

# 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](#) > 23:

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

Cloudera quickstart

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

 Posted on [June 15, 2019](#)

**Prerequisite:** Docker is installed on your Windows or Mac, and you have a basic understanding of Docker.

[Docker tutorials step by step](#) | [Hadoop, Hive, Impala & Spark on Cloudera quickstart on Docker tutorials](#)

**Step 1:** Pull the modified image “gdancik/cloudera” of cloudera/quickstart with python3.4 & vim installed.

### 300+ Java Interview FAQs

300+ Java FAQs



16+ Java Key Areas Q&amp;As



150+ Java Architect FAQs



80+ Java Code Quality Q&amp;As



150+ Java Coding Q&amp;As



### 300+ Big Data Interview FAQs

300+ Big Data FAQs

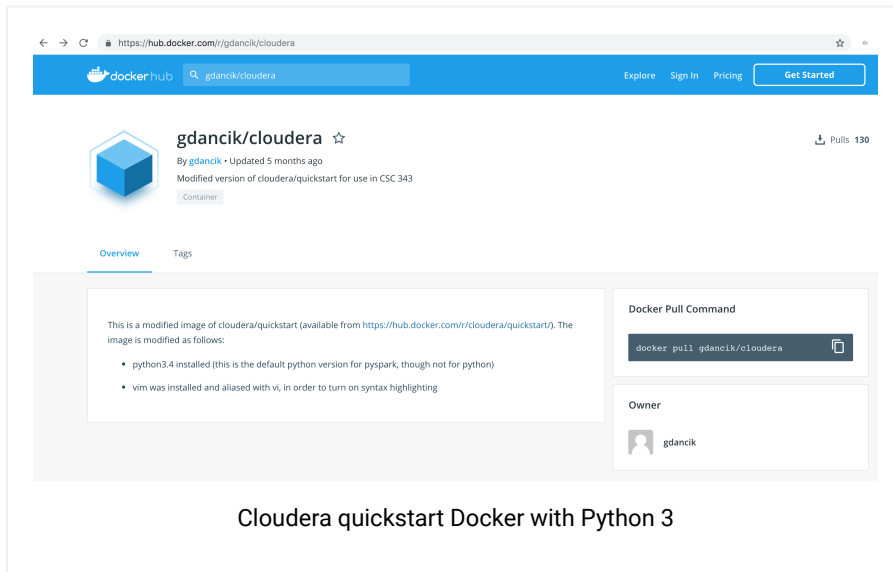


Tutorials - Big Data

TUT -  Starting Big Data

TUT - Starting Spark &amp; Scala

vim is aliased with “vi”. This image is available via Docker hub –  
<https://hub.docker.com/r/gdancik/cloudera>.



```
1 | $ docker pull gdancik/cloudera
```

**Step 2:** Run the container on a command line.

```
1 | ~/projects/docker-hadoop]$ docker run --hostname=qu
2 | --privileged=true -t -i -v /Users/arulkumarankumar
3 | --publish-all=true -p 8888:8888 -p 80:80 -p 7180:71
```

## Python3

**Step 3:** Configure python3.

Image “gdancik/cloudera” comes with python3.

```
1 | [root@quickstart /]# python3 --version
2 | Python 3.4.8
3 | [root@quickstart /]#
```

To use Python3 for pyspark:

```
1 |
```

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 

```
1 [root@quickstart ~]# export PYSARK_PYTHON=python3
2
```

If you want python to point python3

```
1 [root@quickstart ~]# alias python=python3
2 [root@quickstart ~]# python --version
3 Python 3.4.8
4 [root@quickstart ~]#
5
```

## Install pip3

Step 4: Install pip3.

```
1 [root@quickstart /]# sudo yum install python34-setuptools
2 [root@quickstart /]# sudo easy_install-3.4 pip
3
```

```
1 [root@quickstart /]# pip3 list
2 DEPRECATION: Python 3.4 support has been deprecated
3 Package      Version
4 -----
5 pip          19.1.1
6 setuptools   19.6.2
7 [root@quickstart /]#
8
```

```
1 [root@quickstart /]# pip3 freeze
2
```

## Virtual Environment

Step 5: Install virtualenv.

```
1 [root@quickstart ~]# sudo pip3 install virtualenv
```

```
1 [root@quickstart ~]# which virtualenv
2 /usr/bin/virtualenv
3
```

```
1 [root@quickstart ~]# pip3 freeze
```

```
2 | virtualenv==16.6.0
3 |
1 | [root@quickstart ~]# pip3 list
2 | Package      Version
3 | -----
4 | pip           19.1.1
5 | setuptools    19.6.2
6 | virtualenv     16.6.0
7 |
```

### Step 6: Create “projects/my-app” directory

```
1 | [root@quickstart projects]# mkdir -p /root/projects/my-app
2 | [root@quickstart projects]# cd !$
3 | cd /root/projects/my-app
4 | [root@quickstart my-app]#
5 |
```

