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

Posted on June 19, 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/arulkumarankumara
--publish-all=true -p 8888:8888 -p 80:80 -p 7180:71
cloudera/quickstart /usr/bin/docker-quickstart
```

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

Install maven

Step 2: Download maven from "http://mirror.ventraip.net.au/apache/maven/maven-3/3.6.1/binaries/apache-maven-3.6.1-bin.tar.gz"

```
[root@quickstart /]# sudo curl -L -O http://mirror
```

```
[root@quickstart /]# cp apache-maven-3.6.1-bin.tar
[root@quickstart /]# cd /opt
[root@quickstart /]# sudo tar xvfz apache-maven-3.6
[root@quickstart /]# sudo rm -f apache-maven-3.6.1-
```

Step 3: Add it to your profile so that you can run the "mvn" command. Add it to "**.bashrc**" if you are a root user or to "**.bash_profile**" if you are a any other user.

```
[root@quickstart opt]# vi ~/.bash_profile
```

add the following lines at the end of "~/.bash_profile"

```
M3_HOME=/opt/apache-maven-3.6.1
export MAVEN_OPTS=-Dhttps.protocols=TLSv1,TLSv1.1,
export PATH=$PATH:$M3_HOME/bin
```

Apply the changes with the "**source**" command.

```
[root@quickstart opt]# source  ~/.bash_profile
```

```
[root@quickstart opt]# mvn -version
Apache Maven 3.6.1 (d66c9c0b3152b2e69ee9bac180bb8f
Maven home: /opt/apache-maven-3.6.1
Java version: 1.7.0_67, vendor: Oracle Corporation
Default locale: en_US, platform encoding: UTF-8
OS name: "linux", version: "4.9.125-linuxkit", arc
```

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

```
[root@quickstart opt]#

```

Now Maven is installed.

Create a simple maven project structure

Step 4: Create "**projects**" folder and create a maven project structure.

```
[root@quickstart ~]# mkdir /projects
[root@quickstart /]# cd !$
cd projects
[root@quickstart projects]#

```

Create a maven project structure.

```
[root@quickstart projects]# mvn archetype:generate
   -DartifactId=simple-hbase \
   -DarchetypeArtifactId=maven-archetype-quickstar
   -DinteractiveMode=false

```

Give it a few minutes.

```
[root@quickstart projects]# tree
.
└── simple-hbase
    ├── pom.xml
    └── src
        ├── main
        │   └── java
        │       └── com
        │           └── mycompany
        │               └── app
        │                   └── App.java
        └── test
            └── java
                └── com
                    └── mycompany
                        └── app
                            └── AppTest.java
```

```
18
19 12 directories, 3 files
20 [root@quickstart projects]#
21
```

pom.xml add hbase-client

```
1 [root@quickstart projects]# vi simple-hbase/pom.xm
2
```

```
1  <project xmlns="http://maven.apache.org/POM/4.0.0
2    xsi:schemaLocation="http://maven.apache.org/POM
3    <modelVersion>4.0.0</modelVersion>
4    <groupId>com.mycompany.app</groupId>
5    <artifactId>simple-hbase</artifactId>
6    <packaging>jar</packaging>
7    <version>1.0-SNAPSHOT</version>
8    <name>simple-hbase</name>
9    <url>http://maven.apache.org</url>
10   <dependencies>
11     <dependency>
12       <groupId>junit</groupId>
13       <artifactId>junit</artifactId>
14       <version>3.8.1</version>
15       <scope>test</scope>
16     </dependency>
17
18     <dependency>
19       <groupId>org.apache.hbase</groupId>
20       <artifactId>hbase-client</artifactId>
21       <version>1.1.3</version>
22     </dependency>
23   </dependencies>
24
25   <!-- Build uber jar -->
26   <build>
27       <plugins>
28         <plugin>
29            <groupId>org.apache.maven.plugins</grou
30            <artifactId>maven-shade-plugin</artifa
31            <executions>
32               <execution>
33                 <phase>package</phase>
34                 <goals>
35                     <goal>shade</goal>
36                 </goals>
37             </execution>
38           </executions>
39         </plugin>
40       </plugins>
41   </build>
42
```

```
</project>

```

Create a HBase table via Java API

```
[root@quickstart projects]# vi simple-hbase/src/ma

```

```
package com.mycompany.app;

import java.io.IOException;
import org.apache.hadoop.hbase.HTableDescriptor;
import org.apache.hadoop.hbase.client.HBaseAdmin;
import org.apache.hadoop.hbase.HColumnDescriptor;
import org.apache.hadoop.hbase.TableName;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.hbase.HBaseConfiguration
import org.apache.hadoop.hbase.client.Table;
import org.apache.hadoop.hbase.client.Put;
import org.apache.hadoop.hbase.util.Bytes;
import org.apache.hadoop.hbase.client.Connection;
import org.apache.hadoop.hbase.client.ConnectionF

public class CreateAnHbaseTable
{
  public static void main(String[] args) throws IO
  {

