

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](#) > 26:

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

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.

```
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

```
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
```

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



You can now check:

```
1 [kafka@quickstart ~]$ groups
2 kafka wheel
3 [kafka@quickstart ~]$ id
4 uid=502(kafka) gid=504(kafka) groups=504(kafka),100(kafka)
5 [kafka@quickstart ~]$
6
```

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

Step 4: Set the password using passwd:

```
1 [root@quickstart /]# sudo passwd kafka
2
```

Switch user to kafka & home dir

Step 5: Switch user to kafka.

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

Install kafka binaries

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

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

Step 7: Install kafka binaries.

```
1 [kafka@quickstart ~]$ wget https://www.apache.org/
2 .....
3
```

untar it:

```
1 [kafka@quickstart ~]$ tar -xvzf kafka_2.11-2.1.1.tg
2
```

Create a symbolic link:

```
1 [kafka@quickstart ~]$ ln -s kafka_2.11-2.1.1 kafka
2 [kafka@quickstart ~]$ ls -ltr
3 total 60640
4 drwxr-xr-x 6 kafka kafka 4096 Feb  8 18:32 kafl
5 -rw-rw-r-- 1 kafka kafka 62090909 Feb 19 23:17 kafl
6 lrwxrwxrwx 1 kafka kafka 16 Jun 20 13:27 kafl
7
```

Install Java 8

As Kafka requires Java 8.

```
1 [kafka@quickstart ~]$ sudo yum install java-1.8.0-c
2
```

```
1 [kafka@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 m
5
```

Kafka needs zookeeper

for coordination and config management. Kafka installation comes with zookeeper

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

```

3  -rwxr-xr-x 1 kafka kafka 968 Feb 8 18:30 zookeeper
4  -rwxr-xr-x 1 kafka kafka 1001 Feb 8 18:30 zookeeper
5  -rwxr-xr-x 1 kafka kafka 1393 Feb 8 18:30 zookeeper
6  -rwxr-xr-x 1 kafka kafka 867 Feb 8 18:30 zookeeper
7  drwxr-xr-x 2 kafka kafka 4096 Feb 8 18:30 windows
8  -rwxr-xr-x 1 kafka kafka 1722 Feb 8 18:30 trogdor
9  -rwxr-xr-x 1 kafka kafka 958 Feb 8 18:30 kafka-v
10 -rwxr-xr-x 1 kafka kafka 958 Feb 8 18:30 kafka-v
11 -rwxr-xr-x 1 kafka kafka 863 Feb 8 18:30 kafka-t
12 -rwxr-xr-x 1 kafka kafka 945 Feb 8 18:30 kafka-s
13 -rwxr-xr-x 1 kafka kafka 997 Feb 8 18:30 kafka-s
14 -rwxr-xr-x 1 kafka kafka 1376 Feb 8 18:30 kafka-s
15 -rwxr-xr-x 1 kafka kafka 9290 Feb 8 18:30 kafka-i
16 -rwxr-xr-x 1 kafka kafka 874 Feb 8 18:30 kafka-i
17 -rwxr-xr-x 1 kafka kafka 874 Feb 8 18:30 kafka-i
18 -rwxr-xr-x 1 kafka kafka 959 Feb 8 18:30 kafka-p
19 -rwxr-xr-x 1 kafka kafka 886 Feb 8 18:30 kafka-p
20 -rwxr-xr-x 1 kafka kafka 862 Feb 8 18:30 kafka-r
21 -rwxr-xr-x 1 kafka kafka 863 Feb 8 18:30 kafka-T
22 -rwxr-xr-x 1 kafka kafka 866 Feb 8 18:30 kafka-c
23 -rwxr-xr-x 1 kafka kafka 869 Feb 8 18:30 kafka-c
24 -rwxr-xr-x 1 kafka kafka 871 Feb 8 18:30 kafka-c
25 -rwxr-xr-x 1 kafka kafka 948 Feb 8 18:30 kafka-c
26 -rwxr-xr-x 1 kafka kafka 871 Feb 8 18:30 kafka-c
27 -rwxr-xr-x 1 kafka kafka 944 Feb 8 18:30 kafka-c
28 -rwxr-xr-x 1 kafka kafka 945 Feb 8 18:30 kafka-c
29 -rwxr-xr-x 1 kafka kafka 864 Feb 8 18:30 kafka-c
30 -rwxr-xr-x 1 kafka kafka 873 Feb 8 18:30 kafka-b
31 -rwxr-xr-x 1 kafka kafka 861 Feb 8 18:30 kafka-c
32 -rwxr-xr-x 1 kafka kafka 1418 Feb 8 18:30 connect
33 -rwxr-xr-x 1 kafka kafka 1421 Feb 8 18:30 connect
34 [kafka@quickstart ~]$
35

```

and

