**Installing and configuring the Hadoop, Flume, Hive and Spark services.**

**Steps to install and run Apache Flume,Hadoop services**

1. Both Apache flume and Hadoop File System(HDFS) work seamlessly on Linux Operating System so it is recommended for the user to have Flume Agent and HDFS installed on Linux Operating System. However with the users who are using Mac or Windows they need to install any virtual machine application such as VMware Workstation Pro or Oracle Virtual Box. For our project, we have installed VMware workstation pro. One can download the VMware workstation pro setup file from the following link <https://www.vmware.com/products/workstation-pro.html> . Once the file is downloaded one can just setup the software following the steps displayed on the screen .
2. After VMWorkstation Pro software is installed we need to set up a new virtual Machine. Again this is just a simple straight step where the user needs to click on the ‘Create a New Virtual Machine’ button displayed on the home page and the user is asked to provide all the specifications for the virtual machine they wanted to set up. Once all the steps are completed we have successfully set up a new virtual machine.
3. After setting up a new virtual machine, it is now time for us to install and configure Apache Flume on our new Virtual Machine. Apache Flume is an open source application so we can download the software free of cost visiting the following link <https://flume.apache.org/download.html> . There are multiple versions of Apache Flume available but for the project we have used flume version 1.6.0. There are 2 types of Apache Flume setup file available one is with the suffix of bin.tar.gz and the other is src.tar.ga. Among the two we need to choose the file with a suffix bin.tar.gz. After downloading the file it gets a file in a zip format. So we just need to unzip and extract the files out of that zipped folder.
4. Extracting the file will install Flume in the Virtual Machine. So in order to check that Flume has installed correctly in our Virtual Machine we need to open a new terminal and type the following commands

***getit .bashrc***

***source . bashrc***

***flume-ng***

After entering the above commands if the Flume has been installed correctly in our system then it will display all the help commands for Flume agent. After confirming that we have installed Flume correctly then it is now time to configure Flume to extract Twitter data. For that we need to create a config file under the conf folder of the directory where we have installed the Flume application. In our case our conf file is located at the following location

***/usr/local/hadoop-env/flume-1.6.0/conf***

In order in create a conf file we can open any text editor and specify the following properties such as

Flume agent name, source where the data is being extracted, channel which is a temporary data

Staging storage and finally sink information which is our destination where the data is finally stored.

Here is the code snippet of our config file. If you need to explore more on the config file please check for

the file twitter-flume-hdfs.conf on our project package.

***twtagent.sources = Twitter***

***twtagent.channels = memchannel***

***twtagent.sinks = HDFS***

***# Source Information***

***t.sources.Twitter.type = org.apache.flume.source.twitter.TwitterSource***

***twtagent.sources.Twitter.type = com.cloudera.flume.source.TwitterSource***

***# Sink Information***

***twtagent.sinks.HDFS.channel = memchannel***

***twtagent.sinks.HDFS.type = hdfs***

***twtagent.sinks.HDFS.hdfs.path = hdfs://localhost:9000/twitter\_Extract/***

***# Channel Information***

***twtagent.channels.memchannel.capacity = 10000***

***twtagent.channels.memchannel.type = memory***

***twtagent.channels.memchannel.transactionCapacity = 100***

1. After configuring Flume now we need to install the HDFS on our Virtual Machine. Again HDFS is an open source application so one can download it free of cost visiting the following URL

<https://archive.apache.org/dist/hadoop/core/> . There are multiple versions of HDFS available but for our project we have installed hadoop 2.7.0. In the download page we get two types of files one is with the suffix .src.tar.gz and the other is .tar.gz. Among the two we need to choose .tar.gz. Once the file is downloaded we need to untar the file from the terminal. We need to execute the following command in the terminal to untar our application.

***Tar -xvf hadoop-2.7.0.tar.gz -C /usr/local/hadoop-env***

After untarring the file we need to set up environment variables and we can do that editing .bashrc file. In order to edit .bashrc file we can open a terminal and type the following command

**gedit .bashrc**

It will open the .bashrc file in edit mode and we need to set up JAVA path and Hadoop path as follows

