

# 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](#) > 29:

Docker Tutorial: Apache Kafka with Python 3 on Cloudera quickstart

## 29: Docker Tutorial: Apache Kafka with Python 3 on Cloudera quickstart

 Posted on [July 1, 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. vim is aliased with “vi”. This image is available via Docker hub – <https://hub.docker.com/r/gdancik/cloudera>.

### 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

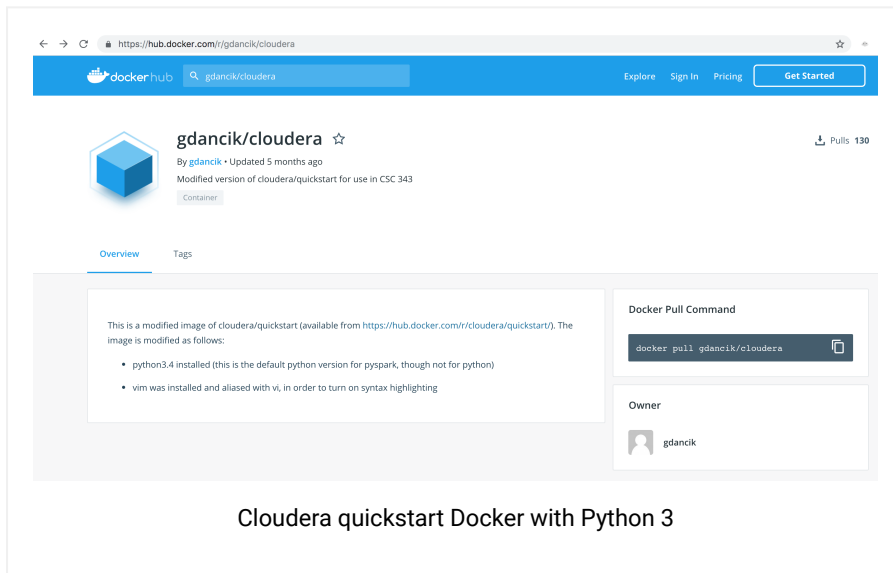


Tutorials - Big Data



TUT -  Starting Big Data

TUT - Starting Spark & Scala



```
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 7180:7
4 | -p 8042:8042 gdancik/cloudera /usr/bin/docker-quick
```

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

If you want python to point python3

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 ~]# 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
```

## Install Kafka

**Step 5:** Add new user, add to sudoers & set password.

```
1 [root@quickstart /]# sudo useradd -m kafka
2
```

```
1 [root@quickstart /]# visudo
2
```

Uncomment the following line in the file “/etc/sudoers”. You can’t edit this file with any other editor like vi. At the top of the file you will see the statement that “This file MUST be edited with the ‘visudo’ command as root.”

```
1 ....
2 ## Allows people in group wheel to run all commands
3 %wheel ALL=(ALL) ALL
4
5 ....
6
```

```
1 [root@quickstart /]# usermod -aG wheel kafka
```

Set the password.

```
1 [root@quickstart /]# sudo passwd kafka
```

## Install Kafka binaries

Step 6: Install Kafka.

```
1 [root@quickstart /]# su kafka
2 [kafka@quickstart /]$ cd ~
3 [kafka@quickstart ~]$ pwd
4 /home/kafka
5
```

```
1 [kafka@quickstart ~]$ sudo yum install wget
2
```

```
1 [kafka@quickstart ~]$ wget https://www.apache.org/d
2 [kafka@quickstart ~]$ tar -xvzf kafka_2.11-2.1.1.t
3 [kafka@quickstart ~]$ ln -s kafka_2.11-2.1.1 kafka
4
```

```
1 [kafka@quickstart ~]$ tree -L 2
2 .
3 ├── kafka -> kafka_2.11-2.1.1
4 ├── kafka_2.11-2.1.1
5 |   ├── bin
6 |   ├── config
7 |   ├── libs
8 |   ├── LICENSE
9 |   ├── NOTICE
10 |   └── site-docs
11 └── kafka_2.11-2.1.1.tgz
12
```

# Install Java 8

**Step 7:** Install Java 8 as Kafka requires Java.

Switch to root user. The password is “cloudera”.

```
1 [kafka@quickstart ~]$ su -
2 Password:
3 [root@quickstart ~]#
4 [root@quickstart ~]# pwd
5 /root
6
```

The **java-1.8.0-openjdk** package contains just the Java Runtime Environment. If you want to develop Java programs then install the **java-1.8.0-openjdk-devel** package.

```
1 [root@quickstart ~]# yum install java-1.8.0-openjdk
2
1 [root@quickstart ~]# [root@quickstart ~]# vi ~/.bashrc
2
```

Add the following lines to the end of “~/.bashrc”.

```
1 ...
2 export JAVA_HOME=/usr/lib/jvm/jre-1.8.0-openjdk.x86_64
3 export PATH=$JAVA_HOME/bin:$PATH
4
```

Activate the change.

```
1 [root@quickstart ~]# source ~/.bashrc
2
1 [root@quickstart ~]# java -version
2 openjdk version "1.8.0_212"
3 OpenJDK Runtime Environment (build 1.8.0_212-b04)
```

```
4 OpenJDK 64-Bit Server VM (build 25.212-b04, mixed mode)
5
```

## Start the Kafka server

Switch user to “kafka” and cd to home dir.

```
1 [root@quickstart ~]# su kafka
2 [kafka@quickstart root]$ cd ~
3 [kafka@quickstart ~]$ mkdir -p kafka/logs
4
```

**Step 8:** Start the Kafka server.

