PROJET BIG DATA

Gestion des base de données massive ‘us accident avec Hadoop network , hive , mongodb , nifi , tableau

Réalisé par :

ALAOUI Brahim

UNIVERSITE CHOUAÏB DOUKKALI
FACULTE DES SCIENCES
EL JADIDA
DEPARTEMENT D’INFORMATIQUE

Master

Table des matières

Remerciements

Au terme de ce travail, je tiens à exprimer ma profonde gratitude ainsi que mes sincères remerciements à toute l'équipe pédagogique de UNIVERSITE CHOUAÏB DOUKKALI Faculté des Sciences EL JADIDA, et aux intervenants professionnels responsables de la formation en sciences mathématiques et informatiques, pour avoir assuré la partie théorique de cette formation.

Je souhaite également adresser mes remerciements les plus distingués à Monsieur Arround pour son soutien, son aide et ses conseils précieux qui m'ont guidé tout au long de ce projet. Il n'a jamais hésité à m'orienter avec ses directives avisées, surtout aux moments où je me sentais complètement perdu. Sa présence et ses encouragements m'ont été d'une aide inestimable. Qu'il trouve ici l'expression de mon profond respect et de ma reconnaissance extrême.

Mes remerciements vont également à toutes les personnes qui m'ont aidé, de près ou de loin, à réaliser ce projet. En réponse aux sacrifices de tous ces gens, je leur dédie ce travail, fruit de leur assistance et de leur dévouement indéniable.

Introduction générale

- Ce mémoire présente le fruit d'un travail de trois mois portant sur le monde de big data et ses principes outils . Ce travail nous a permis d'acquérir une expérience importante dans la science big data .

Mon projet est la gestion d’une base de données massive des accidents de US (2016 -2020) avec mongodb sur hadoop distribue et hive

Chapitre 1 : Notions de base

1. Hadoop définition :

Hadoop est un framework logiciel open source permettant de stocker des données, et de lancer ds applications sur des grappes de machines standards. Cette solution offre un espace de stockage massif pour tous les types de données, une immense puissance de traitement et la possibilité de prendre en charge une quantité de tâches virtuellement illimitée. Basé sur Java, ce framework fait partie du projet Apache, sponsorisé par Apache Software Foundation.

2. Hadoop network :

pour cette partie ,nous basons sur le guide du groupe hadoop network

3. Hive :

Apache Hive est un logiciel de Data Warehouse initialement créé par Facebook. Il permet **d'effectuer facilement et rapidement des requêtes » SQL-like «** pour extraire efficacement des données en provenance de Apache Hadoop. Contrairement à Hadoop, Hive permet d'effectuer des requêtes SQL sans avoir besoin d'écrire en Java.

4. Mongodb :

MongoDB est une base de données orientée documents. En clair, vous bénéficiez de la scalabilité et de la flexibilité que vous voulez, avec les fonctions d'interrogation et d'indexation qu'il vous faut.

5. Mongo-hadoop conector :

Le connecteur MongoDB pour Hadoop est une bibliothèque qui permet à MongoDB (ou à des fichiers de sauvegarde dans son format de données, BSON) d'être utilisé comme source d'entrée, ou destination de sortie, pour les tâches Hadoop MapReduce.

6. Tableau

Tableau est un logiciel de data visualization d'origine américaine destiné à aider les utilisateurs à percevoir et à comprendre leurs données, sans l'intervention d'experts en informatiques. Il participe par conséquent à l'autonomie des utilisateurs qui peuvent créer leurs propres analyses et les représenter visuellement pour une meilleure compréhension. En outre, il permet la transformation de données brutes en informations exploitables, mais aussi leur présentation interactive. Ici, le graphisme est particulièrement soigné et le type de données soumise à l'analyse est large. Données SQL ou Hadoop, feuilles de calcul Excel, tous types de données internes à l'entreprise peuvent être confrontées entre elles, ainsi qu'à d'autres données présentes sur le web. L'utilisateur a la possibilité de partager ses analyses en toute facilité sur le web et sur les appareils mobiles. En somme, Tableau est une approche globale et complète de la visualisation de données.

7. **Nifi**

NiFi est un logiciel libre de gestion de flux de données. Il permet de gérer et d'automatiser des flux de données entre plusieurs systèmes informatiques, à partir d'une interface web et dans un environnement distribué.

Chapitre 2 : Outils et l'environnement de travail

- **Oracle vm virtualBox :**

Oracle VM VirtualBox est un logiciel de virtualisation open source qui permet aux utilisateurs d'exécuter plusieurs systèmes d'exploitation sur un seul terminal

Ubuntu : est un système d'exploitation GNU/Linux basé sur la distribution Linux Debian. Il est développé, commercialisé et maintenu pour les ordinateurs individuels (desktop), les serveurs (Server) et les objets connectés (Core) par la société Canonical.

- **Hadoop 3 :**

On a Travaillée avec hadoop 3 monde distribué et aussi hadoop 2

- **Hive 2**

- **Nifi 1**

- **Tableau**

Chapitre 3 : les étapes de projet

I. Installation de hadoop 3

Steps to Install Hadoop 3 on Ubuntu

1-download the Hadoop 3.1.2 from the below link:
https://archive.apache.org/dist/hadoop/common/hadoop-3.1.2/hadoop-3.1.2.tar.gz

2-Install ssh on your system using the below command:

sudo apt-get install ssh

Type the password for the sudo user and then press **Enter**.

Type **'Y'** and then press **Enter** to continue with the installation process.

3-Install pdsh on your system using the below command:

sudo apt-get install pdsh

Type **'Y'** and then press **Enter** to continue with the installation process.

4-Open the .bashrc file in the nano editor using the following command:

nano .bashrc

Now set the **PDSH_RCMD_TYPE** environment variable to **ssh**

export PDSH_RCMD_TYPE=ssh

To save the changes you've made, press **Ctrl+O**. To exit the nano editor, press **Ctrl+X** and then press **'Y'** to exit the editor.

5-Now configure ssh. To do so, create a new key with the help of the following command (don't copy-paste following command, rather type):

ssh-keygen -t rsa -P ""

Press **Enter** when asked the **file name**.

6- Copy the content of the public key to authorized_keys.

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

7- Now examine the SSH setup by connecting to the localhost.

ssh localhost

Type **'Y'** and then press **Enter** to continue with the connection

8- Update the source lists.

sudo apt-get update

9-Now install Java 8 using the following command:

sudo apt-get install openjdk-8-jdk

Type **'Y'** and then press **Enter** to finish with the installation process.

10- To cross-check whether you have successfully installed Java on your machine or not, run the below command:

java –version

11- Download Hadoop from the link given in the first section and copy the setup in your home directory (/home/USER-NAME)

→ Hadoop Installation on Ubuntu

12-Now locate the Hadoop tar file in your system.

13- Extract the **hadoop-3.1.2.tar.gz** file using the below command:

tar xzf hadoop-3.1.2.tar.gz

14- Rename **hadoop-3.1.2.tar.gz** as **hadoop** for ease of use.

mv hadoop-3.1.2.tar.gz hadoop

15- Now check the Java home path

ls /usr/lib/jvm/java-8-openjdk-amd64/

16- Open the **hadoop-env.sh** file in the nano editor. This file is located in **~/hadoop/etc/hadoop** (Hadoop configuration directory).

nano hadoop-env

Now, Set JAVA_HOME path:

export JAVA_HOME=<path-to-the-root-of-your-Java-installation> (eg: /usr/lib/jvm/java-8-openjdk-amd64/

To save the changes you've made, press **Ctrl+O**. To exit the nano editor, press **Ctrl+X** and then press **'Y'** to exit the editor.

17- Open the **core-site.xml** file in the nano editor. This file is also located in the **~/hadoop/etc/hadoop** (Hadoop configuration directory).

nano core-site.xml

Add the following configuration properties:

```
<configuration>
<property>
<name>fs.defaultFS</name>
<value>hdfs://localhost:9000</value>
</property>
<property>
<name>hadoop.tmp.dir</name>
<value>/home/dataflair/hdata</value>
</property>
</configuration>
```

Note: /home/dataflair/hdata is a sample location; please specify a location where you have Read Write privileges

18-Open the **hdfs-site.xml** file in the nano editor. This file is also located in **~/hadoop/etc/hadoop** (Hadoop configuration directory):

nano hdfs-site.xml

Add the following entries in core-site.xml:

```
<configuration>
<property>
<name>dfs.replication</name>
<value>1</value>
</property>
</configuration>
```

19- Open the **mapred-site.xml** file in the nano editor. This file is also located in **~/hadoop/etc/hadoop** (Hadoop configuration directory).

nano mapred-site.xml

Add the following entries in core-site.html:

```
<configuration>
<property>
```

<name>mapreduce.framework.name</name>
<value>yarn</value>
</property>
<property>
<name>yarn.app.mapreduce.am.env</name>
<value>HADOOP_MAPRED_HOME=/home/dataflair/hadoop</value>
</property>
<property>
<name>mapreduce.map.env</name>
<value>HADOOP_MAPRED_HOME=/home/dataflair/hadoop</value>
</property>
<property>
<name>mapreduce.reduce.env</name>
<value>HADOOP_MAPRED_HOME=/home/dataflair/hadoop</value>
</property>
</configuration>

20-Open the **yarn-site.xml** file in the nano editor. This file is also located in **~/hadoop/etc/hadoop** (Hadoop configuration directory).

nano yarn-site.xml

Add the following entries in the yarn-site.xml:

<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>
</configuration>

21-Open the bashrc files in the nano editor using the following command:

nano .bashrc

Edit .bashrc file located in the user's home directory and add the following parameters:

```
export HADOOP_HOME="/home/dataflair/hadoop"
export PATH=$PATH:$HADOOP_HOME/bin
export PATH=$PATH:$HADOOP_HOME/sbin
export HADOOP_MAPRED_HOME=${HADOOP_HOME}
export HADOOP_COMMON_HOME=${HADOOP_HOME}
export HADOOP_HDFS_HOME=${HADOOP_HOME}
export YARN_HOME=${HADOOP_HOME}
```

To save the changes you've made, press **Ctrl+O**. To exit the nano editor, press **Ctrl+X** and then press **'Y'** to exit the editor.

