Cluster installation

System setup

Install Java 8 on all nodes:

```
sudo add-apt-repository ppa:webupd8team/java
sudo apt-get update
sudo apt-get install oracle-java8-installer
sudo apt-get install oracle-java8-set-default2- Install hadoop:
```

One node should be selected as Namenode and the others will serve as data nodes.

Add system group and user for hadoop in all nodes:

```
sudo addgroup hadoop
sudo adduser --ingroup hadoop hduser
sudo adduser hduser sudo
```

Setup passwordless SSH for hduser in all nodes.

```
sudo su - hduser
ssh-keygen -t rsa -P "" -f ~/.ssh/id_rsa
cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
ssh localhost (to test if passwordless ssh to local host has been setup properly)
```

Make sure hduser on namenode can SSH to other data nodes:

```
ssh-copy-id -i /home/hduser/.ssh/id_rsa.pub hduser@115.146.87.105
ssh-copy-id -i /home/hduser/.ssh/id_rsa.pub hduser@115.146.87.103
```

Hadoop installation

Install hadoop required dependencies, then download hadoop source and build it in the Namenode.

```
wget https://protobuf.googlecode.com/files/protobuf-2.5.0.tar.gz
tar -xvf protobuf-2.5.0.tar.gz
cd protobuf-2.5.0/
sudo ./configure
sudo make
sudo make check
sudo make install
sudo ldconfig
sudo apt-get install maven build-essential zlib1g-dev cmake pkg-config libssl-dev
wget http://apache.mirror.anlx.net/hadoop/core/hadoop-2.6.0/hadoop-2.6.0-src.tar.gz
tar -xvf hadoop-2.6.0-src.tar.gz
cd hadoop-2.6.0-src/
mvn clean package -Pdist,native -Dmaven.javadoc.skip=true -DskipTests -Dtar
```

Copy compiled hadoop to other Datanodes.

```
cp /home/hduser/hadoop-2.6.0-src/hadoop-dist/target/hadoop-2.6.0.tar.gz /home/hduser/
scp /home/hduser/hadoop-2.6.0-src/hadoop-dist/target/hadoop-2.6.0.tar.gz hduser@115.146.87.105:/home/hduser
/
scp /home/hduser/hadoop-2.6.0-src/hadoop-dist/target/hadoop-2.6.0.tar.gz hduser@115.146.87.103:/home/hduser
/
```

Unpack compiled archive and change its ownership on all nodes.

```
sudo tar -xvf /home/hduser/hadoop-2.6.0.tar.gz -C /usr/local/
sudo ln -s /usr/local/hadoop-2.6.0 /usr/local/hadoop
sudo chown -R hduser:hadoop /usr/local/hadoop-2.6.0
```

Edit the bash.bashrc file on all nodes using vim /etc/bash.bashrc command and append the following environment variable declarations at the end of it.

```
export JAVA_HOME=$(readlink -f /usr/bin/java | sed "s:bin/java::")
export HADOOP_INSTALL=/usr/local/hadoop
export PATH=$PATH:$HADOOP_INSTALL/bin
export PATH=$PATH:$HADOOP_INSTALL/sbin
export HADOOP_MAPRED_HOME=$HADOOP_INSTALL
export HADOOP_COMMON_HOME=$HADOOP_INSTALL
export HADOOP_HDFS_HOME=$HADOOP_INSTALL
export YARN_HOME=$HADOOP_INSTALL
export HADOOP_HOME=$HADOOP_INSTALL
export HADOOP_CONF_DIR=${HADOOP_HOME}"/etc/hadoop"

export ZOOKEEPER_HOME=/usr/local/zookeeper

export SPARK_HOME=/usr/local/spark
export ACCUMULO_HOME=/usr/local/accumulo
export GEOMESA_HOME=/usr/local/geomesa
```

After saving the .profile file. use the following command to source it:

```
source /etc/bash.bashrc
```

