

# Java-Success.com

Prepare to fast-track, choose & go places with 800+ Java & Big Data Q&As with lots of code & diagrams.

[Home](#) [Why? ▾](#) [300+ Java FAQs ▾](#) [300+ Big Data FAQs ▾](#) [Courses ▾](#)

[👤 Membership ▾](#) [Your Career ▾](#)

[Home](#) › [bigdata-success.com](#) › [Tutorials - Big Data](#) › [TUT - Cloudera on Docker](#) › 22:

Docker Tutorial: Apache Spark (spark-submit) in Python 2.6 on Cloudera quickstart

## 22: Docker Tutorial: Apache Spark (spark-submit) in Python 2.6 on Cloudera quickstart

 Posted on [June 9, 2019](#)

Extends [20: Docker Tutorial: Apache Spark-submit in Java – on Cloudera quickstart](#), and [Docker Tutorial: BigData on Cloudera quickstart via Docker](#).

**Step 1:** Run the container on a command line.

```
1 ~/projects/docker-hadoop]$ docker run --hostname=q
2 --privileged=true -t -i -v /Users/arulkumarankumar
3 --publish-all=true -p 8888:8888 -p 80:80 -p 7180:7
```

### 300+ Java Interview FAQs

300+ Java FAQs



16+ Java Key Areas Q&As



150+ Java Architect FAQs



80+ Java Code Quality Q&As



150+ Java Coding Q&As



### 300+ Big Data Interview FAQs

300+ Big Data FAQs



Tutorials - Big Data



TUT -  Starting Big Data

TUT - Starting Spark & Scala

## Python version

**Step 2:** The cloudera/quickstart comes with Python 2.6.

```
1 [root@quickstart /]# python --version
2 Python 2.6.6
3 [root@quickstart /]#
4
```

## Create SimpleSpark.py

```
1 [root@quickstart ~]# vi SimpleSpark.py
1 from pyspark import SparkConf, SparkContext
2
3 def main(sc):
4     rdd = sc.parallelize(["John", "Peter", "Samuel"])
5     print(rdd.collect())
6
7 if __name__ == "__main__":
8     conf = SparkConf().setAppName("Simple App")
9     conf = conf.setMaster("local[*]")
10    sc = SparkContext(conf=conf)
11    main(sc)
12
```

## spark-submit

**Step 3:** Submit the job to Spark.

```
1 [root@quickstart ~]# spark-submit /root/SimpleSpark.py
2
1 ...
2 ['John', 'Peter', 'Samuel']
3 .....
4
```

Let's do the industrial way with packages or modules, egg (i.e. zip) files, etc.

TUT - Starting with Python

TUT - Kafka

TUT - Pig

TUT - Apache Storm

TUT - Spark Scala on Zeppelin

TUT - Cloudera

TUT - Cloudera on Docker

TUT - File Formats

TUT - Spark on Docker

TUT - Flume

TUT - Hadoop (HDFS)

TUT - HBase (NoSQL)

TUT - Hive (SQL)

TUT - Hadoop & Spark

TUT - MapReduce

TUT - Spark and Scala

TUT - Spark & Java

TUT - PySpark on Databricks

TUT - Zookeeper

## 800+ Java Interview Q&As

300+ Core Java Q&As



300+ Enterprise Java Q&As



150+ Java Frameworks Q&As



120+ Companion Tech Q&As



Tutorials - Enterprise Java



## Install pip

**Step 1:** PIP is a package manager for Python packages, or modules. Similar to maven for Java & sbt for Scala where you can manage the dependencies.

```
1 [root@quickstart ~]# yum install python-pip
1 [root@quickstart ~]# pip --version
2 pip 7.1.0 from /usr/lib/python2.6/site-packages (py
3 [root@quickstart ~]#
4
```

## Create a Python project structure

**Step 2:** Create the project structure and the relevant python files.

```
1 [root@quickstart projects]# mkdir -p /root/projects
2 [root@quickstart projects]# cd my-app
3
```