```

1  [kafka@quickstart ~]$ ls -ltr kafka/config/
2  total 68
3  -rw-r--r-- 1 kafka kafka 1169 Feb 8 18:30 trogdor
4  -rw-r--r-- 1 kafka kafka 1032 Feb 8 18:30 tools-
5  -rw-r--r-- 1 kafka kafka 6851 Feb 8 18:30 server
6  -rw-r--r-- 1 kafka kafka 1925 Feb 8 18:30 produce
7  -rw-r--r-- 1 kafka kafka 4727 Feb 8 18:30 log4j.p
8  -rw-r--r-- 1 kafka kafka 1221 Feb 8 18:30 consume
9  -rw-r--r-- 1 kafka kafka 2262 Feb 8 18:30 connect
10 -rw-r--r-- 1 kafka kafka 1111 Feb 8 18:30 connect
11 -rw-r--r-- 1 kafka kafka 881 Feb 8 18:30 connect
12 -rw-r--r-- 1 kafka kafka 883 Feb 8 18:30 connect
13 -rw-r--r-- 1 kafka kafka 5321 Feb 8 18:30 connect
14 -rw-r--r-- 1 kafka kafka 909 Feb 8 18:30 connect
15 -rw-r--r-- 1 kafka kafka 906 Feb 8 18:30 connect
16 -rw-r--r-- 1 kafka kafka 1023 Jun 20 13:31 zookeeper

```

```
17 [kafka@quickstart ~]$
18
```

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

```
1 [kafka@quickstart ~]$ netstat -anp | grep 2181
2 tcp        0      0 0.0.0.0:2181        0.0.0.0:*           0.0.0.0:2181
3 tcp        0      0 172.17.0.2:39114    172.17.0.2:2181     172.17.0.2:2181
4 tcp        0      0 172.17.0.2:39092    172.17.0.2:2181     172.17.0.2:2181
5 tcp        0      0 172.17.0.2:2181     172.17.0.2:2181     172.17.0.2:2181
6 tcp        0      0 172.17.0.2:2181     172.17.0.2:2181     172.17.0.2:2181
7 [kafka@quickstart ~]$
8
```

Start the kafka server

```
1 [kafka@quickstart ~]$ mkdir -p kafka/logs
2 $ ./kafka/bin/kafka-server-start.sh ./kafka/config/
3
```

Test connection

```
1 [kafka@quickstart ~]$ jobs
2 [1]+  Running                  ./kafka/bin/kafka-se
3 [kafka@quickstart ~]$
4
```

```
1 [kafka@quickstart ~]$ netstat -anlp | grep 9092
2 (Not all processes could be identified, non-owned p
3 will not be shown, you would have to be root to s
4 tcp        0      0 0.0.0.0:9092        0.0.0.0:*           0.0.0.0:9092
5 tcp        0      0 172.17.0.2:9092     172.17.0.2:9092     172.17.0.2:9092
6 tcp        0      0 172.17.0.2:39888    172.17.0.2:9092     172.17.0.2:9092
7 [kafka@quickstart ~]$
8
```

Create a topic & publish to it

