26: Docker Tutorial: Apache Kafka install, create topic & publish message on Cloudera quickstart

Posted on June 21, 2019

This extends 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 7180:71
```

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

```
4 cloudera/quickstart /usr/bin/docker-quickstart
```

Add new user & add to sudoers

Step 2: "-m" creates a home directory "/home/kafka", and the new user will be added to "**/etc/passwd**" & "**/etc/group**" files.

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

```
1 [root@quickstart /]# cat /etc/passwd
2 .....
3 kafka:x:502:504::/home/kafka:/bin/bash
4
```

```
1 [root@quickstart /]# cat /etc/group
2 .....
3 kafka:x:504:
4
```

Step 3: Add user to **sudo**.

```
1 [root@quickstart kafka]# visudo
```

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

```
1 [root@quickstart kafka]# usermod -aG wheel kafka
2
```

You can now check:

```
[kafka@quickstart ~]$ groups
kafka wheel
[kafka@quickstart ~]$ id
uid=502(kafka) gid=504(kafka) groups=504(kafka),10
[kafka@quickstart ~]$

```

This will allow sudo access to install “wget” in a later step.

Step 4: Set the password using passwd:

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

```

Switch user to kafka & home dir

Step 5: Switch user to kafka.

```
[root@quickstart /]# su kafka
[kafka@quickstart /]$ cd ~
[kafka@quickstart ~]$ pwd
/home/kafka

```

Install kafka binaries

Step 6: Install “wget”. And enter the password when prompted.

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

```

Step 7: Install kafka binaries.

```
[kafka@quickstart ~]$ wget https://www.apache.org/
.........
```

untar it:

```
[kafka@quickstart ~]$ tar -xvzf kafka_2.11-2.1.1.t
```

Create a symbolic link:

```
[kafka@quickstart ~]$ ln -s kafka_2.11-2.1.1 kafka
[kafka@quickstart ~]$ ls -ltr
total 60640
drwxr-xr-x 6 kafka kafka     4096 Feb  8 18:32 kafl
-rw-rw-r-- 1 kafka kafka 62090909 Feb 19 23:17 kafl
lrwxrwxrwx 1 kafka kafka       16 Jun 20 13:27 kafl
```

Install Java 8

As Kafka requires Java 8.

```
[kafka@quickstart ~]$ sudo yum install java-1.8.0-
```

```
[kafka@quickstart ~]$ java -version
openjdk version "1.8.0_212"
OpenJDK Runtime Environment (build 1.8.0_212-b04)
OpenJDK 64-Bit Server VM (build 25.212-b04, mixed r
```

Kafka needs zookeeper

for coordination and config management. Kafka installation comes with zookeeper

```
[kafka@quickstart ~]$ ls -ltr kafka/bin
total 132
```

```
-rwxr-xr-x 1 kafka kafka  968 Feb  8 18:30 zookeep
-rwxr-xr-x 1 kafka kafka 1001 Feb  8 18:30 zookeep
-rwxr-xr-x 1 kafka kafka 1393 Feb  8 18:30 zookeep
-rwxr-xr-x 1 kafka kafka  867 Feb  8 18:30 zookeep
drwxr-xr-x 2 kafka kafka 4096 Feb  8 18:30 windows
-rwxr-xr-x 1 kafka kafka 1722 Feb  8 18:30 trogdor
-rwxr-xr-x 1 kafka kafka  958 Feb  8 18:30 kafka-v
-rwxr-xr-x 1 kafka kafka  958 Feb  8 18:30 kafka-v
-rwxr-xr-x 1 kafka kafka  863 Feb  8 18:30 kafka-t
-rwxr-xr-x 1 kafka kafka  945 Feb  8 18:30 kafka-s
-rwxr-xr-x 1 kafka kafka  997 Feb  8 18:30 kafka-s
-rwxr-xr-x 1 kafka kafka 1376 Feb  8 18:30 kafka-s
-rwxr-xr-x 1 kafka kafka 9290 Feb  8 18:30 kafka-r
-rwxr-xr-x 1 kafka kafka  874 Feb  8 18:30 kafka-r
-rwxr-xr-x 1 kafka kafka  874 Feb  8 18:30 kafka-r
-rwxr-xr-x 1 kafka kafka  959 Feb  8 18:30 kafka-p
-rwxr-xr-x 1 kafka kafka  886 Feb  8 18:30 kafka-p
-rwxr-xr-x 1 kafka kafka  862 Feb  8 18:30 kafka-r
-rwxr-xr-x 1 kafka kafka  863 Feb  8 18:30 kafka-T
-rwxr-xr-x 1 kafka kafka  866 Feb  8 18:30 kafka-c
-rwxr-xr-x 1 kafka kafka  869 Feb  8 18:30 kafka-c
-rwxr-xr-x 1 kafka kafka  871 Feb  8 18:30 kafka-c
-rwxr-xr-x 1 kafka kafka  948 Feb  8 18:30 kafka-c
-rwxr-xr-x 1 kafka kafka  871 Feb  8 18:30 kafka-c
-rwxr-xr-x 1 kafka kafka  944 Feb  8 18:30 kafka-c
-rwxr-xr-x 1 kafka kafka  945 Feb  8 18:30 kafka-c
-rwxr-xr-x 1 kafka kafka  864 Feb  8 18:30 kafka-c
-rwxr-xr-x 1 kafka kafka  873 Feb  8 18:30 kafka-k
-rwxr-xr-x 1 kafka kafka  861 Feb  8 18:30 kafka-c
-rwxr-xr-x 1 kafka kafka 1418 Feb  8 18:30 connect
-rwxr-xr-x 1 kafka kafka 1421 Feb  8 18:30 connect
[kafka@quickstart ~]$
```

and

