Installationsanleitung Accumulo

Vorraussetzungen:

- Ubuntu 14.04 Server (vorzugsweise 32 Bit)
- sudo User
- mindestens 2GB swap space

Vorbemerkung:
Sollten eine oder mehrere Versionen der genannten Programme nicht mehr erhältlich sein, empfiehlt es sich eine neuere, der genannten aber möglichst nahe Version zu wählen.

1. Schritt - Installation JDK 8:
Accumulo, HDFS und ZooKeeper sind alle in Java programmiert, also wird eine JVM benötigt.

Da Java 8 (noch) nicht direkt über den Befehl "apt-get" bezogen werden kann, muss erst die Quelle hierfür bestimmt werden.

```
sudo add-apt-repository ppa:webupd8team/java
```

Updaten sie den package list index.

```
sudo apt-get update
```

Installieren sie OpenJDK via apt-get.

```
sudo apt-get install oracle-java8-installer
```

Editieren sie das shell environment file ".bashrc".

```
nano ~/.bashrc
```

Fügen sie die Umgebungsvariable `JAVA_HOME` am Ende der Datei ein.

```
export JAVA_HOME=/usr/lib/jvm/java-8-oracle
```

Anmerkung: Der Wert für `JAVA_HOME` kann sich je nach gewählter Version unterscheiden. Wurde Java von einer anderen Quelle bezogen, muss unter Umständen auch beachtet werden, ob es sich um eine 32Bit- oder eine 64Bit-Version handelt.

Speichern sie die Änderungen und verlassen sie die Editierung.
Nun müssen die Umgebungsvariablen der aktuellen Sitzung erneuert werden.

```
. ~/.bashrc
```

Editieren sie das `java.security`-File der JVM

```
sudo nano $JAVA_HOME/jre/lib/security/java.security
```

Suchen sie nach dem Parameter `securerandom.source` und editieren sie die Zeile, so dass sie wie folgt aussieht:

```
securerandom.source=file:/dev/./urandom
```

Speichern und verlassen sie nun die Datei. Diese Änderung ist nötig um die Startzeit der JVM zu verringern. Ohne können sich sehr lange Wartezeiten auf den meisten virtuellen Servern ergeben.

2. Schritt - Installation SSH
Hadoop benötigt SSH und Rsync für dessen daemons. Installieren sie sie über folgenden Befehl:

```
sudo apt-get install ssh rsync
```

3. Schritt - Ermöglichen der passwortlosen SSH-Verbindung
Hadoop sollte die Möglichkeit haben ohne Passwort über SSH auf ihren Server zugreifen zu können.

Generieren sie einen RSA-Key.

```
ssh-keygen -P ''
```

Drücken sie ENTER um die Standardwerte zu wählen.
Fügen sie den Schlüssel nun dem `authorized_keys` -File hinzu.

```
cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
```

Die Werte `localhost` und `0.0.0.0` sollten nun den bekannten hosts hinzugefügt werden. Der einfachste Weg ist die Benutzung des `ssh` -Befehls.

Zunächst `localhost:`

```
ssh localhost
```

Sie sollten nun eine Ausgabe erhalten die wie folgt aussieht:

```
The authenticity of host 'localhost (127.0.0.1)' can't be
established.
ECDSA key fingerprint is
bf:01:63:5b:91:aa:35:db:ee:f4:7e:2d:36:e7:de:42.
Are you sure you want to continue connecting (yes/no)?
```

Geben sie `yes` ein und drücken sie anschließend ENTER.

Sobald der Login vollendet wurde, verlassen sie die SSH-Child-Session mit dem Befehl

```
exit
```

Selbiges führen sie nun für `0.0.0.0` aus.

```
ssh 0.0.0.0
```

Wiederum `yes` und ENTER.

Verlassen sie nun wieder mit dem Befehl `exit` die SSH-Child-Session.

Schritt 4 - Erstellen eines "Download"-Verzeichnisses
Dieser Schritt ist nicht zwingend notwendig, trägt jedoch einiges zur Übersicht im System und in dieser Anleitung bei. Erstellen sie ein neues Verzeichnis "Download".

```
mkdir -p ~/Downloads
```

Öffnen sie das Verzeichnis.

```
cd ~/Downloads
```

Schritt 5 - Downloads
Nutzen sie den `wget`-Befehl um die benötigte Software zu laden.

