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.

```
~/projects/docker-hadoop]$ docker run --hostname=qu
--privileged=true -t -i -v /Users/arulkumarankumar
--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.

```
[root@quickstart /]# python --version
Python 2.6.6
[root@quickstart /]#
```

Create SimpleSpark.py

```
[root@quickstart ~]# vi SimpleSpark.py
```

```
from pyspark import SparkConf, SparkContext

def main(sc):
    rdd = sc.parallelize(["John", "Peter", "Samue
    print(rdd.collect())

if __name__ == "__main__":
    conf = SparkConf().setAppName("Simple App")
    conf = conf.setMaster("local[*]")
    sc = SparkContext(conf=conf)
    main(sc)
```

spark-submit

Step 3: Submit the job to Spark.

```
[root@quickstart ~]# spark-submit /root/SimpleSparl

```

```
...
['John', 'Peter', 'Samuel']
.......
```

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

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.

```
[root@quickstart ~]# yum install python-pip
```

```
[root@quickstart ~]# pip --version
pip 7.1.0 from /usr/lib/python2.6/site-packages (py
[root@quickstart ~]#
```

Create a Python project structure

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

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

simple.py

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

```
[root@quickstart my-app]# vi mypackage/simple.py
```

```
from pyspark import SparkConf, SparkContext

class SimpleSpark:

    def myfunc(self, sc):
        rdd = sc.parallelize(["John", "Peter", "Sar
        print(rdd.collect())
```

driver.py

```
[root@quickstart my-app]# vi driver.py

```

```
from pyspark import SparkConf, SparkContext
from mypackage import simple

if __name__ == "__main__":
    conf = SparkConf().setAppName("Simple App")
    conf = conf.setMaster("local[*]")
    sc = SparkContext(conf=conf)
    simple.SimpleSpark().myfunc(sc)

```

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.

```
[root@quickstart my-app]#  vi setup.py

```

```
from setuptools import setup

setup(
    name = 'simple-spark',
    author = 'java-success',
    packages=['mypackage'],
    # Whatever arguments you need/want
)

```

```
[root@quickstart my-app]# tree
.
├── driver.py
├── mypackage
│   └── simple.py
└── setup.py

1 directory, 3 files
[root@quickstart my-app]#

```

Build an .egg file

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

```
[root@quickstart my-app]# python setup.py install

[root@quickstart my-app]# tree
.
├── build
│   ├── bdist.linux-x86_64
│   └── lib
│       └── mypackage
│           └── simple.py
├── dist
│   └── simple_spark-0.0.0-py2.6.egg
├── driver.py
├── mypackage
│   └── simple.py
├── setup.py
└── simple_spark.egg-info
    ├── dependency_links.txt
    ├── PKG-INFO
    ├── SOURCES.txt
    └── top_level.txt

7 directories, 9 files
[root@quickstart my-app]#
```

View an .egg file

Step 6: egg file is basically a zip file.

```
[root@quickstart my-app]# unzip -l dist/simple_spa
Archive:  dist/simple_spark-0.0.0-py2.6.egg
  Length      Date    Time    Name
---------  ---------- -----   ----
       10  06-09-2019 04:17   EGG-INFO/top_level.t
        1  06-09-2019 04:17   EGG-INFO/zip-safe
      172  06-09-2019 04:17   EGG-INFO/SOURCES.tx
        1  06-09-2019 04:17   EGG-INFO/dependency_
      191  06-09-2019 04:17   EGG-INFO/PKG-INFO
      224  06-09-2019 04:17   mypackage/__init__.p
       55  06-09-2019 04:17   mypackage/__init__.p
      679  06-09-2019 04:17   mypackage/simple.pyc
      181  06-09-2019 04:13   mypackage/simple.py
---------                     -------
```

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

spark-submit

Step 7: driver.py file gets invoked, which uses modules in the .egg file. "–verbose" is optional.

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

Output:

```
...
['John', 'Peter', 'Samuel']
.....
```

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.

```
[root@quickstart my-app]# pip list
You are using pip version 7.1.0, however version 1
You should consider upgrading via the 'pip instal'
argparse (1.4.0)
distribute (0.6.10)
iniparse (0.3.1)
iotop (0.3.2)
MySQL-python (1.2.3rc1)
ordereddict (1.2)
pip (7.1.0)
psycopg2 (2.0.14)
pycurl (7.19.0)
pygpgme (0.1)
python-snappy (0.5)
setuptools (0.6rc11)
simple-spark (0.0.0)
```

```
urlgrabber (3.9.1)
yum-metadata-parser (1.1.2)
[root@quickstart my-app]#
```

python global sitepackages

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

```
[root@quickstart my-app]# ls -ltr /usr/lib/python2
total 368
-rw-r--r--  1 root root  1085 Aug 17  2010 inipars
-rw-r--r--  1 root root  2362 Nov 12  2010 site.py
-rw-r--r--  1 root root 88196 Nov 12  2010 pkg_res
-rw-r--r--  1 root root   126 Nov 12  2010 easy_in
-rw-r--r--  2 root root  1771 Nov 12  2010 site.py
-rw-r--r--  2 root root  1771 Nov 12  2010 site.py
-rw-r--r--  1 root root    34 Nov 12  2010 setupto
-rw-r--r--  1 root root   144 Nov 12  2010 setupto
-rw-r--r--  2 root root 92875 Nov 12  2010 pkg_res
-rw-r--r--  2 root root 92875 Nov 12  2010 pkg_res
-rw-r--r--  2 root root   317 Nov 12  2010 easy_in
-rw-r--r--  2 root root   317 Nov 12  2010 easy_in
-rw-r--r--  1 root root  2285 Jul 30  2013 urlgrab
drwxr-xr-x  2 root root  4096 Mar  4  2015 urlgrab
drwxr-xr-x  2 root root  4096 Mar  4  2015 inipars
drwxr-xr-x  2 root root  4096 Mar  4  2015 yum
drwxr-xr-x  2 root root  4096 Mar  4  2015 rpmUtil
drwxr-xr-x  3 root root  4096 Apr  6  2016 setupto
drwxr-xr-x  2 root root  4096 Apr  6  2016 distrib
drwxr-xr-x  2 root root  4096 Apr  6  2016 iotop-0
drwxr-xr-x  2 root root  4096 Apr  6  2016 iotop
drwxr-xr-x  3 root root  4096 Apr  6  2016 python_
drwxr-xr-x  3 root root  4096 Apr  6  2016 argpars
drwxr-xr-x  2 root root  4096 Jun  9 15:46 pip-7.1
drwxr-xr-x 10 root root  4096 Jun  9 15:46 pip
drwxr-xr-x  4 root root  4096 Jun  9 15:55 simple_
-rw-r--r--  1 root root   286 Jun  9 15:55 easy-in
```

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