Change JAVA_HOME variable value in $HADOOP_CONF_DIR/hadoop-env.sh

```
export JAVA_HOME=$(readlink -f /usr/bin/java | sed "s:bin/java::")
```

Configuring Namenode (115.146.87.101)

```
mkdir -pv $HADOOP_INSTALL/data/namenode
mkdir -pv $HADOOP_INSTALL/logs
```

Edit $HADOOP_CONF_DIR/hdfs-site.xml file and replace configuration content with the following:

```
<configuration>
<property>
<name>dfs.namenode.name.dir</name>
<value>file:///usr/local/hadoop/data/namenode</value>
<description>NameNode directory for namespace and transaction logs storage.</description>
</property>
<property>
<name>dfs.replication</name>
<value>3</value>
</property>
<property>
<name>dfs.permissions</name>
<value>false</value>
</property>
<property>
<name>dfs.datanode.use.datanode.hostname</name>
<value>false</value>
</property>
<property>
<name>dfs.namenode.datanode.registration.ip-hostname-check</name>
<value>false</value>
</property>
<property>
<name>dfs.namenode.http-address</name>
<value>115.146.87.101:50070</value>
<description>Your NameNode hostname for http access.</description>
</property>
<property>
<name>dfs.namenode.secondary.http-address</name>
<value>115.146.87.101:50090</value>
<description>Your Secondary NameNode hostname for http access.</description>
</property>
</configuration>
```

Edit the $HADOOP_CONF_DIR/slaves file and replace its content with the following:

```
115.146.87.105
115.146.87.103
```

Configuring DataNodes (repeat the following for all Datanodes}:

```
mkdir -pv $HADOOP_INSTALL/data/datanode
mkdir -pv $HADOOP_INSTALL/logs
```

Edit $HADOOP_CONF_DIR/hdfs-site.xml file and replace configuration content with the following:

```
<configuration>
<property>
<name>dfs.datanode.data.dir</name>
<value>file:///usr/local/hadoop/data/datanode</value>
<description>DataNode directory</description>
</property>

<property>
<name>dfs.replication</name>
<value>3</value>
</property>
<property>
<name>dfs.permissions</name>
<value>false</value>
</property>
<property>
<name>dfs.datanode.use.datanode.hostname</name>
<value>false</value>
</property>
<property>
<name>dfs.namenode.http-address</name>
<value>115.146.87.101:50070</value>
<description>Your NameNode hostname for http access.</description>
</property>
<property>
<name>dfs.namenode.secondary.http-address</name>
<value>115.146.87.101:50090</value>
<description>Your Secondary NameNode hostname for http access.</description>
</property>
</configuration>
```

Apply remaining configuration to every node:

Open the $HADOOP_CONF_DIR/core-site.xml file and and replace configuration content with the following:

```
<configuration>
<property>
<name>fs.defaultFS</name>
<value>hdfs://115.146.87.101/</value>
<description>NameNode URI</description>
</property>
</configuration>
```

Open the $HADOOP_CONF_DIR/mapred-site.xml file and and replace configuration content with the following:

```
<configuration>
<property>
<name>mapreduce.framework.name</name>
<value>yarn</value>
</property>
</configuration>
```

Open the $HADOOP_CONF_DIR/yarn-site.xml file and and replace configuration content with the following:

```
<configuration>
<property>
<name>yarn.nodemanager.aux-services</name>
<value>mapreduce_shuffle</value>
</property>
<property>
<name>yarn.nodemanager.aux-services.mapreduce_shuffle.class</name>
<value>org.apache.hadoop.mapred.ShuffleHandler</value>
</property>
<property>
<name>yarn.resourcemanager.resource-tracker.address</name>
<value>115.146.87.101:8025</value>
</property>
<property>
<name>yarn.resourcemanager.scheduler.address</name>
<value>115.146.87.101:8030</value>
</property>
<property>
<name>yarn.resourcemanager.address</name>
<value>115.146.87.101:8050</value>
</property>
</configuration>
```