Now, source the bashrc file so that the changes will come into effect:

source ~/.bashrc

22-Before starting Hadoop, we need to format HDFS, which can be done using the below command:

hdfs namenode –format

23-Start the HDFS services:

sbin/start-dfs.sh

24-Open the HDFS web console:

localhost:9870

25-Now start the yarn services:

sbin/start-yarn.sh

The **'jps'** command is used to check whether all the Hadoop processes are running or not.

$jps
NameNode
DataNode
ResourceManager
NodeManager
SecondaryNameNode

26-Open the yarn web console:

localhost:8088

Version 2 installation

For installing Hadoop you need to install Java on Ubuntu.

Commands to install Java

java -version

sudo apt-get install update

sudo apt-get update

sudo apt-get install default-jdk

Check Java Version

java -version

Adding New User

sudo addgroup hadoop

sudo adduser --ingroup hadoop hadoopuser

sudo adduser hadoopuser sudo

sudo apt-get install openssh-server

su - hadoopuser

Generating ssh key

ssh-keygen -t rsa -P “”

cat $HOME .ssh/id_rsa.pub >> $HOME/.ssh/authorized_keys

ssh localhost

(yes)

Download Hadoop from this link

https://hadoop.apache.org/release/3.2.1.html

*** Paste the .tar file on Desktop ****

*** Move to Desktop in Terminal****

Commands

sudo tar -xvzf (hadoop filename)

sudo mv (hadoop Folder name) /usr/local/hadoop

sudo chown -R hadoopuser /usr/local

Change File Configurations

Change ~/.bashrc file

sudo nano ~/.bashrc

**** Content to be copied *****

export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64

export HADOOP_HOME=/usr/local/hadoop

export PATH=$PATH:$HADOOP_HOME/bin

export PATH=$PATH:$HADOOP_HOME/sbin

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_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/native

export HADOOP_OPTS="-Djava.library.path=$HADOOP_HOME/native"

---

source ~/.bashrc

Change hadoop-env.sh file

sudo nano /usr/local/hadoop/etc/hadoop/hadoop-env.sh

**** Content to be copied *****

export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64

Change core-site.xml file

sudo nano /usr/local/hadoop/etc/hadoop/core-site.xml

**** Content to be copied *****

<property>

<name>fs.default.name</name>

<value>hdfs://localhost:9000</value>

</property>

Change hdfs-site.xml file

sudo nano /usr/local/hadoop/etc/hadoop/hdfs-site.xml

**** Content to be copied *****

<property>

<name>dfs.replication</name>

<value>1</value>

</property>

<property>

<name>dfs.namenode.name.dir</name>

<value>file:/usr/local/hadoop_tmp/hdfs/namenode</value>

</property>

<property>

<name>dfs.datanode.data.dir</name>

<value>file:/usr/local/hadoop_tmp/hdfs/datanode</value>

</property>

Change yarn-site.xml file

sudo nano /usr/local/hadoop/etc/hadoop/yarn-site.xml

**** Content to be copied *****

<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>

Change mapred-site.xml file

sudo nano /usr/local/hadoop/etc/hadoop/mapred-site.xml

**** Content to be copied *****

<property>

<name>mapreduce.framework.name</name>

<value>yarn</value>

</property>

Now Run these commands to create directories

sudo mkdir -p /usr/local/hadoop_space

sudo mkdir -p /usr/local/hadoop_space/hdfs/namenode

sudo mkdir -p /usr/local/hadoop_space/hdfs/datanode

sudo chown -R hadoopuser /usr/local/hadoop_space

Running Hadoop:

cd

hdfs namenode -format

start-dfs.sh

start-yarn.sh

jps

II. Installation de Hive

Download hive .tar file from https://hive.apache.org/downloads.html

Extract the .tar file and Move it to some directory.

sudo mv apache-hive-3.1.2-bin /usr/local/hive/

Now set some variables in the `bashrc` file

nano ~/.bashrc

Add the following lines to this file

export HIVE_HOME={path of hive folder}

export PATH= $HIVE_HOME/bin

export HIVE_CONF_DIR=$HIVE_HOME/conf

Now run the following step to activate the above changes.

source ~/.bashrc

Create the hive-default.xml file.

sudo touch /usr/local/hive/conf/hive-default.xml

Copy the contents of hive-default.xml.template file into hive-default.xml file.

sudo cp /usr/local/hive/conf/hive-default.xml.template /usr/local/hive/conf/hive-default.xml

Copy the contents of hive-default.xml.template file into hive-site.xml file.

sudo cp /usr/local/hive/conf/hive-default.xml /usr/local/hive/conf/hive-site.xml

Check the contents of hive-default.xml file.

sudo cat /usr/local/hive/conf/hive-default.xml

Copy the contents of hive-env.sh..template file into hive-env.sh file.

sudo cp /usr/local/hive/conf/hive-env.sh.template /usr/local/hive/conf/hive-env.sh

Now enter the HADOOP_HOME path in the hive-env.sh file

sudo nano /usr/local/hive/conf/hive-env.sh

export HADOOP_HOME = {path to hadoop}

Now cd to hadoop home

cd $HADOOP_HOME

Start hadoop

start-dfs.sh

start-yarn.sh

jps

Now make the warehouse directory.

hdfs dfs -mkdir -p /user/hive/warehouse

Give permissions for warehouse directory.

hdfs dfs -chmod g+w /user/hive/warehouse

Similarly make the temp directory.

hdfs dfs -mkdir -p /tmp

hdfs dfs -chmod g+w /tmp

Now cd to HIVE_HOME/bin and run the following command

cd $HIVE_HOME/bin

schematool -dbType derby -initSchema

Start hive

hive

This command will run the Hive Shell.

III. Installation de mongodb

Sudo apt update
Sudo apt install mongodb (Y)
Sudo systemctl status mongodb
Mongo
Mongodb –version
Mongo

```
> show  dbs
> use admin
> show collections
```

IV. Installation de mongo hadoop conector (mongodbConnector r1.4.0-rc0)

Hadoop connection with mongodb using mongoDBConnector

if you are using maven to build your project then follow these steps to process mongodb data with hadoop .

step 1 - Add dependency into your pom.xml file and also download jars which will be required later to run mapreduce programme from command line
click here to download mongodbConnector jars
https://github.com/mongodb/mongo-hadoop/releases

step 2 - Create maven based java project 'HadoopWithMongo'

step 3 - Add mongo-hadoop-core-1.4-rc0 dependency into pom.xml file

step 4 - Add hadoop liberaries into your project classpath

NOTE: hadoop lib folder location vary on the basis of hadoop version.
In Hadoop-2.6.0 use this path "hadoop/share/hadoop/common/lib" , ignore this path "hadooop/lib direcotry"

step 5 - Create java class MongoConnector

step 6 - Write a MapReduce programme

```
import java.io.IOException;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.util.ToolRunner;
import org.bson.BSONObject;

import com.mongodb.hadoop.MongoConfig;
import com.mongodb.hadoop.MongoInputFormat;
import com.mongodb.hadoop.MongoOutputFormat;
import com.mongodb.hadoop.util.MapredMongoConfigUtil;
import com.mongodb.hadoop.util.MongoConfigUtil;
import com.mongodb.hadoop.util.MongoTool;


public class MongoConnector extends MongoTool{
        public static class Map extends Mapper{
                public void map(final Object key, final BSONObject value, final
Context context) throws IOException, InterruptedException{
                        System.out.println(value);
                        /**
                         * write your mapper logic
                         */
                        context.write(new Text(), new IntWritable(1));
                }
        }

        public static class Reduce extends Reducer{
                public void reduce(Text key,Iterable values,Context context) throws
IOException, InterruptedException{
                        /**
                         * write your reducer logic
                         */
                        context.write(new Text(), new IntWritable(1));
                }
        }

        public MongoConnector(){
                Configuration conf = new Configuration();
                MongoConfig mongoConfig = new MongoConfig(conf);
                setConf(conf);
                if (MongoTool.isMapRedV1()) {
          MapredMongoConfigUtil.setInputFormat(getConf(),
com.mongodb.hadoop.mapred.MongoInputFormat.class);
          MapredMongoConfigUtil.setOutputFormat(getConf(),
com.mongodb.hadoop.mapred.MongoOutputFormat.class);
       } else {
          MongoConfigUtil.setInputFormat(getConf(), MongoInputFormat.class);
          MongoConfigUtil.setOutputFormat(getConf(), MongoOutputFormat.class);
       }
                mongoConfig.setInputFormat(MongoInputFormat.class);
```

```
        mongoConfig.setInputURI("mongodb://localhost:27017/dbName.collectionName");
                mongoConfig.setMapper((Class) Map.class);
                mongoConfig.setReducer(Reduce.class);
                mongoConfig.setMapperOutputKey(Text.class);
                mongoConfig.setMapperOutputValue(IntWritable.class);
                mongoConfig.setOutputKey(Text.class);
                mongoConfig.setOutputValue(IntWritable.class);

        mongoConfig.setOutputURI("mongodb://localhost:27017/dbName.outputCollectionName");
                mongoConfig.setOutputFormat(MongoOutputFormat.class);
        }

        public static void main(String[] args) throws Exception {
                System.exit(ToolRunner.run(new MongoConnector(), args));
        }
}
```

NOTE: your mongo instance should be started.