    String TABLE_NAME = "employees";

    Configuration config = HBaseConfiguration.cre
    config.setInt("timeout", 120000);

    //zookeeper quorum, basic info needed to proc
    config.set("hbase.zookeeper.quorum","quickstar
    config.set("hbase.zookeeper.property.clientPor

    HTableDescriptor htable = new HTableDescriptor
    htable.addFamily( new HColumnDescriptor("pers
    htable.addFamily( new HColumnDescriptor("prof
    System.out.println( "Connecting..." );
    HBaseAdmin hbaseAdmin = new HBaseAdmin( confi
    System.out.println( "Creating Table..." );
    hbaseAdmin.createTable( htable );
    System.out.println("Done!");


    Connection connection = ConnectionFactory.cre
    Table table  = connection.getTable(TableName.

```

```
43      try {
44          Put put = new Put(Bytes.toBytes("999"));
45          put.add(Bytes.toBytes("personal"),Bytes.t
46          table.put(put);
47          System.out.println("done");
48
49      } catch (Exception e) {
50          e.printStackTrace();
51      } finally {
52          table.close();
53          connection.close();
54      }
55
56    }
57
58  }
59
60
```

Note: Some API could be deprecated, and replaced with newer APIs.

Build package with maven

```
1  [root@quickstart projects]# cd simple-hbase/
2  [root@quickstart simple-hbase]# mvn package
3  .....
4
```

```
1  [root@quickstart projects]# tree
2  .
3  ├── dependency-reduced-pom.xml
4  ├── pom.xml
5  ├── src
6  │   ├── main
7  │   │   └── java
8  │   │       └── com
9  │   │           └── mycompany
10 │   │               └── app
11 │   │                   ├── App.java
12 │   │                   └── CreateAnHbaseTable.jav
13 │   └── test
14 │       └── java
15 │           └── com
16 │               └── mycompany
17 │                   └── app
18 │                       └── AppTest.java
19 └── target
```

```
20      ├── classes
21      │   └── com
22      │       └── mycompany
23      │           └── app
24      │               ├── App.class
25      │               └── CreateAnHbaseTable.class
26      ├── maven-archiver
27      │   └── pom.properties
28      ├── maven-status
29      │   └── maven-compiler-plugin
30      │       ├── compile
31      │       │   └── default-compile
32      │       │       ├── createdFiles.lst
33      │       │       └── inputFiles.lst
34      │       └── testCompile
35      │           └── default-testCompile
36      │               ├── createdFiles.lst
37      │               └── inputFiles.lst
38      ├── original-simple-hbase-1.0-SNAPSHOT.jar
39      ├── simple-hbase-1.0-SNAPSHOT.jar
40      ├── surefire-reports
41      │   ├── com.mycompany.app.AppTest.txt
42      │   └── TEST-com.mycompany.app.AppTest.xml
43      └── test-classes
44          └── com
45              └── mycompany
46                  └── app
47                      └── AppTest.class
48
```

Run

```
1 [root@quickstart simple-hbase]# java -cp target/sir
2 Connecting...
3 log4j:WARN No appenders could be found for logger (
4 log4j:WARN Please initialize the log4j system prope
5 log4j:WARN See http://logging.apache.org/log4j/1.2/
6 Creating Table...
7 Done!
8 done
9
```

Verify via hbase shell

```
1 [root@quickstart simple-hbase]# hbase shell
2 2019-06-19 12:57:32,277 INFO  [main] Configuration
3 HBase Shell; enter 'help<RETURN>' for list of suppo
4 Type "exit<RETURN>" to leave the HBase Shell
5 Version 1.2.0-cdh5.7.0, rUnknown, Wed Mar 23 11:39
```

```
6
7 hbase(main):001:0>
8
```

list

```
1 hbase(main):002:0> list
2 TABLE
3 employees
4 1 row(s) in 0.1800 seconds
5
6 => ["employees"]
7 hbase(main):003:0>
8
```

scan

```
1 hbase(main):003:0> scan 'employees'
2 ROW                                COLUMN+CELL
3  999                               column=personal:
4 1 row(s) in 0.1230 seconds
5
6 hbase(main):004:0>
7
```

‹ 24: Docker Tutorial: HBase (i.e. NoSQL DB) Shell on Cloudera quickstart

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

Disclaimer