin:$SPARK_HOME/sbin" >> ~/.profile
echo "export PYSPARK_PYTHON=/usr/bin/python3" >> ~/.profile
```

You can also add the export paths by editing the `.profile` file in the editor of your choice, such as nano or vim.

For example, to use nano, enter:

```
nano .profile
```

When the profile loads, scroll to the bottom of the file.

≡ Contents	X
1 Install Packages Required for Spark	
2 Download and Set Up Spark on Ubuntu	
3 Configure Spark Environment	
4 Start Standalone Spark Master Server	
5 Start Spark Slave Server (Start a Worker Process)	
5.1 Specify Resource Allocation for Workers	
6 Test Spark Shell	
7 Test Python in Spark	
8 Basic Commands to Start and Stop Master Server and Workers	



Bare Metal Cloud now available at **special prices!**

[SEE DISCOUNTS](#)



≡ Contents

- 1 Install Packages Required for Spark
- 2 Download and Set Up Spark on Ubuntu
- 3 Configure Spark Environment
- 4 Start Standalone Spark Master Server
- 5 Start Spark Slave Server (Start a Worker Process)
 - 5.1 Specify Resource Allocation for Workers
- 6 Test Spark Shell
- 7 Test Python in Spark
- 8 Basic Commands to Start and Stop Master Server and Workers

Then, add these three lines:

```
export SPARK_HOME=/opt/spark
```

```
export PATH=$PATH:$SPARK_HOME/bin:$SPARK_HOME/sbin
```



Bare Metal Cloud now available at **special prices!**

SEE DISCOUNTS



Exit and save changes when prompted.

When you finish adding the paths, load the `.profile` file in the command line by typing:

```
source ~/ .profile
```

Start Standalone Spark Master Server

Now that you have completed configuring your environment for Spark, you can start a master server.

In the terminal, type:

```
start-master.sh
```

To view the Spark Web user interface, open a web browser and enter the localhost IP address on port 8080.

```
http://127.0.0.1:8080/
```

The page shows your **Spark URL**, status information for workers, hardware resource utilization, etc.



≡ Contents	X
1 Install Packages Required for Spark	
2 Download and Set Up Spark on Ubuntu	
3 Configure Spark Environment	
4 Start Standalone Spark Master Server	
5 Start Spark Slave Server (Start a Worker Process)	
5.1 Specify Resource Allocation for Workers	
6 Test Spark Shell	
7 Test Python in Spark	
8 Basic Commands to Start and Stop Master Server and Workers	

Bare Metal Cloud now available at **special prices!**

SEE DISCOUNTS



≡ Contents

- x 1 Install Packages Required for Spark
- 2 Download and Set Up Spark on Ubuntu
- 3 Configure Spark Environment
- 4 Start Standalone Spark Master Server
- 5 Start Spark Slave Server (Start a Worker Process)
 - 5.1 Specify Resource Allocation for Workers
- 6 Test Spark Shell
- 7 Test Python in Spark
- 8 Basic Commands to Start and Stop Master Server and Workers



<



Bare Metal Cloud now available at **special prices!**

SEE DISCOUNTS

3. `deviceName:8080`



Note: Learn how to automate the deployment of Spark clusters on Ubuntu servers by reading our [Automated Deployment Of Spark Cluster On Bare Metal Cloud](#) article.

≡ Contents X

- 1 Install Packages Required for Spark
- 2 Download and Set Up Spark on Ubuntu
- 3 Configure Spark Environment
- 4 Start Standalone Spark Master Server
- 5 Start Spark Slave Server (Start a Worker Process)
 - 5.1 Specify Resource Allocation for Workers
- 6 Test Spark Shell
- 7 Test Python in Spark
- 8 Basic Commands to Start and Stop Master Server and Workers

Start Spark Slave Server (Start a Worker Process)

In this single-server, standalone setup, we will start one slave server along with the master server.

To do so, run the following command in this format:

```
start-slave.sh spark://master:port
```

The `master` in the command can be an IP or hostname.

In our case it is `ubuntu1`:

```
start-slave.sh spark://ubuntu1:7077
```



Bare Metal Cloud now available at **special prices!**

[SEE DISCOUNTS](#)



Now that a worker is up and running, if you reload Spark Master's Web UI, you should see it on the list:

≡ Contents

- x 1 Install Packages Required for Spark
- 2 Download and Set Up Spark on Ubuntu
- 3 Configure Spark Environment
- 4 Start Standalone Spark Master Server
- v 5 Start Spark Slave Server (Start a Worker Process)
 - 5.1 Specify Resource Allocation for Workers
- 6 Test Spark Shell
- 7 Test Python in Spark
- 8 Basic Commands to Start and Stop Master Server and Workers

Specify Resource Allocation for Workers

The default setting when starting a worker on a machine is to use all available CPU cores. You can specify the number of cores by passing the `-c` flag to the `start-slave` command.





Bare Metal Cloud now available at **special prices!**

SEE DISCOUNTS

```
start-slave.sh -c 1 spark://ubuntu1:7077
```

Reload Spark Master's Web UI to confirm the worker's configuration.

Contents

- 1 Install Packages Required for Spark
- 2 Download and Set Up Spark on Ubuntu
- 3 Configure Spark Environment
- 4 Start Standalone Spark Master Server
- 5 Start Spark Slave Server (Start a Worker Process)
 - 5.1 Specify Resource Allocation for Workers
- 6 Test Spark Shell
- 7 Test Python in Spark
- 8 Basic Commands to Start and Stop Master Server and Workers

Similarly, you can assign a specific amount of memory when starting a worker. The default setting is to use whatever amount of RAM your machine has, minus 1GB.

To start a worker and assign it a specific amount of memory, add the `-m` option and a number. For gigabytes, use **G** and for megabytes, use **M**.

For example, to start a worker with 512MB of memory, enter this command:

```
start-slave.sh -m 512M spark://ubuntu1:7077
```

Reload the Spark Master Web UI to view the worker's status and confirm the configuration.



Bare Metal Cloud now available at **special prices!**

SEE DISCOUNTS



Test Spark Shell

After you finish the configuration and start the master and slave server, test if the Spark shell works.

Load the shell by entering:

```
spark-shell
```

You should get a screen with notifications and Spark information. Scala is the default interface, so that shell loads when you run *spark-shell*.

The ending of the output looks like this for the version we are using at the time of writing this guide:



≡ Contents

- 1 Install Packages Required for Spark
- 2 Download and Set Up Spark on Ubuntu
- 3 Configure Spark Environment
- 4 Start Standalone Spark Master Server
- 5 Start Spark Slave Server (Start a Worker Process)
 - 5.1 Specify Resource Allocation for Workers
- 6 Test Spark Shell
- 7 Test Python in Spark
- 8 Basic Commands to Start and Stop Master Server and Workers

Bare Metal Cloud now available at **special prices!**

SEE DISCOUNTS



≡ Contents

- 1 Install Packages Required for Spark
- 2 Download and Set Up Spark on Ubuntu
- 3 Configure Spark Environment
- 4 Start Standalone Spark Master Server
- 5 Start Spark Slave Server (Start a Worker Process)
 - 5.1 Specify Resource Allocation for Workers
- 6 Test Spark Shell
- 7 Test Python in Spark
- 8 Basic Commands to Start and Stop Master Server and Workers

Type :q and press Enter to exit Scala.

Test Python in Spark

If you do not want to use the default Scala interface, you can switch to Python.

Make sure you quit Scala and then run this command:

```
pyspark
```

The resulting output looks similar to the previous one. Towards the bottom, you will see the version of Python.



Bare Metal Cloud now available at **special prices!**

SEE DISCOUNTS



≡ Contents

- 1 Install Packages Required for Spark
- 2 Download and Set Up Spark on Ubuntu
- 3 Configure Spark Environment
- 4 Start Standalone Spark Master Server
- 5 Start Spark Slave Server (Start a Worker Process)
 - 5.1 Specify Resource Allocation for Workers
- 6 Test Spark Shell
- 7 Test Python in Spark
- 8 Basic Commands to Start and Stop Master Server and Workers

To exit this shell, type `quit()` and hit Enter.

Basic Commands to Start and Stop Master Server and Workers

Below are the basic commands for starting and stopping the Apache Spark master server and workers. Since this setup is only for one machine, the scripts you run default to the localhost.

To start a **master server** instance on the current machine, run the command we used earlier in the guide:

```
start-master.sh
```

To stop the **master** instance started by executing the script above, run:



Bare Metal Cloud now available at **special prices!**

SEE DISCOUNTS

To stop a running worker process, enter this command:

```
stop-slave.sh
```

The Spark Master page, in this case, shows the worker status as DEAD.

≡ Contents

- 1 Install Packages Required for Spark
- 2 Download and Set Up Spark on Ubuntu
- 3 Configure Spark Environment
- 4 Start Standalone Spark Master Server
- 5 Start Spark Slave Server (Start a Worker Process)
 - 5.1 Specify Resource Allocation for Workers
- 6 Test Spark Shell
- 7 Test Python in Spark
- 8 Basic Commands to Start and Stop Master Server and Workers

```
stop-all.sh
```

Similarly, you can stop all instances by using the following command:



Bare Metal Cloud now available at special prices!

SEE DISCOUNTS



This tutorial showed you **how to install Spark on an Ubuntu machine**, as well as the necessary dependencies.

The setup in this guide enables you to perform basic tests before you start configuring a Spark cluster and performing advanced actions.



≡ Contents

- 1 Install Packages Required for Spark
- 2 Download and Set Up Spark on Ubuntu
- 3 Configure Spark Environment
- 4 Start Standalone Spark Master Server
- 5 Start Spark Slave Server (Start a Worker Process)
 - 5.1 Specify Resource Allocation for Workers
- 6 Test Spark Shell
- 7 Test Python in Spark
- 8 Basic Commands to Start and Stop Master Server and Workers

Next you should also read

✉ DevOps And Development,

Databases

How to Install Elasticsearch on Ubuntu 18.04

April 23, 2020

Deploying MySQL in a container is a fast and efficient solution for small Elasticsearch is an open-source engine that enhances searching,

✉ Virtualization, Databases, MySQL

MySQL Docker Container Tutorial: How to Set Up & Configure

February 10, 2020

✉ SysAdmin, Databases, I

How to Improve Performance With Tuning

January 15, 2020

PostgreSQL is the third most popular Docker image used for deploying

The performance of MySQL databases is an essential factor in the





Bare Metal Cloud now available at special prices!

SEE DISCOUNTS

READ MORE

RFID MORF

RFID MORF

RFID MORF

Contents

- 1 Install Packages Required for Spark
- 2 Download and Set Up Spark on Ubuntu
- 3 Configure Spark Environment
- 4 Start Standalone Spark Server
- 5 Start Spark Slave Server (as a Worker Process)
 - 5.1 Specify Resource Allocation for Workers
- 6 Test Spark Shell
- 7 Test Python in Spark
- 8 Basic Commands to Start Stop Master Server and Workers

Author

Goran Jevtic

Goran combines his passions for research, writing and technology as a technical writer at phoenixNAP. Working in multiple departments and on a variety of projects, he has developed extraordinary understanding of cloud and virtualization technology trends and best practices.

RECENT POSTS

SYSADMIN	BARE METAL SERVERS	ABOUT US	COLOCATION	EVENTS
VIRTUALIZATION	WEB SERVERS	GITHUB	SERVERS	PRESS
DEVS AND DEVELOPMENT	NETWORKING	BLOG	CLOUD	CONTACT US
SECURITY	DATABASES	RFP TEMPLATE	SERVICES	SOLUTIONS
		CAREERS		CAREERS
		BACKUP AND		LOCATIONS

Bare Metal Cloud now available at **special prices!**

SEE DISCOUNTS



Live Chat Get a Quote Support | 1-855-330-1509 Sales | 1-877-588-5918



[Privacy Policy](#) [GDPR](#) [Sitemap](#)

≡ Contents

- 1 Install Packages Required for Spark
- 2 Download and Set Up Spark on Ubuntu
- 3 Configure Spark Environment
- 4 Start Standalone Spark Master Server
- 5 Start Spark Slave Server (Start a Worker Process)
 - 5.1 Specify Resource Allocation for Workers
- 6 Test Spark Shell
- 7 Test Python in Spark
- 8 Basic Commands to Start and Stop Master Server and Workers