Your connection is setup successfully if you want to run mapreduce programme using jar then follow these steps

step 1 - First of all put mongo connector jars downloaded in first step in hadoop lib directory
step 2 - start hadoop services
step 3 - create jar file of above java project
step 4 - Hit this command - hadoop jar HadoopWithMongo.jar MongoConnector

This will start your mapreduce programme.

V. configure **hive** with mongo-hadoop-core 2.0.2 (**mongo-hadoop-hive 2.0.2)**

https://github.com/mongodb/mongo-hadoop/wiki/Hive-Usage

→Mongo and hive integration

----------Hadoop HA cluster setup and Hive installation----------

1-Hadoop HA high availability cluster construction (2.7.2)

2-Hive installation and use demo

---------- Official start ----------

1-**Installation**

Obtain the MongoDB Hadoop Connector. You can either build it or download the jars. For Hive, you'll need the "core" jar and the "hive" jar.

Get a JAR for the MongoDB Java Driver. The connector requires at least version 3.0.0 of the driver "uber" jar (called "mongo-java-driver.jar").

In your Hive script, use ADD JAR commands to include these JARs (core, hive, and the Java driver), e.g., ADD JAR /path-to/mongo-hadoop-hive-<version>.jar;.

1. The version must come as it requires, the jar package goeshttp://mvnrepository.com/Just download it. Only three are needed to use Hive:
   mongo-hadoop-core-1.5.1.jar
   mongo-hadoop-hive-1.5.1.jar
   mongo-java-driver-3.2.1.jar
2. Copy the jar package to HADOOPHOME/lib and HADOOPHOME/lib and {HIVE_HOME}/lib, then start Hive and add the jar package :

```
[hadoop@DEV21 ~]$ hive

Logging initialized using configuration in jar:file:/home/hadoop/opt/apache-hive-1.2.1-bin/lib/hive-common-1.2.1.jar!/hive-log4j.properties

 hive> add jar /home/hadoop/opt/hive/lib/mongo-hadoop-core-1.5.1.jar;#Add all three, I won't write this.
```

3.Hive Usage has two connection methods:

First, MongoDB-based directly connects to hidden nodes, and uses com.mongodb.hadoop.hive.MongoStorageHandler as data Serde

Second, BSON-based dumps the data into a bson file and uploads it to the HDFS system, using com.mongodb.hadoop.hive.BSONSerDe

3. MongoDB-based approach

```
hive> CREATE TABLE eventlog
    > (
    >   id string,
    >   userid string,
    >   type string,
    >   objid string,
    >   time string,
    >   source string
    > )
    > STORED BY 'com.mongodb.hadoop.hive.MongoStorageHandler'
    > WITH SERDEPROPERTIES('mongo.columns.mapping'='{"id":"_id"}')
    >
TBLPROPERTIES('mongo.uri'='mongodb://username:password@ip:port/xxx.xxxxxx');
hive> select * from eventlog limit 10;
OK
5757c2783d6b243330ec6b25    NULL    shb NULL    2016-06-08 15:00:07 NULL
5757c27a3d6b243330ec6b26    NULL    shb NULL    2016-06-08 15:00:10 NULL
5757c27e3d6b243330ec6b27    NULL    shb NULL    2016-06-08 15:00:14 NULL
5757c2813d6b243330ec6b28    NULL    shb NULL    2016-06-08 15:00:17 NULL
5757ee443d6b242900aead78    NULL    shb NULL    2016-06-08 18:06:59 NULL
5757ee543d6b242900aead79    NULL    smb NULL    2016-06-08 18:07:16 NULL
5757ee553d6b242900aead7a    NULL    cmcs    NULL    2016-06-08 18:07:17 NULL
5757ee593d6b242900aead7b    NULL    vspd    NULL    2016-06-08 18:07:21 NULL
575b73b2de64cc26942c965c    NULL    shb NULL    2016-06-11 10:13:06 NULL
575b73b5de64cc26942c965d    NULL    shb NULL    2016-06-11 10:13:09 NULL
```

Time taken: 0.101 seconds, Fetched: 10 row(s)

La suite : https://www.programmersought.com/article/33174063710/

VI. Data clean

VII. Tableau

Pour l'installation de tableau on a l'installe sur Windows et on a virtulaize les données de US accident

Chapitre 4 : démonstration

Voila quelque capture d'ecrans qui resume notre fonctionment de hive

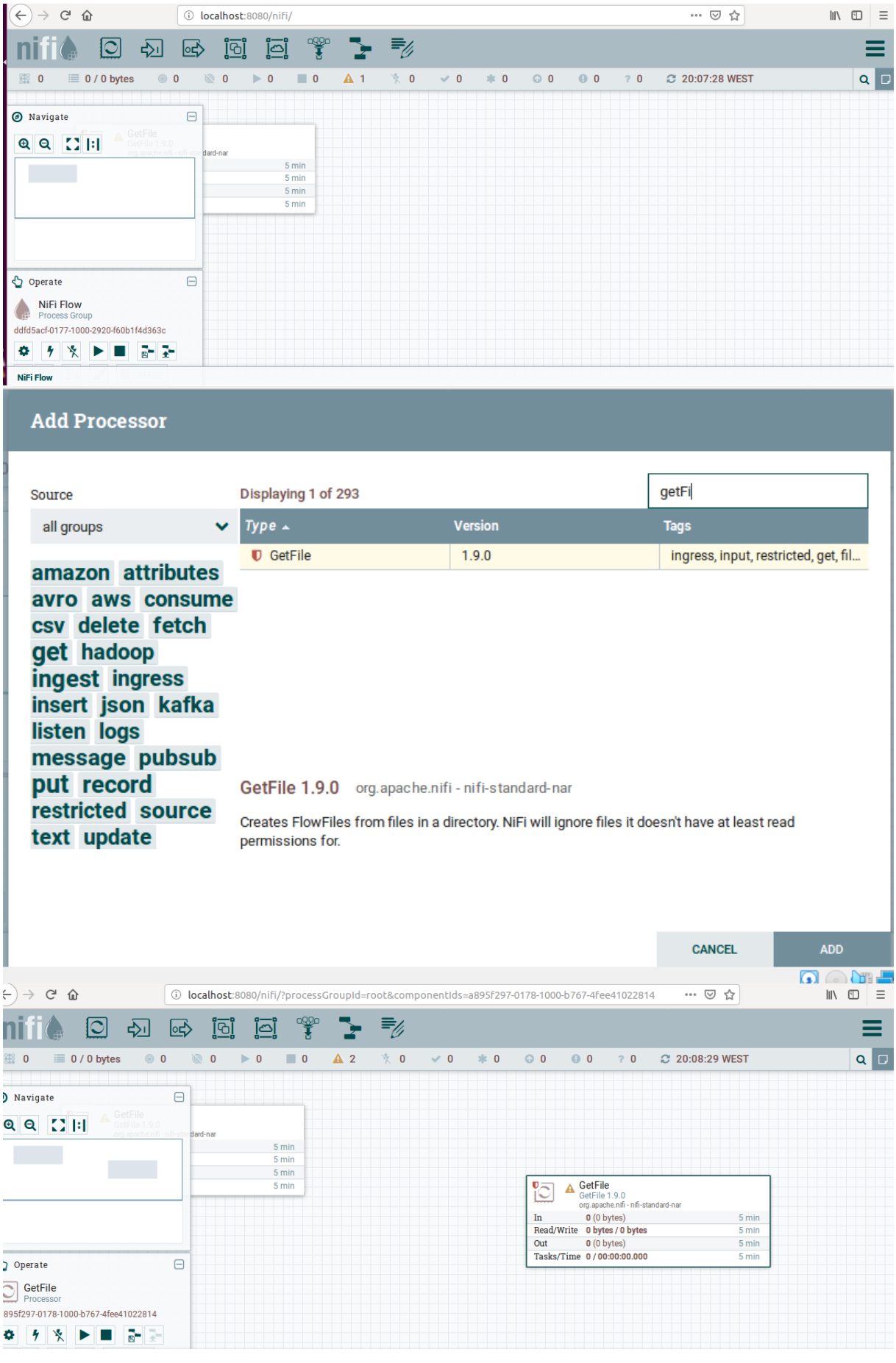
Hive :