```
[kafka@quickstart ~]$ ls -ltr kafka/config/
total 68
-rw-r--r-- 1 kafka kafka 1169 Feb  8 18:30 trogdor
-rw-r--r-- 1 kafka kafka 1032 Feb  8 18:30 tools-T
-rw-r--r-- 1 kafka kafka 6851 Feb  8 18:30 server
-rw-r--r-- 1 kafka kafka 1925 Feb  8 18:30 produce
-rw-r--r-- 1 kafka kafka 4727 Feb  8 18:30 log4j.p
-rw-r--r-- 1 kafka kafka 1221 Feb  8 18:30 consume
-rw-r--r-- 1 kafka kafka 2262 Feb  8 18:30 connect
-rw-r--r-- 1 kafka kafka 1111 Feb  8 18:30 connect
-rw-r--r-- 1 kafka kafka  881 Feb  8 18:30 connect
-rw-r--r-- 1 kafka kafka  883 Feb  8 18:30 connect
-rw-r--r-- 1 kafka kafka 5321 Feb  8 18:30 connect
-rw-r--r-- 1 kafka kafka  909 Feb  8 18:30 connect
-rw-r--r-- 1 kafka kafka  906 Feb  8 18:30 connect
-rw-r--r-- 1 kafka kafka 1023 Jun 20 13:31 zookeep
```

```
[kafka@quickstart ~]$
```

But, cloudera/quickstart container already has zookeeper installed as HBase requires Zookeeper, and you can check the client port 2181 is already established.

```
 [kafka@quickstart ~]$ netstat -anp | grep 2181
tcp        0      0 0.0.0.0:2181                0.0
tcp        0      0 172.17.0.2:39114            172
tcp        0      0 172.17.0.2:39092            172
tcp        0      0 172.17.0.2:2181             172
tcp        0      0 172.17.0.2:2181             172
[kafka@quickstart ~]$
```

Start the kafka server

```
[kafka@quickstart ~]$ mkdir -p  kafka/logs
$ ./kafka/bin/kafka-server-start.sh ./kafka/config/
```

Test connection

```
[kafka@quickstart ~]$ jobs
[1]+  Running                 ./kafka/bin/kafka-ser
[kafka@quickstart ~]$
```

```
[kafka@quickstart ~]$ netstat -anlp | grep 9092
(Not all processes could be identified, non-owned p
 will not be shown, you would have to be root to se
tcp        0      0 0.0.0.0:9092                0.0
tcp        0      0 172.17.0.2:9092             172
tcp        0      0 172.17.0.2:39888            172
[kafka@quickstart ~]$
```

Create a topic & publish to it

```
[kafka@quickstart ~]$ kafka/bin/kafka-topics.sh --
--zookeeper localhost:2181 \
--replication-factor 1 \
--partitions 1 \
--topic MyTestTopic
Created topic "MyTestTopic".
[kafka@quickstart ~]$
```

List the topics

```
[kafka@quickstart ~]$ ./kafka/bin/kafka-topics.sh
--zookeeper localhost:2181   \
--list localhost:9092 \
MyTestTopic
```

publish to a topic

Publish the string “Hello, my topic” to the **MyTestTopic** topic by typing:

```
[kafka@quickstart ~]$ echo "Hello, my topic" | kafk
       --broker-list localhost:9092 \
       --topic MyTestTopic > /dev/null
[kafka@quickstart ~]$
```

consume from a topic

The following command consumes messages from MyTestTopic.