Apache Hadoop:

```
wget "https://archive.apache.org/dist/hadoop/common/hadoop-
2.2.0/hadoop-2.2.0.tar.gz"
```

Apache ZooKeeper:

```
wget "http://www.eu.apache.org/dist/zookeeper/stable/
zookeeper-3.4.6.tar.gz"
```

Apache Accumulo:

```
wget "http://archive.apache.org/dist/accumulo/1.6.4/
accumulo-1.6.4-bin.tar.gz"
```

Schritt 6 - Erstellen eines Installationsverzeichnisses
Aus den selben Gründen, aus denen sich ein "Download"-Verzeichnis empfiehlt, erzeugen sie am besten auch ein "Installations"-Verzeichnis.

```
mkdir -p ~/Installs
```

Öffnen sie anschließend das Verzeichnis

```
cd ~/Installs
```

Schritt 7 - Installation Hadoop
Mit Hilfe des `tar`-Befehls können sie die Download-Datei entpacken.

```
tar -xvzf ~/Downloads/hadoop-2.2.0.tar.gz
```

> **Anmerkung:** Wenn sie eine andere Version geladen haben, muss der Befehl, und alle folgenden, natürlich dementsprechend angepasst werden.

Editieren sie die Datei `hadoop-env.sh`.

```
nano ~/Installs/hadoop-2.2.0/etc/hadoop/hadoop-env.sh
```

Suchen sie nach der Zeile, welche mit `export JAVA_HOME` beginnt und ändern sie sie wie folgt:

```
export JAVA_HOME=/usr/lib/jvm/java-8-oracle
```

Beachten sie, dass sie hier den selben Pfad benötigen, den sie auch in `.bashrc` eingetragen haben.
Standardmäßig generiert Hadoop sehr viele Debug Logs. Um das zu verhindern, suchen sie nach der Zeile, welche mit `export HADOOP_OPTS` beginnt und ändern sie zu folgendem:

```
export HADOOP_OPTS="$HADOOP_OPTS -XX:-PrintWarnings -
Djava.net.preferIPv4Stack=true"
```

Speichern und verlassen sie die Datei.

Editieren sie außerdem die Datei `core-site.xml`.

```
nano ~/Installs/hadoop-2.2.0/etc/hadoop/core-site.xml
```

Fügen sie einen `<property>`-Block mit Namen `fs.defaultFS` ein. Sein `value` sollte auf Hostname und Port der namenode verweisen (hier localhost und Port 9000). Die Datei sollte dann (mit Ausnahme der Kommentare) so aussehen:

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<configuration>
    <property>
       <name>fs.defaultFS</name>
       <value>hdfs://localhost:9000</value>
   </property>
</configuration>
```

Speichern und verlassen sie die Datei.

Editieren sie die Datei `hdfs-site.xml`.

```
nano ~/Installs/hadoop-2.2.0/etc/hadoop/hdfs-site.xml
```

Die folgenden Werte müssen dieser Datei hinzugefügt werden:

- `dfs.replication`: Diese Zahl gibt an, wie oft ein block von Hadoop repliziert wird. Per default erzeugt Hadoop 3 Instanzen. In dieser Anleitung setzen wir den Wert auf 1, da wir kein Cluster erzeugen.
- `dfs.name.dir`: Dies verweist auf das Verzeichnis, in dem die namenode die name table ablegt. Da Hadoop standardmäßig /tmp benutzt muss dieser Wert verändert werden. Wir verwenden hier `hdfs_storage/name` als Pfad, welcher aber beliebig verändert werden kann.
- `dfs.data.dir`: In diesem Verzeichnis speichert die datanode ihre blocks. Wieder muss der Pfad von /tmp abgeändert werden. Der Pfad `hdfs_storage/data` kann wieder nach belieben ausgetauscht werden.

Ohne die Kommentare sollte die Datei nach den Änderungen so aussehen:

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<configuration>
   <property>
       <name>dfs.replication</name>
      <value>1</value>
   </property>
   <property>
      <name>dfs.name.dir</name>
      <value>hdfs_storage/name</value>
   </property>
   <property>
      <name>dfs.data.dir</name>
      <value>hdfs_storage/data</value>
   </property>
</configuration>
```

Erzeugen sie eine neue Datei `mapred-site.xml` mit dem `nano`-Befehl

```
nano ~/Installs/hadoop-2.2.0/etc/hadoop/mapred-site.xml
```

Fügen sie eine `property` mit Namen `mapred.job.tracker` ein. Diese `property` beinhaltet den Hostnamen und Port auf dem der MapReduce job tracker ausgeführt wird. Für den Aufbau dieser Anleitung werden die Werte `localhost` und Port `9001` verwendet.

