02: Simple Apache Storm application running on a single node local cluster

Posted on January 18, 2018

This extends Installing & getting started with Apache Storm on Cloudera quickstart and Simple Apache Storm application running inside Eclipse in a local cluster to run the same example in Storm cluster mode.

Step 1: Firstly modify the "storm.yaml" in the folder "/opt/storm/apache-storm-1.1.1/conf" after backing up file. Storm works with zookeeper, and we can specify Cloudera's zookeeper host name by looking it

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

up via the Cloudera manager configuration. Also create a folder "/app/storm" folder for Nimbus, Supervisor, and UI processes createfiles under thhis folder whilst running.

```

# limitations under the License.

########### These MUST be filled in for a storm co
storm.zookeeper.servers:
     - "quickstart.cloudera"

storm.zookeeper.port: 2181

storm.local.dir: "/app/storm/"

nimbus.seeds: ["localhost"]

supervisor.slots.ports:
    - 6700
    - 6701
    - 6702
    - 6703


#
# ##### These may optionally be filled in:
#
## List of custom serializations
# topology.kryo.register:
#     - org.mycompany.MyType
#     - org.mycompany.MyType2: org.mycompany.MyTyp
#
## List of custom kryo decorators
# topology.kryo.decorators:
#     - org.mycompany.MyDecorator
#
## Locations of the drpc servers
# drpc.servers:
#     - "localhost"

```

The above "**storm.yaml**" configuration overrides the "**default.yaml**" file that is in the "storm-core-1.1.1.jar".

Step 2: The pom.xml file will build the jar. The "storm-core-1.1.1.jar" should have the "provided" scope as it

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

is present in the folder “/opt/storm/apache-storm-1.1.1/lib”.

```

<project xmlns="http://maven.apache.org/POM/4.0.0
    xsi:schemaLocation="http://maven.apache.org/P
    <modelVersion>4.0.0</modelVersion>
    <groupId>com.mytutorial</groupId>
    <artifactId>simple-storm</artifactId>
    <packaging>jar</packaging>
    <version>1.0-SNAPSHOT</version>
    <name>simple-storm</name>
    <url>http://maven.apache.org</url>
    <dependencies>
        <dependency>
            <groupId>junit</groupId>
            <artifactId>junit</artifactId>
            <version>3.8.1</version>
            <scope>test</scope>
        </dependency>

        <dependency>
            <groupId>org.apache.storm</groupId>
            <artifactId>storm-core</artifactId>
            <version>1.1.1</version>
            <scope>provided</scope>
        </dependency>
    </dependencies>

    <build>
        <plugins>
            <plugin>
                <artifactId>maven-assembly-plugin
                <configuration>
                    <archive>
                        <manifest>
                            <mainClass>com.mytuto
                        </manifest>
                    </archive>
                    <descriptorRefs>
                        <descriptorRef>jar-with-d
                    </descriptorRefs>
                </configuration>
            </plugin>
        </plugins>
    </build>

</project>

```

Step 3: The main method topology code will be slighly different to running in the local cluster.

```

package com.mytutorial;

import org.apache.storm.Config;
import org.apache.storm.StormSubmitter;
import org.apache.storm.generated.AlreadyAliveExc
import org.apache.storm.generated.AuthorizationEx
import org.apache.storm.generated.InvalidTopology
import org.apache.storm.topology.TopologyBuilder;

public class SimpleTopology {

    public static void main(String[] args) throws
        TopologyBuilder builder = new TopologyBui
        builder.setSpout("simple-spout", new Simp
        builder.setBolt("isOdd", new SimpleIsOddB

        Config conf = new Config();

        StormSubmitter.submitTopology(args[0], co
    }

}

```

Step 4: Build the jar file.

```

bash-4.1$ cd /home/cloudera/projects/simple-storm
bash-4.1$ mvn clean package assembly:single

```

This will build “simple-storm-1.0-SNAPSHOT-jar-with-dependencies.jar”,

Step 5: Open three terminal windows and start Nimbus, Supervisor, and UI as shown below.

```

bash-4.1$ cd /opt/storm/apache-storm-1.1.1/bin
bash-4.1$ sudo ./storm nimbus
```

```

bash-4.1$ cd /opt/storm/apache-storm-1.1.1/bin
bash-4.1$ sudo ./storm supervisor

```

```

bash-4.1$ cd /opt/storm/apache-storm-1.1.1/bin
bash-4.1$ sudo ./storm ui

```

After the UI has started, open a browser and enter the URL “http://quickstart.cloudera:8080”.

Step 6: Submit the Topology to the above 1 node cluster with the following command by opening a new terminal window.

```

bash-4.1$ cd /opt/storm/apache-storm-1.1.1/bin
bash-4.1$ sudo ./storm jar /home/cloudera/projects/

```

Step 7: You can see the topology that you had submitted via the UI by refreshing it and also vi the coman-line

```

bash-4.1$ cd /opt/storm/apache-storm-1.1.1/bin
bash-4.1$ sudo ./storm list

```

```

Topology_name        Status     Num_tasks  Num_wor
-------------------------------------------------------
simple-storm-tutorial ACTIVE     3          1

```

Step 8: You can remove the topology with the following command.

```

bash-4.1$ cd /opt/storm/apache-storm-1.1.1/bin
bash-4.1$ sudo ./storm kill simple-storm-tutorial

```

If you list it again, the topology would not be running on the cluster.

‹ 02: Simple Apache Storm application running inside Eclipse in a local cluster

18: Q121 – Q124 Choosing between HDFS & AWS S3 for BigData projects interview Q&As ›

Disclaimer