### Step 7: Create a virtual environment named “my-app-env”.

```
1 | [root@quickstart my-app]# python3 -m venv my-app-env
2 | [root@quickstart my-app]# ls -ltr
3 | total 4
4 | drwxr-xr-x 5 root root 4096 Jun 15 05:20 my-app-env
5 | [root@quickstart my-app]#
6 |
```

### Step 8: Activate the virtual environment.

```
1 | [root@quickstart my-app]# source my-app-env/bin/activate
2 | (my-app-env) [root@quickstart my-app]#
3 |
```

(my-app-env) means we are in “my-app-env” virtual environment. So if you install a package like say “pytest” it will be installed in the virtual environment site packages and not in the global site packages.

```
1 (my-app_env) [root@quickstart my-app]# pip3 install
2 ....
3 (my-app_env) [root@quickstart my-app]# pip3 freeze
4 atomicwrites==1.3.0
5 attrs==19.1.0
6 importlib-metadata==0.18
7 more-itertools==7.0.0
8 packaging==19.0
9 pathlib2==2.3.3
10 pluggy==0.12.0
11 py==1.8.0
12 pyparsing==2.4.0
13 pytest==4.6.3
14 scandir==1.10.0
15 six==1.12.0
16 wcwidth==0.1.7
17 zipp==0.5.1
18
```

```
1 (my-app_env) [root@quickstart my-app]# ls -ltr my-c
```

## switch to global

```
1 (my-app_env) [root@quickstart my-app]# deactivate
2 [root@quickstart my-app]#
3
```

```
1 [root@quickstart my-app]# pip3 freeze
2 virtualenv==16.6.0
3 [root@quickstart my-app]#
4
```

## Switch back to virtual env

“history” command to re-run the “source” command with “!”

```
1 [root@quickstart my-app]# history | grep source
2 35 source my-app_env/bin/activate
3 51 history | grep source
4 [root@quickstart my-app]# !35
5 source my-app_env/bin/activate
6 (my-app_env) [root@quickstart my-app]#
7
```

## Create a Python project structure

## Step 9: Create the project structure and the relevant python files.

```
1 (my-app_env) [root@quickstart my-app]# mkdir -p /root/my-app
```

### simple.py

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

### driver.py

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

### setup.py

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 (my-app_env) [root@quickstart my-app]# vi setup.py
2
3
4
5
6
7
8
9
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
```

## tree -L 4

```
1 (my-app_env) [root@quickstart my-app]# tree -L 5
2 .
3 |__ driver.py
4 |__ my-app_env
5 |   |__ bin
6 |   |   |__ activate
7 |   |   |__ activate.csh
8 |   |   |__ activate.fish
9 |   |   |__ easy_install
10 |   |   |__ easy_install-3.4
11 |   |   |__ pip
12 |   |   |__ pip3
13 |   |   |__ pip3.4
14 |   |   |__ pytest
15 |   |   |__ py.test
16 |   |   |__ python -> python3
17 |   |   |__ python3 -> /usr/bin/python3
18 |   |__ include
19 |   |__ lib
20 |   |   |__ python3.4
21 |   |       |__ site-packages
22 |   |           |__ atomicwrites
23 |   |           |__ atomicwrites-1.3.0.dist-info
24 |   |           |__ attr
25 |   |           |__ attrs-19.1.0.dist-info
26 |   |           |__ easy_install.py
27 |   |           |__ importlib_metadata
28 |   |           |__ importlib_metadata-0.18.dist-info
29 |   |           |__ more_itertools
30 |   |           |__ more_itertools-7.0.0.dist-info
31 |   |           |__ packaging
32 |   |           |__ packaging-19.0.dist-info
```

```

33 | | | | | pathlib2
34 | | | | | pathlib2-2.3.3.dist-info
35 | | | | | pip
36 | | | | | pip-9.0.1.dist-info
37 | | | | | pkg_resources
38 | | | | | pluggy
39 | | | | | pluggy-0.12.0.dist-info
40 | | | | | py
41 | | | | | py-1.8.0.dist-info
42 | | | | | __pycache__
43 | | | | | pyparsing-2.4.0.dist-info
44 | | | | | pyparsing.py
45 | | | | | _pytest
46 | | | | | pytest-4.6.3.dist-info
47 | | | | | pytest.py
48 | | | | | scandir-1.10.0-py3.4.egg-info
49 | | | | | scandir.py
50 | | | | | setuptools
51 | | | | | setuptools-28.8.0.dist-info
52 | | | | | six-1.12.0.dist-info
53 | | | | | six.py
54 | | | | | wcwidth
55 | | | | | wcwidth-0.1.7.dist-info
56 | | | | | zipp-0.5.1.dist-info
57 | | | | | zipp.py
58 | | | | |
59 | | | | | lib64 -> lib
60 | | | | | pip-selfcheck.json
61 | | | | | pyvenv.cfg
62 | | | | |
63 | | | | | mypackage
64 | | | | | | simple.py
65 | | | | | | setup.py
66 | | | | |
67 | | | | |

```

38 directories, 23 files

(my-app\_env) [root@quickstart my-app]#

## Build an .egg file

**Step 10:** Let's build an .egg file with setup.py in a virtual environment.

