

# Analytical stack installation procedure

For the time being, this is only a prototypical stack, composed of:

1. JDK 8
2. Hadoop 2.6
3. Spark 1.3.0-SNAPSHOT
4. Zookeeper 3.4.6
5. Accumulo 1.5.1
6. GeoMesa 1.0.0-SNAPSHOT

## Installation procedure

1. Install Hadoop 2.6.0 following [these instructions](#)
2. Install Spark 1.3 and make it work with Hadoop 2.6 (see [Spark build instructions](#) for context)
  - a. `git clone git@github.com:apache/spark.git` `sbin/spark-config.sh`
  - b. Execute:

```
./make-distribution.sh -Phadoop-2.4 -Dhadoop.version=2.6.0 -Phive -Phive-thriftserver
```
  - c. Add the export of JAVA\_HOME in `sbin/spark-config.sh` (like: `export JAVA_HOME="/usr/lib/jvm/java-8-oracle"`)
  - d. Set the environment variable HADOOP\_HOME and HADOOP\_PREFIX pointing to the directory containing HADOOP
  - e. Start the HDFS service;
  - f. Start Spark;
  - g. Call the Spark shell:

```
$SPARK_HOME/bin/spark-shell --master spark://<servername>:7077
```

- h. Test it by reading and parsing a text file previously stored in HDFS:

```
val textFile= sc.textFile("hdfs://localhost:9000/ulysses.txt");  
textFile.count()
```

3. To install SPARK libraries to the local Maven repo (the SNAPSHOT is not on a public one):

```
export MAVEN_OPTS="-Xmx2g -XX:MaxPermSize=512M -XX:ReservedCodeCacheSize=512m"  
mvn -Phadoop-2.4 -Dhadoop.version=2.6.0 -DskipTests install
```

4. Install Zookeeper

- a. Download a [stable Zookeeper](#)
- b. Unpack the GZipped-tar
- c. Set the ZOOKEEPER\_HOME variable pointing to the unpacked directory
- d. Create a file `zoo.cfg` under `${ZOOKEEPER_HOME}/conf` with the following content:

```
tickTime=2000  
dataDir=/var/zookeeper  
clientPort=2181
```
- e. Create the `/var/zookeeper` directory

5. Install Accumulo

- a. Download the [binary of version of Accumulo 1.5.1](#)
- b. Unpack the GZipped TAR file
- c. Set the `{{ACCUMULO_HOME}}` environment variable
- d. Copy the following to `$ACCUMULO_HOME/conf`

```
$ACCUMULO_HOME/conf/512MB/standalone/masters  
$ACCUMULO_HOME/conf/512MB/standalone/monitor  
$ACCUMULO_HOME/conf/512MB/standalone/slaves  
$ACCUMULO_HOME/conf/512MB/standalone/tracers  
$ACCUMULO_HOME/conf/512MB/standalone/accumulo-env.sh  
$ACCUMULO_HOME/conf/512MB/standalone/accumulo-site.xml  
$ACCUMULO_HOME/conf/512MB/standalone/log4j.properties
```
- e. Add the following to `${ACCUMULO_HOME}/conf/accumulo-site.xml` in the value of the `{{general.classpaths}}` property:

```
$ACCUMULO_HOME/server/target/classes/  
$ACCUMULO_HOME/lib/accumulo-server.jar,  
$ACCUMULO_HOME/core/target/classes/  
$ACCUMULO_HOME/lib/accumulo-core.jar,  
$ACCUMULO_HOME/start/target/classes/  
$ACCUMULO_HOME/lib/accumulo-start.jar,
```

- ```

$ACCUMULO_HOME/fate/target/classes/,
$ACCUMULO_HOME/lib/accumulo-fate.jar,
$ACCUMULO_HOME/proxy/target/classes/,
$ACCUMULO_HOME/lib/accumulo-proxy.jar,
$ACCUMULO_HOME/lib/[^.]*.jar,
$ZOOKEEPER_HOME/zookeeper/[^.]*.jar,
$HADOOP_PREFIX/share/hadoop/common/*.jar,
$HADOOP_PREFIX/share/hadoop/common/lib/*.jar,
$HADOOP_PREFIX/share/hadoop/hdfs/*.jar,
$HADOOP_PREFIX/share/hadoop/hdfs/lib/*.jar,
$HADOOP_PREFIX/share/hadoop/mapreduce/lib/*.jar,
$HADOOP_PREFIX/share/hadoop/yarn/lib/*.jar,
$HADOOP_PREFIX/etc/hadoop
f. Configure accumulo-env.sh to set JAVA_HOME, HADOOP_HOME, and ZOOKEEPER_HOME variables.
g. Start HDFS (if not started already) by issuing {{SHADOOP_HOME/sbin/start-dfs.sh}}
h. Start Accumulo by issuing {{ACCUMULO_HOME/bin/start-all.sh}}
i. Initialize the instance by executing {{ACCUMULO_HOME/bin/accumulo init}} and set its name to "tweeter" and password to
   "tweeter"
6. Create the Twitter tables in Accumulo:
a. Start the shell $ACCUMULO_HOME/bin/accumulo shell -u root
b. Create a twitter user: createuser tweeter
c. Create a table to hold the Tweets: createtable tweet
d. Set permissions:

grant Table.WRITE -t tweet -u tweeter
grant Table.READ -t tweet -u tweeter

7. Install GeoMesa
a. Clone master branch of GeoMesa git clone https://github.com/locationtech/geomesa.git
b. Move in the GeoMesa directory Build it mvn clean install
c. After the building is completed:
d. Set GEOMESA_HOME
e. Copy the geomesa-distributed-runtime-accumulo1.5-1.0.0-rc.5-SNAPSHOT.jar to $ACCUMULO_HOME lib/ext
f. Re-start Accumulo
8. Run the quick-start GeoMesa program
a. Clone q-s master branch git clone https://github.com/geomesa/geomesa-quickstart.git
b. Edit the pom.xml file, updating the dependency on GeoMesa to the one matching the GeoMeda version (for instance 1.0.0-rc
   .5-SNAPSHOT)
c. Add a dependency on Objenesis:

<dependency>

    <groupId>org.objenesis</groupId>

    <artifactId>objenesis</artifactId>

    <version>2.1</version>

</dependency>
d. Move in the q-s directory and build it mvn clean install
e. Run the q-s program: java -cp ./target/geomesa-quickstart-1.0-SNAPSHOT.jar org.geomesa.QuickStart
   -instanceId tweeter -zookeepers "localhost:2181" -user root -password tweeter -tableName tweet