```
1 [kafka@quickstart ~]$ kafka/bin/kafka-topics.sh --c
2 --zookeeper localhost:2181 \
3 --replication-factor 1 \
4 --partitions 1 \
5 --topic MyTestTopic
6 Created topic "MyTestTopic".
7 [kafka@quickstart ~]$
8
```

List the topics

```
1 [kafka@quickstart ~]$ ./kafka/bin/kafka-topics.sh
2 --zookeeper localhost:2181 \
3 --list localhost:9092 \
4 MyTestTopic
5
```

publish to a topic

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

```
1 [kafka@quickstart ~]$ echo "Hello, my topic" | kafk
2 --broker-list localhost:9092 \
3 --topic MyTestTopic > /dev/null
4 [kafka@quickstart ~]$
5
```

consume from a topic

The following command consumes messages from **MyTestTopic**.

```
1 [kafka@quickstart ~]$ ./kafka/bin/kafka-console-co
2 --bootstrap-server localhost:9092 \
3 --topic MyTestTopic \
4 --from-beginning
5
```

```
1
2 Hello, my topic
3
```

Setting up a multi-broker cluster

```
1  
2 [kafka@quickstart ~]$ cp kafka/config/server.properties  
3 [kafka@quickstart ~]$ cp kafka/config/server.properties  
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 s  
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      17  
10 tcp      0      0 172.17.0.2:50832      17  
11 tcp      0      0 172.17.0.2:9094       17  
12 tcp      0      0 172.17.0.2:42430      17  
13 tcp      0      0 172.17.0.2:9093       17  
14 tcp      0      0 172.17.0.2:9092       17  
15 [kafka@quickstart ~]$
```


16

Create a new topic

```
1 [kafka@quickstart ~]$ kafka/bin/kafka-topics.sh --c
2 --zookeeper localhost:2181 \
3 --replication-factor 3 \
4 --partitions 1 \
5 --topic MyClusterTestTopic
6 Created topic "MyClusterTestTopic".
7 [kafka@quickstart ~]$
8
```

Describe topic

```
1 [kafka@quickstart ~]$ ./kafka/bin/kafka-topics.sh
2 --zookeeper localhost:2181 \
3 --describe localhost:9092 \
4 --topic MyClusterTestTopic
5
6 Topic:MyClusterTestTopic      PartitionCount:1
7      Topic: MyClusterTestTopic      Partition:
8 [kafka@quickstart ~]$
9
```

Publish a message to the topic

```
1 [kafka@quickstart ~]$
2 [kafka@quickstart ~]$ ./kafka/bin/kafka-console-pro
3      --broker-list localhost:9092 \
4      --topic MyClusterTestTopic
5 >Hello sent to cluster topic
6 >multiple lines
7 >^C[kafka@quickstart ~]$
8
```

Consume messages

```
1 [kafka@quickstart ~]$ ./kafka/bin/kafka-console-co
2 --bootstrap-server localhost:9092 \
3 --topic MyClusterTestTopic \
4 --from-beginning
5
```

```
1  
2 Hello sent to cluster topic  
3 multiple lines  
4
```

Test fault-tolerance

```
1 [kafka@quickstart ~]$ ps aux | grep server-1.properties  
2 kafka      16723  2.4  3.5 5312372 346024 pts/0    Sl  
3
```

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

```
1 [kafka@quickstart ~]$ kill -9 16723  
2
```

Describe the topic:

```
1 [kafka@quickstart ~]$ ./kafka/bin/kafka-topics.sh  
2 --zookeeper localhost:2181 \  
3 --describe localhost:9092 \  
4 --topic MyClusterTestTopic  
5
```

Now the leader is broker 2.

```
1  
2 Topic:MyClusterTestTopic      PartitionCount:1  
3      Topic: MyClusterTestTopic      Partition:  
4
```

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

27: Docker Tutorial: Apache Kafka with Java API 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)