```

1 | (my-app_env) [root@quickstart my-app]# python setup.py build
2 |

```

```

1 | (my-app_env) [root@quickstart my-app]# tree -L 2
2 | .
3 | | build
4 | | | bdist.linux-x86_64
5 | | | lib
6 | | dist

```



```
7 | └─ simple_spark-0.0.0-py3.4.egg
8 | └─ driver.py
9 | └─ my-app_env
10 |   └─ bin
11 |   └─ include
12 |   └─ lib
13 |   └─ lib64 -> lib
14 |   └─ pip-selfcheck.json
15 |   └─ pyvenv.cfg
16 | └─ mypackage
17 |   └─ simple.py
18 | └─ setup.py
19 | └─ simple_spark.egg-info
20 |   └─ dependency_links.txt
21 |   └─ PKG-INFO
22 |   └─ SOURCES.txt
23 |   └─ top_level.txt
24 |
25 | 11 directories, 10 files
26 | (my-app_env) [root@quickstart my-app]#
27 |
```

## View an .egg file

```
1 | (my-app_env) [root@quickstart my-app]# unzip -l dist/
2 | Archive:  dist/simple_spark-0.0.0-py3.4.egg
3 |   Length      Date    Time    Name
4 |   -----
5 |          10  06-15-2019 07:53  EGG-INFO/top_level.txt
6 |           1  06-15-2019 07:53  EGG-INFO/zip-safe
7 |         172  06-15-2019 07:53  EGG-INFO/SOURCES.txt
8 |           1  06-15-2019 07:53  EGG-INFO/dependency_links.txt
9 |         191  06-15-2019 07:53  EGG-INFO/PKG-INFO
10 |         180  06-15-2019 07:41  mypackage/simple.py
11 |         579  06-15-2019 07:53  mypackage/__pycache__/
12 |   -----
13 |        1134                      7 files
14 | (my-app_env) [root@quickstart my-app]#
15 |
```

## spark-submit

```
1 | (my-app_env) [root@quickstart my-app]# spark-submit
2 | --verbose \
3 | --py-files dist/simple_spark-0.0.0-py3.4.egg \
4 | driver.py
5 |
```

## Outputs:

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

## Create a requirements.txt file

```
1 (my-app_env) [root@quickstart my-app]# pip3 freeze
```

```
1 (my-app_env) [root@quickstart my-app]# tree -L 1
2 .
3 |— build
4 |— dist
5 |— driver.py
6 |— my-app_env
7 |— mypackage
8 |— requirements.txt
9 |— setup.py
10 |— simple_spark.egg-info
11
12 5 directories, 3 files
13 (my-app_env) [root@quickstart my-app]#
14
```

```
1 (my-app_env) [root@quickstart my-app]# cat requirements.txt
2 atomicwrites==1.3.0
3 attrs==19.1.0
4 importlib-metadata==0.18
5 more-itertools==7.0.0
6 packaging==19.0
7 pathlib2==2.3.3
8 pluggy==0.12.0
9 py==1.8.0
10 pyparsing==2.4.0
11 pytest==4.6.3
12 scandir==1.10.0
13 simple-spark==0.0.0
14 six==1.12.0
15 wcwidth==0.1.7
16 zipp==0.5.1
17
```

## Switch to global environment

```
1 (my-app_env) [root@quickstart my-app]# deactivate
2 [root@quickstart my-app]#
3
1 [root@quickstart my-app]# pip3 freeze
2 virtualenv==16.6.0
3
```

Let's install the packages from "requirements.txt" file.  
Remove the line "simple-spark==0.0.0".

```
1 [root@quickstart my-app]# pip3 install -r requirements.txt
2
```

Now the global environment will have the following package dependencies.

```
1 [root@quickstart my-app]# pip3 freeze
2 atomicwrites==1.3.0
3 attrs==19.1.0
4 importlib-metadata==0.18
5 more-itertools==7.0.0
6 packaging==19.0
7 pathlib2==2.3.3
8 pluggy==0.12.0
9 py==1.8.0
10 pyparsing==2.4.0
11 pytest==4.6.3
12 scandir==1.10.0
13 six==1.12.0
14 virtualenv==16.6.0
15 wcwidth==0.1.7
16 zipp==0.5.1
17 [root@quickstart my-app]#
18
```

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

24: Docker Tutorial: HBase (i.e. NoSQL DB) Shell 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)