Before starting the cluster we need to be on Namenode machine and format the Namenode by running the following command: hdfs namenode -format

Spark installation

Download the pre-build spark for Hadoop 2.6 and extract it in all nodes.

```
wget http://www.apache.org/dyn/closer.cgi/spark/spark-1.3.1/spark-1.3.1-bin-hadoop2.6.tgz
tar -xvf spark-1.3.1-bin-hadoop2.6.tgz
chown -R hduser spark-1.3.1-bin-hadoop2.6
mv spark-1.3.1-bin-hadoop2.6 /usr/local/spark
```

Create a file named "slaves" in /usr/local/spark/conf in master node and put the following IP addresses in it, one per line:

```
115.146.87.105
115.146.87.103
```

Edit the ~/.profile file and append the following line at the end, then source it as explained before in all nodes.

```
export SPARK_HOME=/usr/local/spark
```

Zookeeper installation

Get the latest Zookeeper package from http://hadoop.apache.org/zookeeper/releases.html and install it in /usr/local/zookeeper in master node.

Create a file "zoo.cfg" in /usr/local/zookeeper/conf and copy the following configuration in it:

```
tickTime=2000
dataDir=/var/zookeeper/
clientPort=2181
initLimit=5
syncLimit=2
server.1=115.146.87.101:2888:3888
server.2=115.146.87.105:2888:3888
server.3=115.146.87.103:2888:3888
```

Copy /usr/local/zookeeper to all other nodes and put it in the same location

For every node, create a file "/var/zookeeper/myid" and put in it the corresponding node id defined in zoo.cfg configuration file. For the master node, the file will contain 1, for second node the file will contain 2, and so on.

Add the $ZOOKEEPER_HOME environment variable (with the value being /usr/local/zookeeper) to the ~/.profile file and source it.

Accumulo installation

Download the binary of version of Accumulo 1.5.1 and unpack it to /usr/local/accumulo in the master node.

Add the $ACCUMULO_HOME environment variable (with the value being /usr/local/accumulo) to the ~/.profile file and source it in all nodes.

Copy the following files from $ACCUMULO_HOME/conf/example folder to $ACCUMULO_HOME/conf in the master node.

```
$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/conf512MB/standalone/accumulo-site.xml

$ACCUMULO_HOME/conf/512MB/standalone/log4j.properties
```

Edit the $ACCUMULO_HOME/conf/masters file and replace the line containing "localhost" with the IP address of master node.

Edit the $ACCUMULO_HOME/conf/slaves file and replace the line containing "localhost" with the IP addresses of slave nodes, one per line.

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
```

Configure accumulo-env.sh to set JAVA_HOME, HADOOP_HOME, and ZOOKEEPER_HOME variables.

Copy /usr/local/accumulo from master to other DataNodes and place it in the same location.

Follow these steps on the master node to start and initialise Accumulo:

1- Start HDFS (if not started already) by issuing {{$HADOOP_HOME/sbin/start-dfs.sh}} in the master node.

2- Start Accumulo by issuing {{$ACCUMULO_HOME/bin/start-all.sh}}

3- Initialise the instance by executing {{$ACCUMULO_HOME/bin/accumulo init}} and set its name to "tweeter" and password to "tweeter"

4- 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

GeoMesa installation

Clone master branch of GeoMesa: git clone https://github.com/locationtech/geomesa.git

Move in the GeoMesa directory and build it using this command: mvn clean install

After the building has completed, copy geomesa directory to /usr/local/geomesa in all nodes and set **$GEOMESA_HOME** environment variable in ~/.profile file and source it.

Copy the geomesa-distributed-runtime-accumulo1.5-1.0.0-rc.5-SNAPSHOT.jar to $ACCUMULO_HOME lib/ext in all nodes and finally restart Accumulo.

Firewall Rules

Open the following ports on the master(NameNode) instance: 8080 (spark web console), 8020, 50095

Open the following ports on all slave(DataNode) instances: 2181,2888, 3888,7077,9997, 50010