### simple.py

**Step 3:** Create the packages (aka modules) and .py files.

```
1 [root@quickstart my-app]# vi mypackage/simple.py
1 from pyspark import SparkConf, SparkContext
2
3 class SimpleSpark:
4
5     def myfunc(self, sc):
6         rdd = sc.parallelize(["John", "Peter", "Sam"])
7         print(rdd.collect())
8
```

### driver.py

```
1 [root@quickstart my-app]# vi driver.py
2
1 from pyspark import SparkConf, SparkContext
2 from mypackage import simple
3
4 if __name__ == "__main__":
5     conf = SparkConf().setAppName("Simple App")
6     conf = conf.setMaster("local[*]")
7     sc = SparkContext(conf=conf)
8     simple.SimpleSpark().myfunc(sc)
9
```

## setup.py

**Step 4:** setup.py to build .egg (i.e. zip) files containing all the modules.

setup.py is a python file, which usually tells you that the module/package you are about to install has been packaged and distributed with Distutils, which is the standard for distributing Python Modules.

```
1 [root@quickstart my-app]# vi setup.py
2
1 from setuptools import setup
2
3 setup(
4     name = 'simple-spark',
5     author = 'java-success',
6     packages=['mypackage'],
7     # Whatever arguments you need/want
8 )
9
```

```
1 [root@quickstart my-app]# tree
2 .
3 |— driver.py
4 |— mypackage
5 |   |— simple.py
6 |— setup.py
7
8 1 directory, 3 files
9 [root@quickstart my-app]#
10
```

## Build an .egg file

**Step 5:** Let's build an .egg file with setup.py.

```

1 [root@quickstart my-app]# python setup.py install
2
3 [root@quickstart my-app]# tree
4 .
5 |— build
6 |   |— bdist.linux-x86_64
7 |   |— lib
8 |       |— mypackage
9 |           |— simple.py
10 |— dist
11 |   |— simple_spark-0.0.0-py2.6.egg
12 |— driver.py
13 |— mypackage
14 |   |— simple.py
15 |— setup.py
16 |— simple_spark.egg-info
17 |   |— dependency_links.txt
18 |   |— PKG-INFO
19 |   |— SOURCES.txt
20 |   |— top_level.txt
21
22 7 directories, 9 files
23 [root@quickstart my-app]#
24

```

## View an .egg file

**Step 6:** egg file is basically a zip file.

```

1 [root@quickstart my-app]# unzip -l dist/simple_spark-0.0.0-py2.6.egg
2 Archive:  dist/simple_spark-0.0.0-py2.6.egg
3   Length      Date    Time    Name
4  -----
5         10  06-09-2019  04:17  EGG-INFO/top_level.txt
6          1  06-09-2019  04:17  EGG-INFO/zip-safe
7        172  06-09-2019  04:17  EGG-INFO/SOURCES.txt
8          1  06-09-2019  04:17  EGG-INFO/dependency_links.txt
9        191  06-09-2019  04:17  EGG-INFO/PKG-INFO
10       224  06-09-2019  04:17  mypackage/__init__.py
11         55  06-09-2019  04:17  mypackage/__init__.py
12       679  06-09-2019  04:17  mypackage/simple.py
13       181  06-09-2019  04:13  mypackage/simple.py
14  -----

```

```
15 1514 9 files
16 [root@quickstart my-app]#
17
```

## spark-submit

**Step 7:** driver.py file gets invoked, which uses modules in the .egg file. “-verbose” is optional.

```
1 [root@quickstart my-app]# spark-submit \
2 --verbose \
3 --py-files dist/simple_spark-0.0.0-py2.6.egg \
4 driver.py
5
```

