Developer Exercise

Setup of an Eclipse project to develop MapReduce Jobs based on CDH4 / CDH3u5 with Maven and MRUnit.

References:

Maven: Building a Self-Contained Hadoop Job

http://blog.mafr.de/2010/07/24/maven-hadoop-job/

A Maven Archetype for Hadoop Jobs

http://blogs.apache.org/mrunit/

MRUnit Intro

http://de.slideshare.net/emwendelin/testing-hadoop-jobs-with-mrunit

Required Steps:

1) Install Maven2

2) Check Installation

```
mvn -version
```

3) Create a new project

```
mvn archetype:generate -DarchetypeGroupId=org.apache.maven.archetypes
-DarchetypeArtifactId=maven-archetype-quickstart
-DgroupId=HadoopTRAINING -DartifactId=P1
```

4) Create the project Metadata for Eclipse, so you can import the new project directly to your Eclipse workspace.

```
mvn -Declipse.workspace=%eclipse-workspace-path%  eclipse:configure-
workspace eclipse:eclipse
```

Versions and Repositories:

CDH3

DOCU: https://ccp.cloudera.com/display/CDH3U5STAGE/Using+the+CDH3+Maven+Repository

REPO: https://repository.cloudera.com/artifactory/cloudera-repos/

CDH4

DOCU: https://ccp.cloudera.com/display/CDH4DOC/Using+the+CDH4+Maven+Repository

REPO: https://repository.cloudera.com/artifactory/cloudera-repos/.

```
<dependency>
  <groupId>org.apache.hadoop</groupId>
  <artifactId>hadoop-mrunit</artifactId>
  <version>0.20.2-cdh3u5</version>
  <scope>test</scope>
 </dependency>
```

… but the MRUnit package is not in that repositories.

It is here:

<url>https://repository.cloudera.com/content/repositories/releases</url>

so I we have to add a second repository to the pom.xml.

Alternative approach with the latest version of Apache MRUnit

```
<dependency>
  <groupId>org.apache.mrunit</groupId>
  <artifactId>mrunit</artifactId>
  <version>0.9.0</version>
  <scope>test</scope>
</dependency>
```

Install the new MRUnit v0.9.0 package manually.

1) org.apache.mrunit:mrunit:jar:0.9.0

Try downloading the file manually from the project website.

Then, install it using the command:

```
mvn install:install-file -DgroupId=org.apache.mrunit -DartifactId=mrunit -Dversion=0.9.0 \
-Dpackaging=jar -Dfile=/path/to/file
```

Alternatively, if you host your own repository you can deploy the file there:

```
mvn deploy:deploy-file -DgroupId=org.apache.mrunit -DartifactId=mrunit -Dversion=0.9.0 \
-Dpackaging=jar -Dfile=/path/to/file -Durl=[url] -DrepositoryId=[id]
```

The POM file

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xsi:schemaLocation="http://maven.apache.org/POM/4.0.0"
       http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>HadoopTRAINING</groupId>

  <artifactId>P1</artifactId>
  <version>1.0-SNAPSHOT</version>
  <packaging>jar</packaging>
  <!-- The name should likely match the artifact ID -->
  <name>P1</name>
  <url>http://maven.apache.org</url>

  <properties>
    <!-- SELECT THE API version to use -->
    <!--hadoop.version>2.0.0-mr1-cdh4.1.2</hadoop.version-->
    <hadoop.version>0.20.2-cdh3u5</hadoop.version>
  </properties>

  <build>
    <pluginManagement>
      <plugins>
        <plugin>
          <groupId>org.apache.maven.plugins</groupId>
          <artifactId>maven-compiler-plugin</artifactId>
          <version>2.3.2</version>
          <configuration>
            <source>1.6</source>
            <target>1.6</target>
          </configuration>
        </plugin>
      </plugins>
    </pluginManagement>

    <plugins>
      <plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-shade-plugin</artifactId>
        <version>1.7.1</version>
        <executions>
          <execution>
```

```
            <phase>package</phase>
            <goals>
              <goal>shade</goal>
            </goals>
          </execution>
        </executions>
      </plugin>

      <plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-eclipse-plugin</artifactId>
        <version>2.9</version>
        <configuration>
          <buildOutputDirectory>eclipse-classes</buildOutputDirectory>
          <downloadSources>true</downloadSources>
          <downloadJavadocs>false</downloadJavadocs>
        </configuration>
      </plugin>
    </plugins>
  </build>

  <dependencies>

    <!-- what MRUnit should be used ? -->

<!--
    <!dependency>
      <groupId>org.apache.mrunit</groupId>
      <artifactId>mrunit</artifactId>
      <version>0.9.0</version>
      <scope>test</scope>
    </dependency>
-->

   <dependency>
      <groupId>org.apache.hadoop</groupId>
      <artifactId>hadoop-mrunit</artifactId>
      <version>0.20.2-cdh3u5</version>
      <scope>test</scope>
    </dependency>

    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
```

```
      <version>4.8.2</version>
      <scope>test</scope>
    </dependency>

    <dependency>
      <groupId>org.apache.hadoop</groupId>
      <artifactId>hadoop-core</artifactId>
      <version>${hadoop.version}</version>
      <scope>provided</scope>
    </dependency>

    <dependency>
      <groupId>org.apache.hadoop</groupId>
      <artifactId>hadoop-client</artifactId>
      <version>${hadoop.version}</version>
      <scope>provided</scope>
    </dependency>
  </dependencies>

  <!-- WE USE TWO repositories as the MRUnit libs are required in the project -->
  <repositories>
    <repository>
      <id>cdh</id>
      <url>https://repository.cloudera.com/artifactory/cloudera-repos</url>
      <releases>
        <enabled>true</enabled>
      </releases>
      <snapshots>
        <enabled>false</enabled>
      </snapshots>
    </repository>
    <repository>
      <id>cloudera mrunit</id>
      <url>https://repository.cloudera.com/content/repositories/releases</url>
      <releases>
        <enabled>true</enabled>
      </releases>
      <snapshots>
        <enabled>false</enabled>
      </snapshots>
    </repository>
  </repositories>
</project>
```