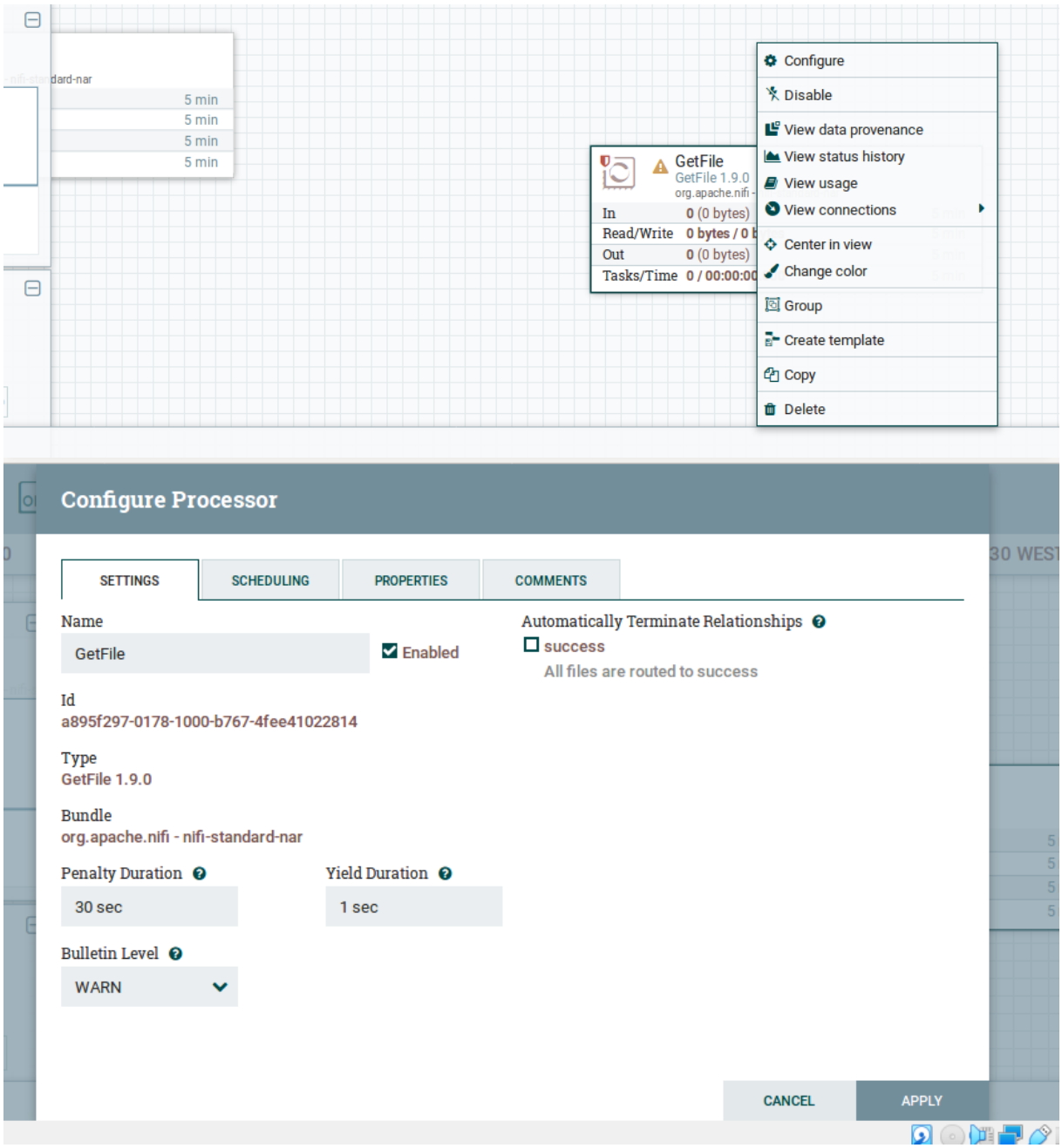
```
hive> !clear;


hive> CREATE  TABLE accidents (ID string,Source string,TMC int,Severity int,Start_Time str
ing,End_Time string,Start_Lat string,Start_Lng string,End_Lat string,End_Lng string,Distan
ce float,Description string,Number int,Street string,Side string,City string,County string
,State string,Zipcode int,Country string,Timezone string,Airport_Code int,Weather_Timestam
p string,Temperature float,Wind_Chill float,Humidity float,Pressure int,Visibility float,W
ind_Direction string,Wind_Speed int,Precipitation boolean,Weather_Condition boolean,Amenit
y boolean,Bump boolean,Crossing boolean,Give_Way boolean,Junction boolean,No_Exit boolean,
Railway boolean,Roundabout string,Station string,Stop boolean,Traffic_Calming string,Traff
ic_Signal string,Turning_Loop boolean,Sunrise_Sunset boolean,Civil_Twilight string,Nautica
l_Twilight string,Astronomical_Twilight string) row format delimited fields terminated by
','  lines terminated by '\n' stored as textfile;
OK
Time taken: 2.579 seconds
hive> load data inpath '/nifi/output/cleaned_new_york_data.csv' into table accidents;
Loading data to table default.accidents
Table default.accidents stats: [numFiles=1, totalSize=4336523]
OK
Time taken: 5.638 seconds
hive>
```

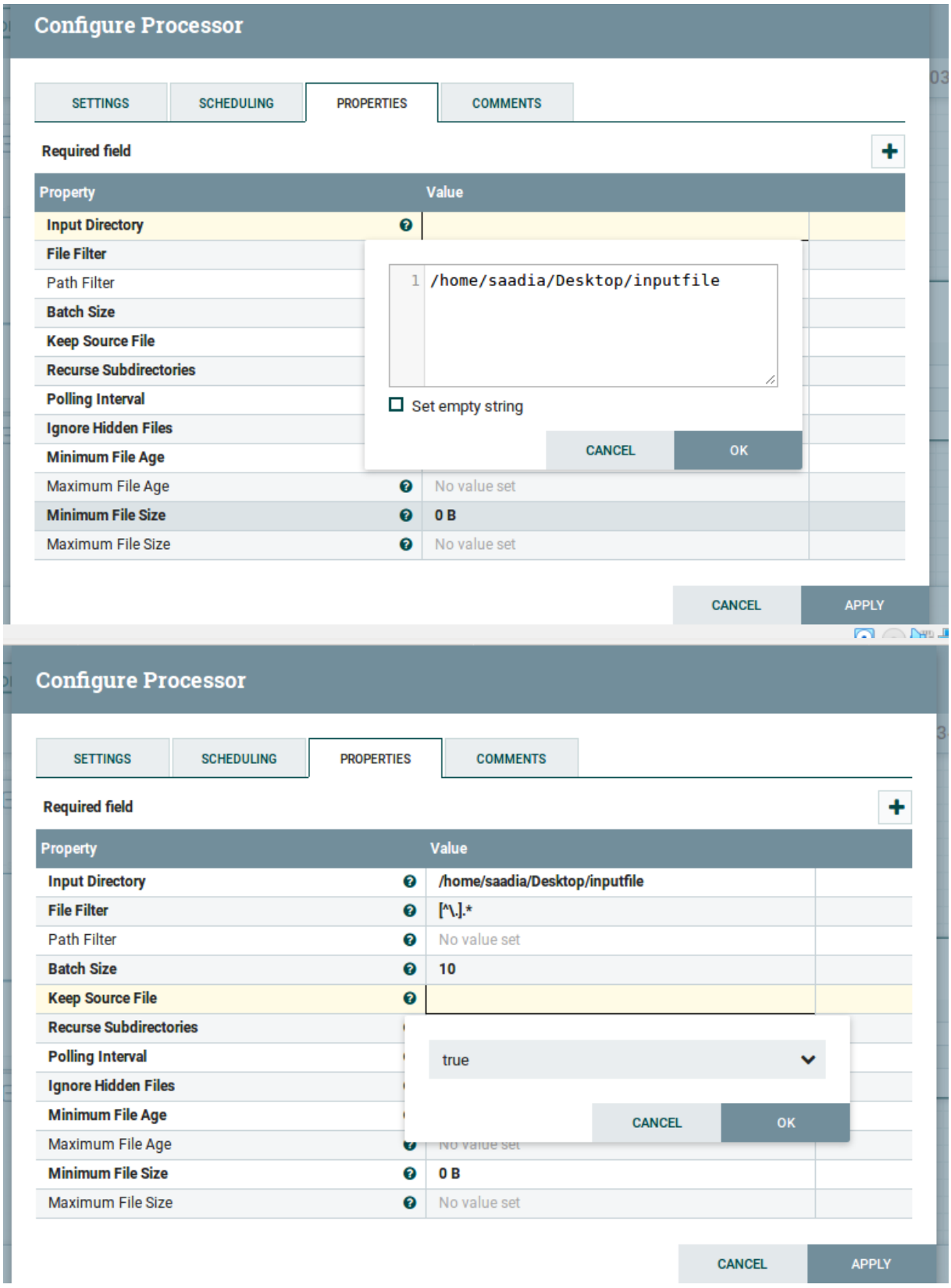
```
Time taken: 5.638 seconds
hive> select * from accidents limit 2;
OK
ID      Source  NULL    NULL    Start_Time      End_Time        Start_Lat       Start_LngD
istance(mi)     Description     NULL    Street  NULL    City    County  State   Zipcode Co
untry   NULL    Airport_Code    Temperature(F)  NULL    Humidity(%)     NULL    NULL    NU
LL      NULL    NULL    Weather_Condition       NULL    NULL    NULL    NULL    NULL    NU
LL      NULL    NULL    NULL    NULL    Traffic_Calming Traffic_Signal  NULL    Sunrise_Su
nset    Civil_Twilight  NULL    NULL    NULL    NULL    NULL
A-194408        MapQuest        201     2       2016-12-01 08:21:11     2016-12-01 08:50:4
8       40.770107       -73.957397      0.01    Accident school bus involved on 2nd Ave bo
th ways at 74th St.     272.0   E 74th St       NULL    New York        New York        NY
10021-3790      US      NULL    KNYC    52.0    50      80.0    29.61   10.0    NULL    50
.12613729       Mostly Cloudy   NULL    false   true    false   false   false   false   fa
lse     false   false   False   True    false   Day     Day     NULL    NULL    NULL    NU
LL      NULL
Time taken: 1.787 seconds, Fetched: 2 row(s)
hive>
```