***#HADOOP path***

***export HADOOP\_HOME=/usr/local/hadoop-env/hadoop-2.7.0***

***export HADOOP\_PREFIX=/usr/local/hadoop-env/hadoop-2.7.0***

***export HADOOP\_MAPRED\_HOME=${HADOOP\_HOME}***

***export HADOOP\_COMMON\_HOME=${HADOOP\_HOME}***

***export HADOOP\_HDFS\_HOME=${HADOOP\_HOME}***

***export YARN\_HOME=${HADOOP\_HOME}***

***export HADOOP\_CONF\_DIR=${HADOOP\_HOME}/etc/hadoop***

***#Java path***

***# Add Hadoop bin/ directory to PATH***

***export PATH=$PATH:$HADOOP\_HOME/bin:$HADOOP\_HOME/sbin***

After finishing this we need additional configuration on core-site.xml , hdfs-site.xml, yarn-site.xml and mapred-site.xml. Since we have included all the files in our project package so one can just copy the content of those files and create individual files on their instances. Just as FYI all these files needs to be created at the following location

***/usr/local/<YOUR FOLDER NAME WHERE HADOOP IS INSTALLED>/etc/hadoop***

1. Once we have configured everything it is now time to check if HDFS is installed correctly on my machine or not. To check that we need to open a new terminal and enter the following command

***hadoop version***

If the hadoop is installed correctly then it will display us the hadoop version and other additional information of the hadoop application. After the installation of Hadoop we need to create a directory in the hadoop system where our Twitter data will be stored. In order to create a new directory we need to enter the following command in a terminal window

***hadoop fs -mkdir /<YOUR CHOICE OF FOLDER NAME>***

1. After setting up both the FLUME and Hadoop system it is now time to run the services to extract the data from Twitter. In order to start hadoop services we need to enter the following command in the terminal window

***Start-all.sh***

To make sure if all the Hadoop services are started correctly or not we can visit the following URL in the browser

[***http://localhost:50070***](http://localhost:50070)

If the browser redirects to the welcome page of Hadoop system then it is confirmed that all the Hadoop services have started successfully. After starting the Hadoop services we need to start the Flume agent and extract the data from Twitter. To do so we need to enter the following command in the terminal window. When the user is executing the command he/she needs to be on the home directory where the Flume agent is installed

***./bin/flume-ng agent -n <YOUR TWITTER AGENT NAME> -c conf -f conf/twitter-flume-<YOUR FLUME CONFIG FILE NAME> -Dflume.root.logger=INFO,console***

1. Once the process gets started we can visit the URL [***http://localhost:50070***](http://localhost:50070) and under Utilities menu bar ,we choose Browse the file system sub menu. There we can see the folder that we created earlier and when we open that folder we can see some data files that are being extracted. We can just download that file and open it in any text editor to view its content.

**Hive Installation:**

1. Verifying JAVA Installation ($ java -version)
2. Verifying Hadoop Installation ($ hadoop version)
3. Download Hive from [http://apache.petsads.us/hive/hive-0.14.0/.](http://apache.petsads.us/hive/hive-0.14.0/)
4. Make sure Hive is installed and HDFS, Hadoop is configured.

### Extracting and verifying Hive Archive

1. ***$ tar zxvf apache-hive-0.14.0-bin.tar.gz***
2. Copy the extracted files to local directory
3. Set up the Hive environment by copying the following path in ~/.bashrc file.
4. ***export HIVE\_HOME=/usr/local/hive***
5. ***export PATH=$PATH:$HIVE\_HOME/bin***
6. ***export CLASSPATH=$CLASSPATH:/usr/local/Hadoop/lib/\*:.***
7. ***export CLASSPATH=$CLASSPATH:/usr/local/hive/lib/\*:.***
8. Execute ***$ source ~/.bashrc*** to run the .bashrc file.

## Configuring Hive

## To configure Hive with Hadoop, we edit the “hive-env.sh file”, located in the ***$HIVE\_HOME/conf*** directory.

## The following commands redirect to Hive config folder and copy the template file:

***$ cd $HIVE\_HOME/conf***

***$ cp hive-env.sh.template hive-env.sh***

## Edit the hive-env.sh file by appending the following line:

## ***export HADOOP\_HOME=/usr/local/hadoop***

## Hive installation is completed

But we also need an external DB server to configure the Metastore, hence we also install Apache Derby.