```
1 [kafka@quickstart ~]$ ./kafka/bin/kafka-server-start.sh config/server.properties
2 [1] 6189
3
```

```
1 [kafka@quickstart ~]$ netstat -tulpn | grep 9092
2 (Not all processes could be identified, non-owned pids may
3 will not be shown, you would have to be root to see those
4 tcp        0      0 0.0.0.0:9092          0.0.0.0:*
5 [kafka@quickstart ~]$
6
```

## Create a new topic

**Step 9:** Create a new topic named “MyTestTopic”

```
1 [kafka@quickstart ~]$ ./kafka/bin/kafka-topics.sh --create --zookeeper
2 --list localhost:9092 \
3 MyTestTopic
4
```

## Virtual Environment

Switch user to root. “cloudera” is the password for the root user.

```
1 [kafka@quickstart ~]$ su -
2 Password:
3 [root@quickstart ~]#
4
```

### Step 10: Install virtualenv.

```
1 [root@quickstart ~]# sudo pip install virtualenv
2
1 [root@quickstart ~]# which virtualenv
2 /usr/bin/virtualenv
3 [root@quickstart ~]#
4
1 [root@quickstart ~]# pip freeze
2 DEPRECATION: Python 3.4 support has been deprecated
3 virtualenv==16.6.1
4 [root@quickstart ~]#
5
1 [root@quickstart ~]# pip list
2 DEPRECATION: Python 3.4 support has been deprecated
3 Python 3.4 won't be maintained after March 2019 (c
4 Package      Version
5 -----
6 pip          19.1.1
7 setuptools   19.6.2
8 virtualenv    16.6.1
9 [root@quickstart ~]#
10
```

### Step 11: Create "projects/my-app" directory

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

### Step 12: Create a virtual environment named "my-app-env".

```
1 [root@quickstart my-app]# python3 -m venv my-app-env
2
```

```
1 [root@quickstart my-app]# tree -L 3
2 .
3 └── my-app_env
4     ├── bin
5     │   ├── activate
6     │   ├── activate.csh
7     │   ├── activate.fish
8     │   ├── easy_install
9     │   ├── easy_install-3.4
10    │   ├── pip
11    │   ├── pip3
12    │   ├── pip3.4
13    │   ├── python -> python3
14    │   └── python3 -> /usr/bin/python3
15    ├── include
16    ├── lib
17    │   └── python3.4
18    ├── lib64 -> lib
19    └── pyvenv.cfg
20
21 6 directories, 11 files
22
```

**Step 13:** 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.

## install kafka-python

**Step 14:** Install kafka-python library in the virtual environment.

```
1 (my-app_env) [root@quickstart my-app]# sudo pip install kafka-python
2
3 (my-app_env) [root@quickstart my-app]# pip freeze
4 kafka-python==1.4.6
5
6 (my-app_env) [root@quickstart my-app]# pip list
7 DEPRECATION: The default format will switch to columns in the future.
8 kafka-python (1.4.6)
```



```
9 | pip (9.0.1)
10 | setuptools (28.8.0)
11 |
```

## Create a Python project structure

**Step 15:** Create the project structure and the relevant python files.

```
1 | (my-app_env) [root@quickstart my-app]# mkdir -p /p
2 |
```

### simple\_producer.py

```
1 | (my-app_env) [root@quickstart my-app]# vi mypackage
2 |
3 |
4 |
5 | 1 | from kafka import KafkaProducer
6 | 2 |
7 | 3 | class SimpleProducer:
8 | 4 |
9 | 5 |     def myfunc(self):
10 | 6 |         producer = KafkaProducer(bootstrap_servers=
11 | 7 |         for _ in range(10):
12 | 8 |             producer.send('MyTestTopic',b'some byt
13 | 9 |
14 | 10 |         producer.flush()
15 | 11 |
```

### driver.py

```
1 | (my-app_env) [root@quickstart my-app]# vi driver.py
2 |
3 |
4 |
5 | 1 | from mypackage import simple_producer
6 | 2 |
7 | 3 | if __name__ == "__main__":
8 | 4 |     simple_producer.SimpleProducer().myfunc()
9 | 5 |
10 | 6 |
```

### 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
1 from setuptools import setup
2
3 setup(
4     name = 'simple-producer',
5     author = 'java-success',
6     packages=['mypackage'],
7     # Whatever arguments you need/want
8 )
9
```

## Build .egg file

```
1 (my-app_env) [root@quickstart my-app]# python setup.py build
2
1 (my-app_env) [root@quickstart my-app]# pip list
2 DEPRECATION: The default format will switch to columns in the future.
3 kafka-python (1.4.6)
4 pip (9.0.1)
5 setuptools (28.8.0)
6 simple-producer (0.0.0)
7
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-producer-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-producer.py
```

```
18 | └─ setup.py
19 | └─ simple_producer.egg-info
20 |     └─ dependency_links.txt
21 |     └─ PKG-INFO
22 |     └─ SOURCES.txt
23 |     └─ top_level.txt
24 |
```

## Run the producer via python code

```
1 (my-app_env) [root@quickstart my-app]# python driver.py
2
```

## Run the consumer

```
1 (my-app_env) [root@quickstart my-app]# /home/kafka/bin/kafka-consumer-shell.sh \
2 localhost:9092 \
3 --topic MyTestTopic \
4 --from-beginning
5
6 some bytes sent
7 some bytes sent
8 some bytes sent
9 some bytes sent
10 some bytes sent
11 some bytes sent
12 some bytes sent
13 some bytes sent
14 some bytes sent
15 some bytes sent
16
```

◀ Top 10 Linux interview Q&As

30: Docker Tutorial: Apache Spark streaming in Python 3 with Apache  
Kafka 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](#)