Nifi :

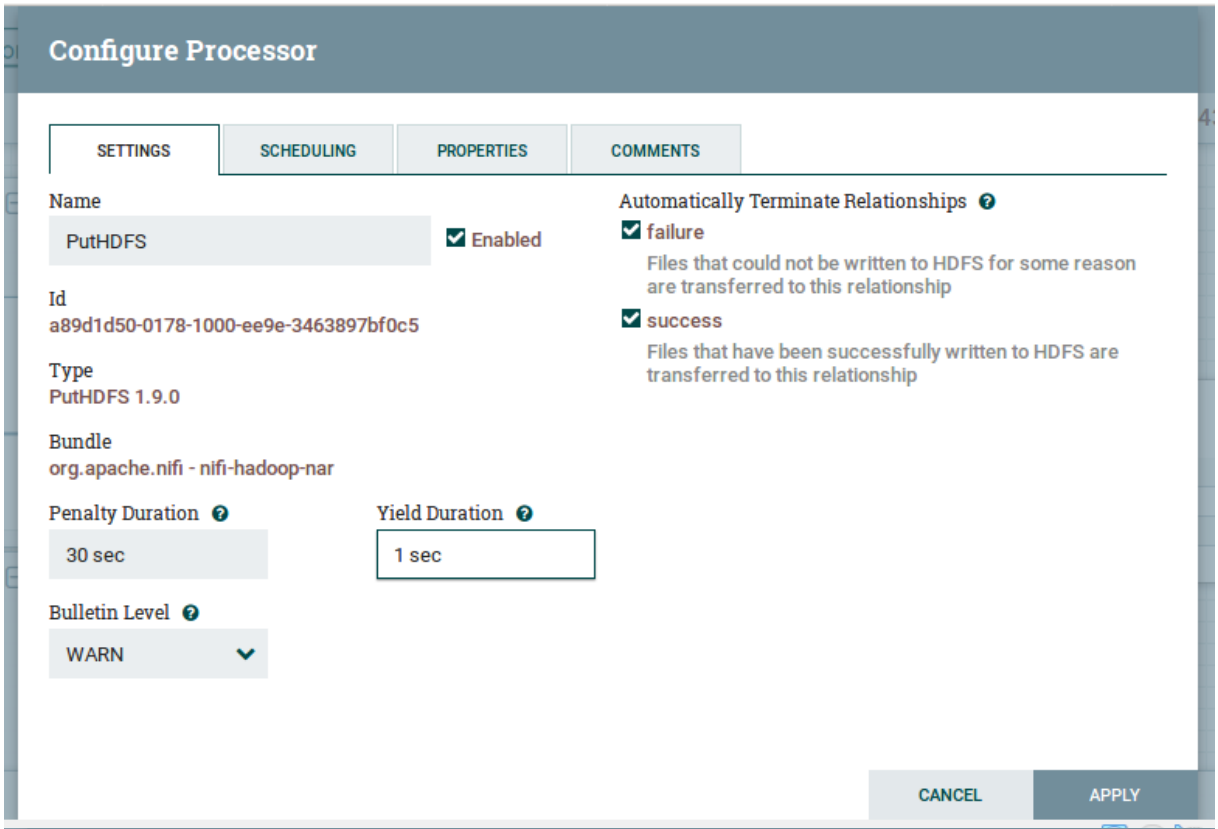
```
saadia@saadia-VirtualBox:~/Desktop/hadoop/nifi-1.9.0/bin$ sudo ./nifi.sh start
[sudo] password for saadia:
nifi.sh: JAVA_HOME not set; results may vary

Java home:
NiFi home: /home/saadia/Desktop/hadoop/nifi-1.9.0

Bootstrap Config File: /home/saadia/Desktop/hadoop/nifi-1.9.0/conf/bootstrap.con
f


saadia@saadia-VirtualBox:~/Desktop/hadoop/nifi-1.9.0/bin$ sudo ./nifi.sh status
nifi.sh: JAVA_HOME not set; results may vary

Java home:
NiFi home: /home/saadia/Desktop/hadoop/nifi-1.9.0

Bootstrap Config File: /home/saadia/Desktop/hadoop/nifi-1.9.0/conf/bootstrap.con
f

2021-04-06 20:01:13,111 INFO [main] org.apache.nifi.bootstrap.Command Apache NiF
i is currently running, listening to Bootstrap on port 36195, PID=5794

saadia@saadia-VirtualBox:~/Desktop/hadoop/nifi-1.9.0/bin$
```

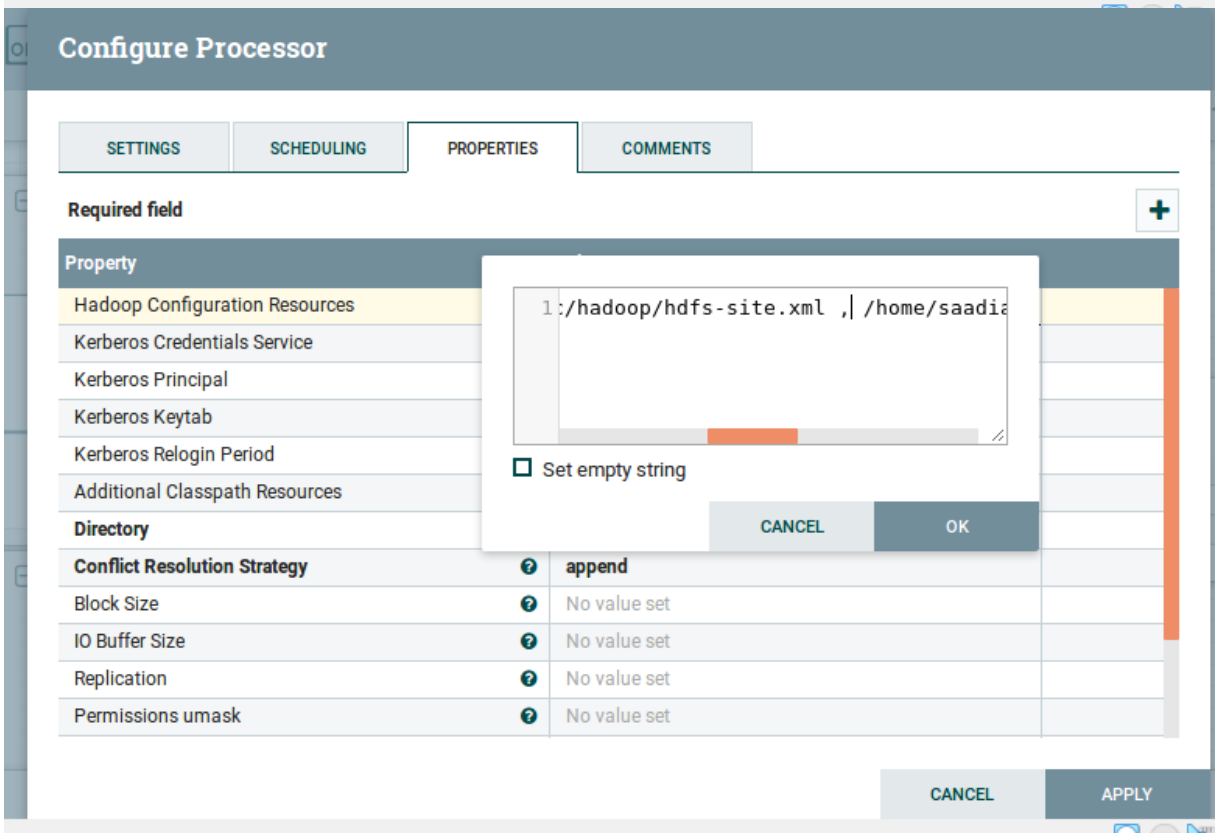

localhost:8080/nifi/
Navigate
Operate
NiFi Flow
Process Group
ddfd5acf-0177-1000-2920-f60b1f4d363c
Add Processor
Source
Displaying 1 of 293
getFi
all groups
Type
Version
Tags
GetFile
1.9.0
ingress, input, restricted, get, fil...
amazon attributes avro aws consume csv delete fetch get hadoop ingest ingress insert json kafka listen logs message pubsub put record restricted source text update
GetFile 1.9.0
org.apache.nifi - nifi-standard-nar
Creates FlowFiles from files in a directory. NiFi will ignore files it doesn't have at least read permissions for.
CANCEL
ADD
localhost:8080/nifi/?processGroupId=root&componentIds=a895f297-0178-1000-b767-4fee41022814
GetFile
Processor
895f297-0178-1000-b767-4fee41022814
In
Read/Write
Out
Tasks/Time