Die Datei sollte so aussehen:

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<configuration>
<property>
     <name>mapred.job.tracker</name>
     <value>localhost:9001</value>
</property>
</configuration>
```

Speichern und verlassen sie die Datei.

Navigieren sie in das Installationsverzeichnis von Hadoop. (Dies ist wichtig, da Hadoop den `hdfs_storage` im aktuellen Verzeichnis erzeugt.)

```
cd ~/Installs/hadoop-2.2.0/
```

Initialisieren sie jetzt die NameNode.

```
~/Installs/hadoop-2.2.0/bin/hdfs namenode -format
```

Es sollte nun eine ganze Menge Konsolenausgaben erscheinen.
Als nächstes starten sie die NameNode.

```
~/Installs/hadoop-2.2.0/sbin/start-dfs.sh
```

Dies kann ein bis zwei Minuten dauern. Sobald der Vorgang abgeschlossen ist, kann über den Browser mit `http://<your-ip>:50070/` auf das Web-Interface der NameNode zugegriffen werden.

Schritt 8 - Installation ZooKeeper
Begeben sie sich ins Installationsverzeichnis.

```
cd ~/Installs
```

Entpacken sie ZooKeeper.

```
tar -xvzf ~/Downloads/zookeeper-3.4.6.tar.gz
```

Benutzen sie die Beispiel-Datei `zoo_sample.cfg` als `zoo.cfg.`

```
cp ~/Installs/zookeeper-3.4.6/conf/zoo_sample.cfg
 ~/Installs/zookeeper-3.4.6/conf/zoo.cfg
```

Öffnen sie `zoo.cfg` und suchen sie nach der Zeile `dataDir=/tmp`. Ändern /tmp zu einem Verzeichnis ihrer Wahl

```
dataDir=~/Installs/zookeeper_data/
```

Starten sie Zookeeper.

```
~/Installs/zookeeper-3.4.6/bin/zkServer.sh start
```

Sie sollten nun folgende Meldung erhalten:

```
JMX enabled by default
Using config: ~/Installs/zookeeper-3.4.6/bin/../conf/zoo.cfg
Starting zookeeper ... STARTED
```

Schritt 9 - Installation Accumulo
Begeben sie sich ins Installationsverzeichnis.

```
cd ~/Installs
```

Entpacken sie Accumulo.

```
tar -xvzf ~/Downloads/accumulo-1.6.4-bin.tar.gz
```

Accumulo biete Beispiel-Konfigurationen für Server mit unterschiedlichen Speichergrößen: 512 MB, 1 GB, 2 GB und 3 GB.
Kopieren sie die Konfigurations-Datei ihrer Wahl in das `conf`-Verzeichnis.

```
cp ~/Installs/accumulo-1.5.4/conf/examples/2GB/
standalone/* ~/Installs/accumulo-1.5.4/conf/
```

Edititieren sie noch einmal `.bashrc`.

```
nano ~/.bashrc
```

Fügen sie folgende Umgebungsvariablen ein:

- `HADOOP_HOME`: Installationsverzeichnis von Hadoop
- `ZOOKEEPER_HOME`: Installationsverzeichnis von ZooKeeper

Fügen sie folgende Zeilen ans Ende der Datei an:

```
export HADOOP_HOME=~/Installs/hadoop-2.2.0/
export ZOOKEEPER_HOME=~/Installs/zookeeper-3.4.6/
```

Speichern und verlassen sie die Datei und führen sie anschließend aus.

```
. ~/.bashrc
```

Editieren sie die Datei `accumulo-env.sh`.

```
nano ~/Installs/accumulo-1.6.4/conf/accumulo-env.sh
```

Um Accumulo's HTTP monitor über das Netzwerk aufrufen zu können, müssen sie die Zeile `ACCUMULO_MONITOR_BIND_ALL to true` entkommentieren.

```
export ACCUMULO_MONITOR_BIND_ALL="true"
```

Speichern und verlassen sie die Datei.

Editieren sie `accumulo-site.xml`.

```
nano ~/Installs/accumulo-1.6.4/conf/accumulo-site.xml
```

Accumulo nutzt intern ein eigenes Passwort. Dieses sollte zu einem sicheren Schlüssel geändert werden. Suchen sie nach `instance.secret` und ändern sie den eingetragenen Wert. Im Beispiel verwenden wir "PASS1234".

```
<property>
    <name>instance.secret</name>
    <value>PASS1234</value>
    <description>A secret unique to a given instance that
      all servers must know in order to communicate with
      one another.
      Change it before initialization. To change it later
      use ./bin/accumulo
       org.apache.accumulo.server.util.ChangeSecret --old
       [oldpasswd] --new [newpasswd], and then update this
      file.
    </description>
</property>
```

Fügen sie als nächstes eine neue `property` mit Namen `instance.volumes` ein. Der Wert gibt an, wo Accumulo seine Daten im HDFS abspeichert. Wir benutzen hier `/accumulo`.