### Output:

```
1 ...
2 ['John', 'Peter', 'Samuel']
3 .....
4
```

## View the packages via pip

**Step 8:** You can also see the “simple-spark (0.0.0)”, and the “setuptools (0.6rc11)” used in the setup.py.

```
1 [root@quickstart my-app]# pip list
2 You are using pip version 7.1.0, however version 1
3 You should consider upgrading via the 'pip instal
4 argparse (1.4.0)
5 distribute (0.6.10)
6 iniparse (0.3.1)
7 iotop (0.3.2)
8 MySQL-python (1.2.3rc1)
9 ordereddict (1.2)
10 pip (7.1.0)
11 psycpg2 (2.0.14)
12 pycurl (7.19.0)
13 pygpgme (0.1)
14 python-snappy (0.5)
15 setuptools (0.6rc11)
16 simple-spark (0.0.0)
```

```

17 urlgrabber (3.9.1)
18 yum-metadata-parser (1.1.2)
19 [root@quickstart my-app]#
20

```

## python global sitepackages

```

1 [root@quickstart my-app]# whereis python
2 python: /usr/bin/python2.6 /usr/bin/python2.6-conf
3

```

```

1 [root@quickstart my-app]# ls -ltr /usr/lib/python2.6
2 total 368
3 -rw-r--r-- 1 root root 1085 Aug 17 2010 inipars
4 -rw-r--r-- 1 root root 2362 Nov 12 2010 site.py
5 -rw-r--r-- 1 root root 88196 Nov 12 2010 pkg_res
6 -rw-r--r-- 1 root root 126 Nov 12 2010 easy_in
7 -rw-r--r-- 2 root root 1771 Nov 12 2010 site.py
8 -rw-r--r-- 2 root root 1771 Nov 12 2010 site.py
9 -rw-r--r-- 1 root root 34 Nov 12 2010 setupto
10 -rw-r--r-- 1 root root 144 Nov 12 2010 setupto
11 -rw-r--r-- 2 root root 92875 Nov 12 2010 pkg_res
12 -rw-r--r-- 2 root root 92875 Nov 12 2010 pkg_res
13 -rw-r--r-- 2 root root 317 Nov 12 2010 easy_in
14 -rw-r--r-- 2 root root 317 Nov 12 2010 easy_in
15 -rw-r--r-- 1 root root 2285 Jul 30 2013 urlgrab
16 drwxr-xr-x 2 root root 4096 Mar 4 2015 urlgrab
17 drwxr-xr-x 2 root root 4096 Mar 4 2015 inipars
18 drwxr-xr-x 2 root root 4096 Mar 4 2015 yum
19 drwxr-xr-x 2 root root 4096 Mar 4 2015 rpmUtil
20 drwxr-xr-x 3 root root 4096 Apr 6 2016 setupto
21 drwxr-xr-x 2 root root 4096 Apr 6 2016 distrib
22 drwxr-xr-x 2 root root 4096 Apr 6 2016 iotop-0
23 drwxr-xr-x 2 root root 4096 Apr 6 2016 iotop
24 drwxr-xr-x 3 root root 4096 Apr 6 2016 python
25 drwxr-xr-x 3 root root 4096 Apr 6 2016 argpars
26 drwxr-xr-x 2 root root 4096 Jun 9 15:46 pip-7.1
27 drwxr-xr-x 10 root root 4096 Jun 9 15:46 pip
28 drwxr-xr-x 4 root root 4096 Jun 9 15:55 simple
29 -rw-r--r-- 1 root root 286 Jun 9 15:55 easy-in
30

```

You can see **simple\_spark-0.0.0-py2.6.egg**.

◀ 21: Docker Tutorial: Apache Spark (spark-submit) in Scala on Cloudera quickstart

23: Docker Tutorial: Apache Spark (spark-submit) in Python 3 with virtual  
env on Cloudera quickstart >

## Disclaimer

The contents in this Java-Success are copyrighted and from EmpoweringTech pty ltd. The EmpoweringTech pty ltd has the right to correct or enhance the current content without any prior notice. These are general advice only, and one needs to take his/her own circumstances into consideration. The EmpoweringTech pty ltd will not be held liable for any damages caused or alleged to be caused either directly or indirectly by these materials and resources. Any trademarked names or labels used in this blog remain the property of their respective trademark owners. Links to external sites do not imply endorsement of the linked-to sites. [Privacy Policy](#)

© 2022 [java-success.com](https://www.java-success.com)