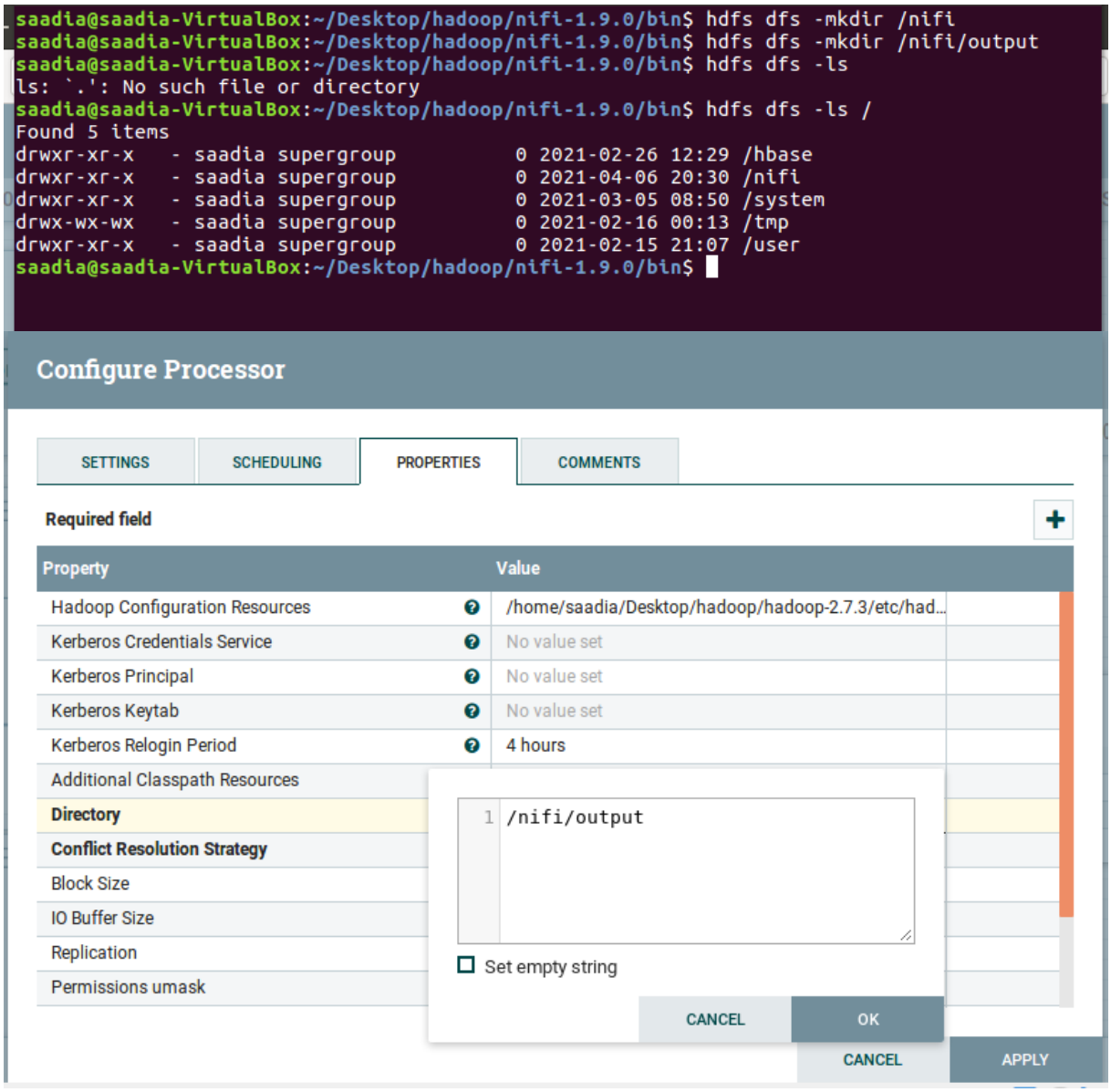

5 min
5 min
5 min
5 min
GetFile
GetFile 1.9.0
In 0 (0 bytes)
Read/Write 0 bytes / 0
Out 0 (0 bytes)
Tasks/Time 0 / 00:00:00
Configure
Disable
View data provenance
View status history
View usage
View connections
Center in view
Change color
Group
Create template
Copy
Delete
Configure Processor
SETTINGS
SCHEDULING
PROPERTIES
COMMENTS
Name
GetFile
Enabled
Automatically Terminate Relationships
success
All files are routed to success
Id
a895f297-0178-1000-b767-4fee41022814
Type
GetFile 1.9.0
Bundle
org.apache.nifi - nifi-standard-nar
Penalty Duration
30 sec
Yield Duration
1 sec
Bulletin Level
WARN
CANCEL
APPLY