### **Downloading Apache Derby:**

## Download the file using following command:

***$ wget http://archive.apache.org/dist/db/derby/db-derby-10.4.2.0/db-derby-10.4.2.0-bin.tar.gz***

## Extracting and verifying Derby archive

***$ tar zxvf db-derby-10.4.2.0-bin.tar.gz***

## Copy the files to /usr/local/derby directory using the cp command.

## Setting up the environment for Derby : edit the ~/.bashrc file and enter the following lines.

***export DERBY\_HOME=/usr/local/derby***

***export PATH=$PATH:$DERBY\_HOME/bin***

***Apache Hive***

***18***

***export CLASSPATH=$CLASSPATH:$DERBY\_HOME/lib/derby.jar:$DERBY\_HOME/lib/derbytools.jar***.

## Create a directory to store Metastore

***$ mkdir $DERBY\_HOME/data***

## **Configuring Metastore of Hive**:

Configuring Metastore means specifying to Hive where the database is stored. You can do this by editing the hive-site.xml file, which is in the $HIVE\_HOME/conf directory. First of all, copy the template file using the following command:

***$ cd $HIVE\_HOME/conf***

***$ cp hive-default.xml.template hive-site.xml***

## Edit hive-site.xml and append the following lines :

***<property>***

***<name>javax.jdo.option.ConnectionURL</name>***

***<value>jdbc:derby://localhost:1527/metastore\_db;create=true </value>***

***<description>JDBC connect string for a JDBC metastore </description>***

***</property>***

## Create a file named jpox.properties and add the following lines into it:

***javax.jdo.PersistenceManagerFactoryClass =***

***org.jpox.PersistenceManagerFactoryImpl***

***org.jpox.autoCreateSchema = false***

***org.jpox.validateTables = false***

***org.jpox.validateColumns = false***

***org.jpox.validateConstraints = false***

***org.jpox.storeManagerType = rdbms***

***org.jpox.autoCreateSchema = true***

***org.jpox.autoStartMechanismMode = checked***

***org.jpox.transactionIsolation = read\_committed***

***javax.jdo.option.DetachAllOnCommit = true***

***javax.jdo.option.NontransactionalRead = true***

***javax.jdo.option.ConnectionDriverName = org.apache.derby.jdbc.ClientDriver***

***javax.jdo.option.ConnectionURL = jdbc:derby://hadoop1:1527/metastore\_db;create = true***

***javax.jdo.option.ConnectionUserName = APP***

***javax.jdo.option.ConnectionPassword = mine***

## Verifying Hive Installation: Before running Hive, you need to create the /tmp folder and a separate Hive folder in HDFS. Here, we use the /user/hive/warehouse folder. You need to set write permission for these newly created folders as shown below: ***chmod g+w***

## Now set them in HDFS before verifying Hive. Use the following commands:

***$ $HADOOP\_HOME/bin/hadoop fs -mkdir /tmp***

***$ $HADOOP\_HOME/bin/hadoop fs -mkdir /user/hive/warehouse***

***$ $HADOOP\_HOME/bin/hadoop fs -chmod g+w /tmp***

***$ $HADOOP\_HOME/bin/hadoop fs -chmod g+w /user/hive/warehouse***

## The following commands are used to verify Hive installation:

***$ cd $HIVE\_HOME***

***$ bin/hive***

## On successful Installation and running the above command, you should see the Hive Terminal.

1. On the Other Hand, we can also download HDP Sandbox 3.0, which is a virtual machine for using HDFS, Hive and other services. More at: https://www.cloudera.com/downloads/hortonworks-sandbox/hdp.html.
2. For more queries regarding this Installation, please follow: https://www.tutorialspoint.com/hive/hive\_installation.html

**Spark Installation:**

1. Verifying Java Installation

***$java -version***

1. Downloading Scala

Download the latest version of Scala from <https://www.scala-lang.org/download/> .

1. Installing Scala

***Extract the Scala tar file : $ tar xvf scala-2.11.6.tgz***

***Set PATH for Scala : $ export PATH = $PATH:/usr/local/scala/bi***

1. Verifying Scala installation

***$scala -version***

1. Downloading Apache Spark

Download the latest version of Spark by visiting https://spark.apache.org/downloads.html.

1. Installing Spark: Extracting Spark tar file using the command below:

***$ tar xvf spark-1.3.1-bin-hadoop2.6.tgz***

1. Moving Spark software files: Use following commands for moving the Spark software files to respective directory (/usr/local/spark).

***$ su –***

***Password:***

***# cd /home/Hadoop/Downloads/***

***# mv spark-1.3.1-bin-hadoop2.6 /usr/local/spark***

***# exit***

1. Setting up the environment for Spark by adding the following line to ~/.bashrc file.

***export PATH=$PATH:/usr/local/spark/bin***

1. Use the following command for sourcing the ~/.bashrc file.

***$ source ~/.bashrc***

1. Verify the Spark Installation:

Use the following command for opening Spark shell.

***$spark-shell***

1. On successful installation, you should see a pyspark terminal. For additional information see:

https://www.tutorialspoint.com/apache\_spark/apache\_spark\_installation.htm