```
[kafka@quickstart ~]$ ./kafka/bin/kafka-console-co
--bootstrap-server localhost:9092 \
--topic MyTestTopic \
--from-beginning
```

```

Hello, my topic
```

Setting up a multi-broker cluster

```
1
2 [kafka@quickstart ~]$ cp kafka/config/server.proper
3 [kafka@quickstart ~]$ cp kafka/config/server.proper
4
```

Edit the files:

kafka/config/server-1.properties

```
1
2 broker.id=1
3 listeners=PLAINTEXT://quickstart.cloudera:9093
4 log.dirs=/tmp/kafka-logs-1
5
```

kafka/config/server-2.properties

```
1
2 broker.id=2
3 listeners=PLAINTEXT://quickstart.cloudera:9094
4 log.dirs=/tmp/kafka-logs-2
5
```

test for the ports: 9092, 9093 and 9094.

```
1  [kafka@quickstart ~]$ netstat -anp | grep 909
2  (Not all processes could be identified, non-owned
3   will not be shown, you would have to be root to
4  tcp        0      0 0.0.0.0:9090                0
5  tcp        0      0 0.0.0.0:9092                0
6  tcp        0      0 172.17.0.2:9093             0
7  tcp        0      0 172.17.0.2:9094             0
8  tcp        0      0 0.0.0.0:9095                0
9  tcp        0      0 172.17.0.2:40568            1
10 tcp        0      0 172.17.0.2:50832            1
11 tcp        0      0 172.17.0.2:9094             1
12 tcp        0      0 172.17.0.2:42430            1
13 tcp        0      0 172.17.0.2:9093             1
14 tcp        0      0 172.17.0.2:9092             1
15 [kafka@quickstart ~]$
```

```
16
```

Create a new topic

```
[kafka@quickstart ~]$ kafka/bin/kafka-topics.sh --
--zookeeper localhost:2181 \
--replication-factor 3 \
--partitions 1 \
--topic MyClusterTestTopic
Created topic "MyClusterTestTopic".
[kafka@quickstart ~]$

```

Describe topic

```
[kafka@quickstart ~]$ ./kafka/bin/kafka-topics.sh
--zookeeper localhost:2181  \
--describe localhost:9092 \
--topic MyClusterTestTopic

Topic:MyClusterTestTopic        PartitionCount:1
        Topic: MyClusterTestTopic       Partition:
[kafka@quickstart ~]$

```

Publish a message to the topic

```
[kafka@quickstart ~]$
[kafka@quickstart ~]$ ./kafka/bin/kafka-console-pr
        --broker-list localhost:9092 \
        --topic MyClusterTestTopic
>Hello sent to cluster topic
>multiple lines
>^C[kafka@quickstart ~]$

```

Consume messages

```
[kafka@quickstart ~]$ ./kafka/bin/kafka-console-co
--bootstrap-server localhost:9092 \
--topic MyClusterTestTopic \
--from-beginning

```

```
Hello sent to cluster topic
multiple lines
```

Test fault-tolerance

```
[kafka@quickstart ~]$ ps aux | grep server-1.proper
kafka    16723  2.4  3.5 5312372 346024 pts/0  Sl
```

Let's kill 16723, which is the **broker 1**, the **leader**:

```
[kafka@quickstart ~]$ kill -9  16723
```

Describe the topic:

```
[kafka@quickstart ~]$ ./kafka/bin/kafka-topics.sh
--zookeeper localhost:2181  \
--describe localhost:9092 \
--topic MyClusterTestTopic
```

Now the leader is broker 2.

```
Topic:MyClusterTestTopic        PartitionCount:1
        Topic: MyClusterTestTopic       Partition:
```

‹ 25: Docker Tutorial: HBase (i.e. NoSQL) Java API on Cloudera quickstart

27: Docker Tutorial: Apache Kafka with Java API on Cloudera quickstart ›

Disclaimer