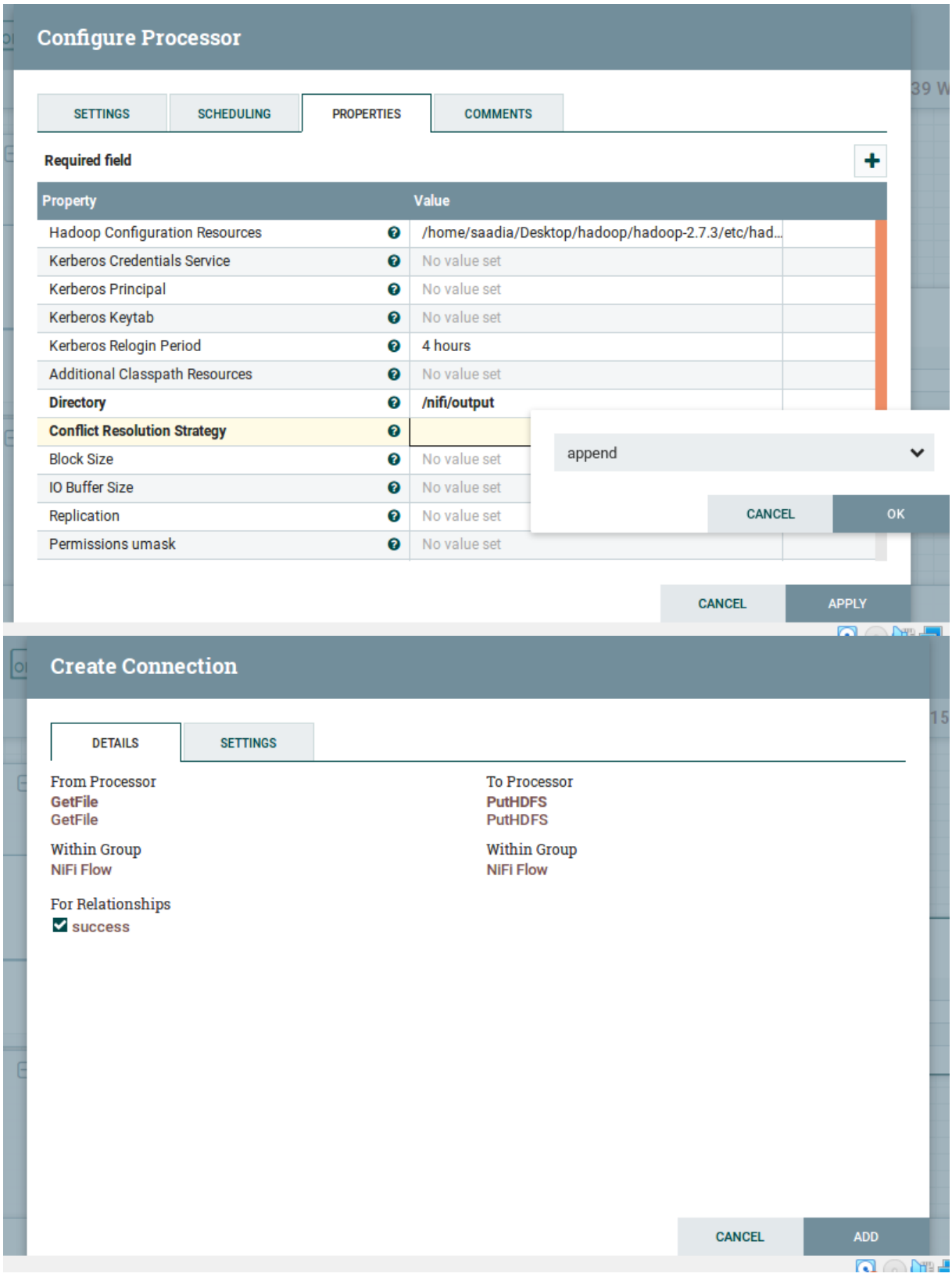

Configure Processor
SETTINGS
SCHEDULING
PROPERTIES
COMMENTS
Required field
Property
Value
Input Directory
File Filter
Path Filter
Batch Size
Keep Source File
Recurse Subdirectories
Polling Interval
Ignore Hidden Files
Minimum File Age
Maximum File Age
No value set
Minimum File Size
0 B
Maximum File Size
No value set
1 /home/saadia/Desktop/inputfile
Set empty string
CANCEL
OK
CANCEL
APPLY
Configure Processor
SETTINGS
SCHEDULING
PROPERTIES
COMMENTS
Required field
Property
Value
Input Directory
/home/saadia/Desktop/inputfile
File Filter
[^\.].*
Path Filter
No value set
Batch Size
10
Keep Source File
Recurse Subdirectories
Polling Interval
Ignore Hidden Files
Minimum File Age
Maximum File Age
Minimum File Size
0 B
Maximum File Size
No value set
true
CANCEL
OK
CANCEL
APPLY

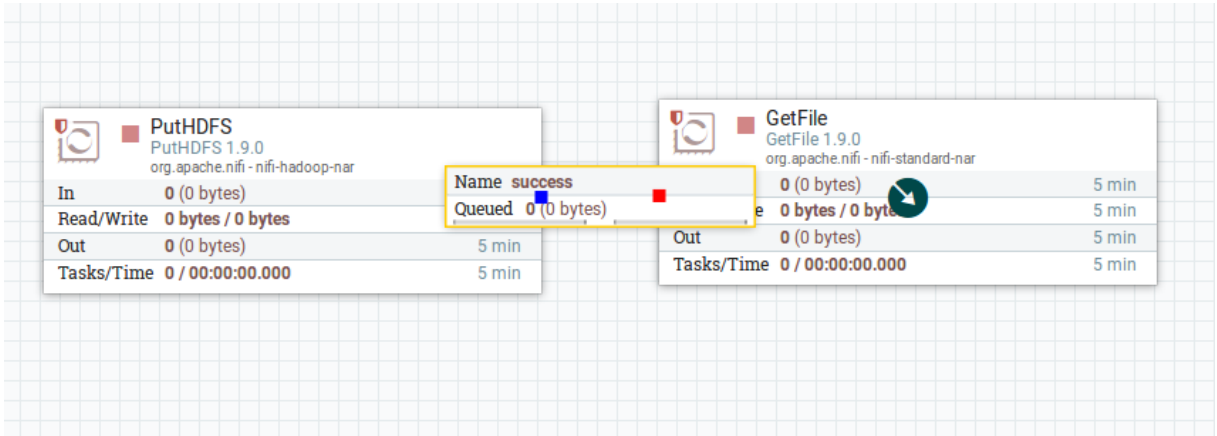

Add Processor
Source
all groups
amazon attributes avro aws consume csv delete fetch get hadoop ingest ingress insert json kafka listen logs message pubsub put record restricted source text update
Displaying 1 of 293
puthdfs
Type
Version
Tags
PutHDFS
1.9.0
restricted, HDFS, hadoop, copy, ...
PutHDFS 1.9.0
org.apache.nifi - nifi-hadoop-nar
Write FlowFile data to Hadoop Distributed File System (HDFS)
CANCEL
ADD
PutHDFS
PutHDFS 1.9.0
org.apache.nifi - nifi-hadoop-nar
In 0 (0 bytes) 5 min
Read/Write 0 bytes / 0 bytes 5 min
Out 0 (0 bytes) 5 min
Tasks/Time 0 / 00:00:00.000 5 min
GetFile
GetFile 1.9.0
org.apache.nifi - nifi-standard-nar
In 0 (0 bytes) 5 min
Read/Write 0 bytes / 0 bytes 5 min
Out 0 (0 bytes) 5 min
Tasks/Time 0 / 00:00:00.000 5 min


Configure Processor
SETTINGS
SCHEDULING
PROPERTIES
COMMENTS
Name
PutHDFS
Enabled
Id
a89d1d50-0178-1000-ee9e-3463897bf0c5
Type
PutHDFS 1.9.0
Bundle
org.apache.nifi - nifi-hadoop-nar
Penalty Duration
30 sec
Yield Duration
1 sec
Bulletin Level
WARN
Automatically Terminate Relationships
failure
Files that could not be written to HDFS for some reason are transferred to this relationship
success
Files that have been successfully written to HDFS are transferred to this relationship
CANCEL
APPLY


Configure Processor
SETTINGS
SCHEDULING
PROPERTIES
COMMENTS
Required field
Property
Hadoop Configuration Resources
Kerberos Credentials Service
Kerberos Principal
Kerberos Keytab
Kerberos Relogin Period
Additional Classpath Resources
Directory
Conflict Resolution Strategy
append
Block Size
No value set
IO Buffer Size
No value set
Replication
No value set
Permissions umask
No value set
1 :/hadoop/hdfs-site.xml ,/home/saadia
Set empty string
CANCEL
OK
CANCEL
APPLY

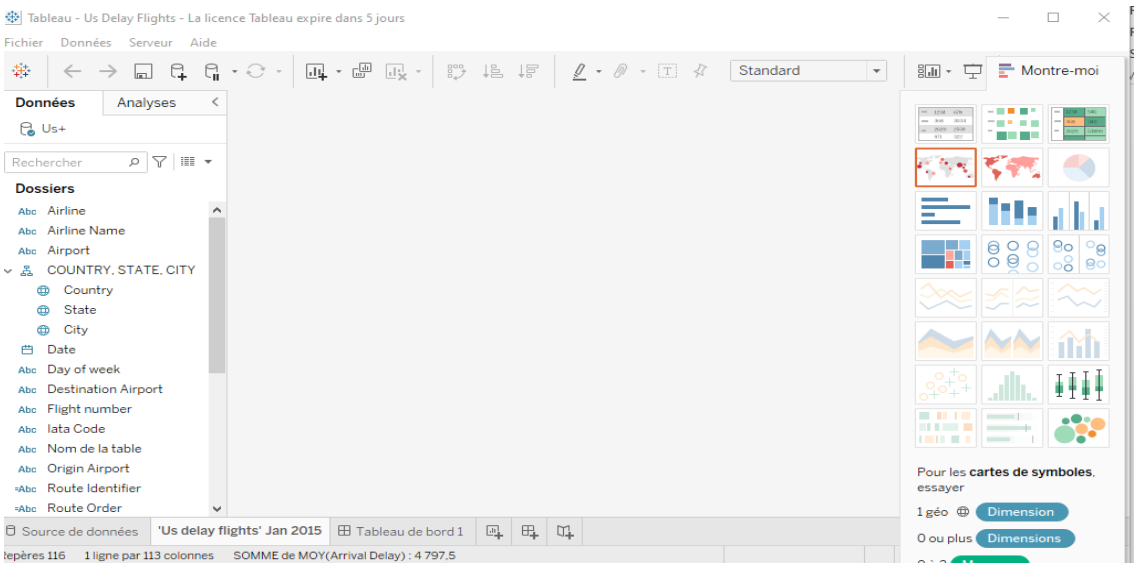

saadia@saadia-VirtualBox:~/Desktop/hadoop/nifi-1.9.0/bin$ hdfs dfs -mkdir /nifi
saadia@saadia-VirtualBox:~/Desktop/hadoop/nifi-1.9.0/bin$ hdfs dfs -mkdir /nifi/output
saadia@saadia-VirtualBox:~/Desktop/hadoop/nifi-1.9.0/bin$ hdfs dfs -ls
ls: `.': No such file or directory
saadia@saadia-VirtualBox:~/Desktop/hadoop/nifi-1.9.0/bin$ hdfs dfs -ls /
Found 5 items
drwxr-xr-x - saadia supergroup 0 2021-02-26 12:29 /hbase
drwxr-xr-x - saadia supergroup 0 2021-04-06 20:30 /nifi
drwxr-xr-x - saadia supergroup 0 2021-03-05 08:50 /system
drwx-wx-wx - saadia supergroup 0 2021-02-16 00:13 /tmp
drwxr-xr-x - saadia supergroup 0 2021-02-15 21:07 /user
saadia@saadia-VirtualBox:~/Desktop/hadoop/nifi-1.9.0/bin$
Configure Processor
SETTINGS
SCHEDULING
PROPERTIES
COMMENTS
Required field
Property
Value
Hadoop Configuration Resources
/home/saadia/Desktop/hadoop/hadoop-2.7.3/etc/had...
Kerberos Credentials Service
No value set
Kerberos Principal
No value set
Kerberos Keytab
No value set
Kerberos Relogin Period
4 hours
Additional Classpath Resources
Directory
Conflict Resolution Strategy
Block Size
IO Buffer Size
Replication
Permissions umask
1 /nifi/output
Set empty string
CANCEL
OK
CANCEL
APPLY