```

If everything goes according to plan, Accumulo should show:

Creating feature-type (schema): QuickStart

Creating new features

Inserting new features

Submitting query

```

1. Bierce|394|Sat Aug 02 09:55:05 AEST 2014|POINT (-77.42555615743139 -37.26710898726304)|null
2. Bierce|640|Mon Sep 15 05:48:25 AEST 2014|POINT (-77.36222958792739 -37.13013846773835)|null
3. Bierce|931|Sat Jul 05 08:25:38 AEST 2014|POINT (-76.51304097832912 -37.49406125975311)|null
4. Bierce|589|Sat Jul 05 16:02:15 AEST 2014|POINT (-76.88146600670152 -37.40156607152168)|null
5. Bierce|322|Wed Jul 16 07:09:42 AEST 2014|POINT (-77.01760098223343 -37.30933767159561)|null
6. Bierce|886|Wed Jul 23 04:12:36 AEST 2014|POINT (-76.59795732474399 -37.18420917493149)|null
7. Bierce|343|Wed Aug 06 18:59:22 AEST 2014|POINT (-76.66826220670282 -37.44503877750368)|null
8. Bierce|925|Mon Aug 18 13:28:33 AEST 2014|POINT (-76.5621106573523 -37.34321201566148)|null
9. Bierce|259|Fri Aug 29 05:59:30 AEST 2014|POINT (-76.90122194030118 -37.148525741002466)|null

```

## Commands to start/stop the stack

- To start the whole stack:  
`${HADOOP_HOME}/sbin/start-dfs.sh`  
`${ZOOKEEPER_HOME}/bin/zkServer.sh start`  
`${SPARK_HOME}/sbin/start-all.sh`  
`${ACCUMULO_HOME}/bin/start-all.sh`
- To stop it:  
`${ACCUMULO_HOME}/bin/stop-all.sh`  
`${SPARK_HOME}/sbin/stop-all.sh`  
`${ZOOKEEPER_HOME}/bin/zkServer.sh stop`  
`${HADOOP_HOME}/sbin/stop-dfs.sh`
- Alternatively, a set of alias to star/stopt the stack and use shorthands for the shells:

```
alias hdfs="hadoop fs"
alias starthdfs="${HADOOP_HOME}/sbin/start-dfs.sh"
alias stophdfs="${HADOOP_HOME}/sbin/stop-dfs.sh"
alias startzoo="${ZOOKEEPER_HOME}/bin/zkServer.sh start"
alias stopzoo="${ZOOKEEPER_HOME}/bin/zkServer.sh stop"
alias startspark="${SPARK_HOME}/sbin/start-all.sh"
alias stopspark="${SPARK_HOME}/sbin/stop-all.sh"
alias spark="${SPARK_HOME}/bin/spark-shell --master spark://vaneyck:7077"
alias startaccumulo="${ACCUMULO_HOME}/bin/start-all.sh"
alias stopaccumulo="${ACCUMULO_HOME}/bin/stop-all.sh"
alias accumulo="${ACCUMULO_HOME}/bin/accumulo shell -u root"
alias startstack="starthdfs;startzoo;startspark;startaccumulo"
alias stopstack="stopaccumulo;stopspark;stopzoo;stophdfs"

alias geomesa="${GEOMESA_HOME}/geomesa-tools/bin/geomesa"
```