```
<property>
    <name>instance.volumes</name>
    <value>hdfs://localhost:9000/accumulo</value>
</property>
```

Suchen sie nun nach `trace.token.property.password` und wählen sie ein sicheres Passwort. Als Beispiel wurde hier "mypassw" verwendet.

```
<property>
    <name>trace.token.property.password</name>
    <value>mypassw</value>
</property>
```

Speichern und verlassen sie die Datei.
Initialisieren sie Accumulo

```
~/Installs/accumulo-1.6.4/bin/accumulo init
```

Suchen sie sich einen Instanznamen aus und bestätigen sie mit ENTER.
Als nächstes wird das Passwort gefordert, welches sie in `trace.token.property.password` eingetragen haben (mypassw).

Sobald der Vorgang beendet wurde, kann Accumulo gestartet werden.

```
~/Installs/accumulo-1.6.4/bin/start-all.sh
```

Es sollten einige Warnungen ausgegeben werden, da hier eine sehr kleine Instanz erstellt wird. Allerdings haben diese keine Auswirkung auf diese Anleitung.

Sobald der Start abgeschlossen wurde, kann das Web-Interface von Accumulo über `http://<your-server-ip>:50095` aufgerufen werden.

Schritt 10 - Installation GitHub

```
sudo apt-get install git
```

Schritt 11 - Installation GeoMesa
Nun werden die .jar-Datein von GeoMesa ins lokale Maven-Repository installiert.

```
cd ~/Installs

git clone https://github.com/locationtech/geomesa.git

cd geomesa

mvn clean install
```

Schritt 11 - Download einiger Testdaten (2013-14)
Im Folgenden werden die Daten des GDELT-Prokets der Jahre 2013-14 geladen.

> **Anmerkung:** Die gesamten GDELT-Daten umfassen komprimiert über 100GB und über 1TB wenn entpackt. Zu Testzwecken werden hier nur Daten aus dem Zeitraum 2013-2014 geladen, welche unkomprimiert ungefähr 2GB umfassen.

Zunächst wir ein passendes Verzeichnis gewählt, z.B.:

```
cd ~/Downloads

mkdir gdelt && cd gdelt
```

Anschließend werden die Daten geladen und überprüft.

```
wget http://data.gdeltproject.org/events/md5sums

for file in `cat md5sums | cut -d' ' -f3 | grep '^201[34]'`
; do wget http://data.gdeltproject.org/events/$file ; done

md5sum -c md5sums 2>&1 | grep '^201[34]'
```

Im Verzeichnis, in dem sich die .zip-Datein nun befinden werden diese nun entpackt und als gdelt.tsv in Hadoop geladen.

```
(ls -1 *.zip | xargs -n 1 zcat) | hadoop fs -put -
/gdelt/uncompressed/gdelt.tsv
```

Schritt 12 - Ingest der Daten via Map/Reduce
Zunächst wir die .jar des Ingest geladen und installiert.

cd ~/Installs

git clone https://.../mapreduce-ingest.git

cd mapreduce-ingest.git

mvn clean install

Nun wird der Befehl `hadoop jar` benutzt, um den Map/Reduce-Ingest zu starten.

```
hadoop jar geomesa-gdelt/target/geomesa-gdelt-1.0-SNAPSHOT.jar\
   com.example.geomesa.gdelt.GDELTIngest            \
    -instanceId bigdata                             \
    -zookeepers "zoo1, zoo2"                        \
    -user <username> -password <password>           \
    -tableName gdelt -featureName event             \
    -ingestFile hdfs:///gdelt/uncompressed/gdelt.tsv
```

Schritt 13 - Installations Eclipse

Schritt 14 - Installation Apache Tomcat 8

Schritt 14 - Download des Front-Ends
Das Front-End kann über GitHub direkt in Eclipse geöffnet werden. Hierfür wird die Import-Funktion für Projekte benutzt.

```
File -> Import... -> Git -> Projects from Git -> Clone URI
```

Jetzt muss dem Projekt noch über `Properties` ein Tomcat-Server zugewiesen werden. Ist dies erledigt kann BigData.html auf dem Server ausgeführt werden.

Anmerkung: Wurden für eine oder mehrere Instanzen/Logins/etc. andere Werte als hier beschrieben verwendet, so müssen diese in der Datei Jersey.java entsprechend angepasst werden.