Configure Processor
SETTINGS
SCHEDULING
PROPERTIES
COMMENTS
Required field
Property
Value
Hadoop Configuration Resources
/home/saadia/Desktop/hadoop/hadoop-2.7.3/etc/had...
Kerberos Credentials Service
No value set
Kerberos Principal
No value set
Kerberos Keytab
No value set
Kerberos Relogin Period
4 hours
Additional Classpath Resources
No value set
Directory
/nifi/output
Conflict Resolution Strategy
Block Size
No value set
IO Buffer Size
No value set
Replication
No value set
Permissions umask
No value set
append
CANCEL
OK
CANCEL
APPLY
Create Connection
DETAILS
SETTINGS
From Processor
GetFile
GetFile
Within Group
NiFi Flow
For Relationships
success
To Processor
PutHDFS
PutHDFS
Within Group
NiFi Flow
CANCEL
ADD




```
saadia@saadia-VirtualBox:~$ ls Desktop/inputfile/
cleaned_new_york_data.csv
saadia@saadia-VirtualBox:~$ hdfs dfs -ls /nifi/output
saadia@saadia-VirtualBox:~$ ls Desktop/inputfile/
saadia@saadia-VirtualBox:~$ hdfs dfs -ls /nifi/output
Found 1 items
-rw-r--r--   1 saadia supergroup    4336523 2021-04-06 21:22 /nifi/output/cleaned_new_york
_data.csv
saadia@saadia-VirtualBox:~$
```

Tableau :



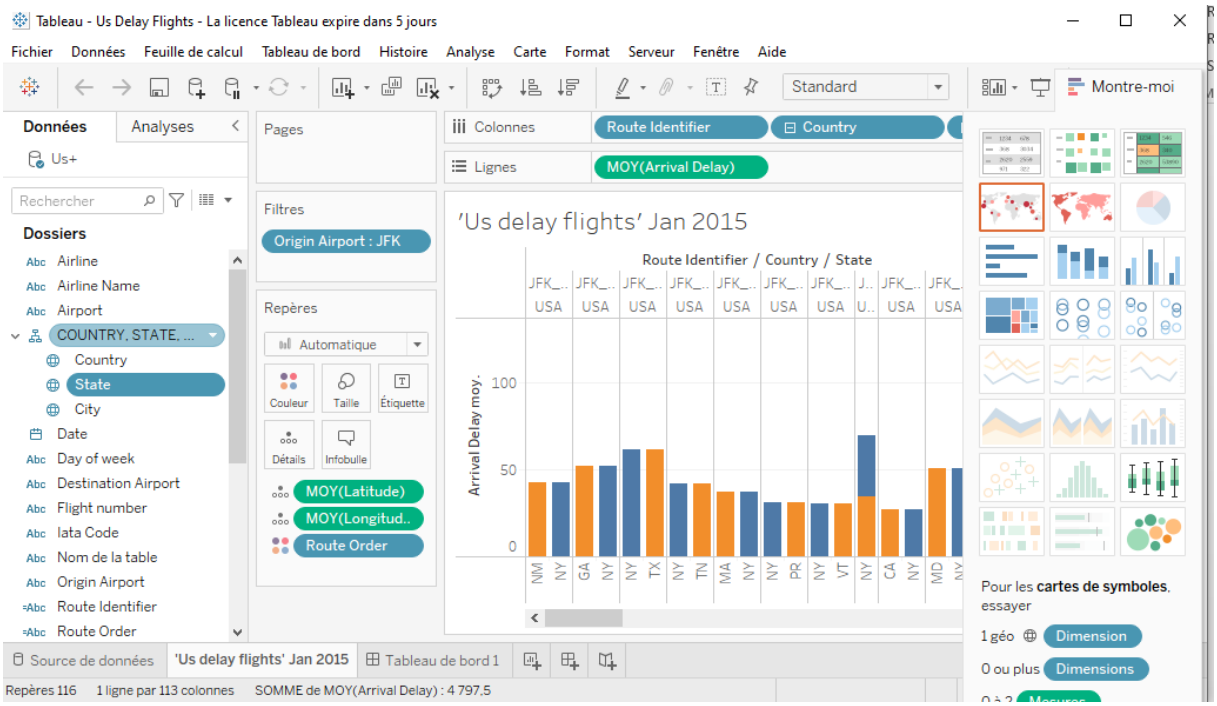
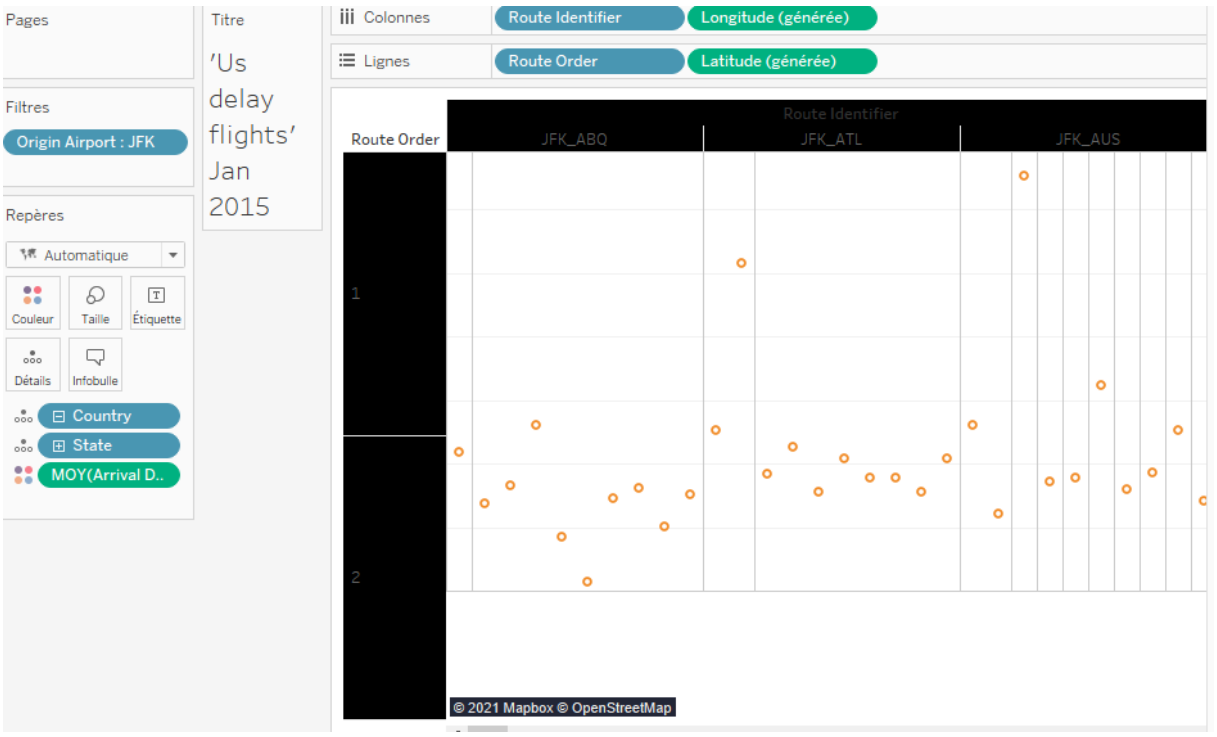


Tableau - Us Delay Flights - La licence Tableau expire dans 5 jours
Fichier Données Feuille de calcul Tableau de bord Histoire Analyse Carte Format Serveur Fenêtre Aide
Standard
Montre-moi
Données
Analyses
Us+
Rechercher
Dossiers
Abc Airline
Abc Airline Name
Abc Airport
COUNTRY, STATE, ...
Country
State
City
Date
Abc Day of week
Abc Destination Airport
Abc Flight number
Abc Iata Code
Abc Nom de la table
Abc Origin Airport
Abc Route Identifier
Abc Route Order
Pages
Filtres
Origin Airport : JFK
Repères
Automatique
Couleur
Taille
Étiquette
Détails
Infobulle
MOY(Latitude)
MOY(Longitud..
Route Order
Colonnes
Route Identifier
Country
Lignes
MOY(Arrival Delay)
'Us delay flights' Jan 2015
Route Identifier / Country / State
Arrival Delay moy.
100
50
0
USA
NM NY GA NY NY TX NY TN MA NY NY PR NY VT NY CA NY MD
Pour les cartes de symboles, essayer
1 géo Dimension
0 ou plus Dimensions
Source de données
'Us delay flights' Jan 2015
Tableau de bord 1
Repères 116
1 ligne par 113 colonnes
SOMME de MOY(Arrival Delay) : 4 797,5


Pages
Filtres
Origin Airport : JFK
Repères
Automatique
Couleur
Taille
Étiquette
Détails
Infobulle
Country
State
MOY(Arrival D..
Titre
'Us delay flights' Jan 2015
Colonnes
Route Identifier
Longitude (générée)
Lignes
Route Order
Latitude (générée)
Route Identifier
Route Order
JFK_ABQ
JFK_ATL
JFK_AUS
1
2
© 2021 Mapbox © OpenStreetMap

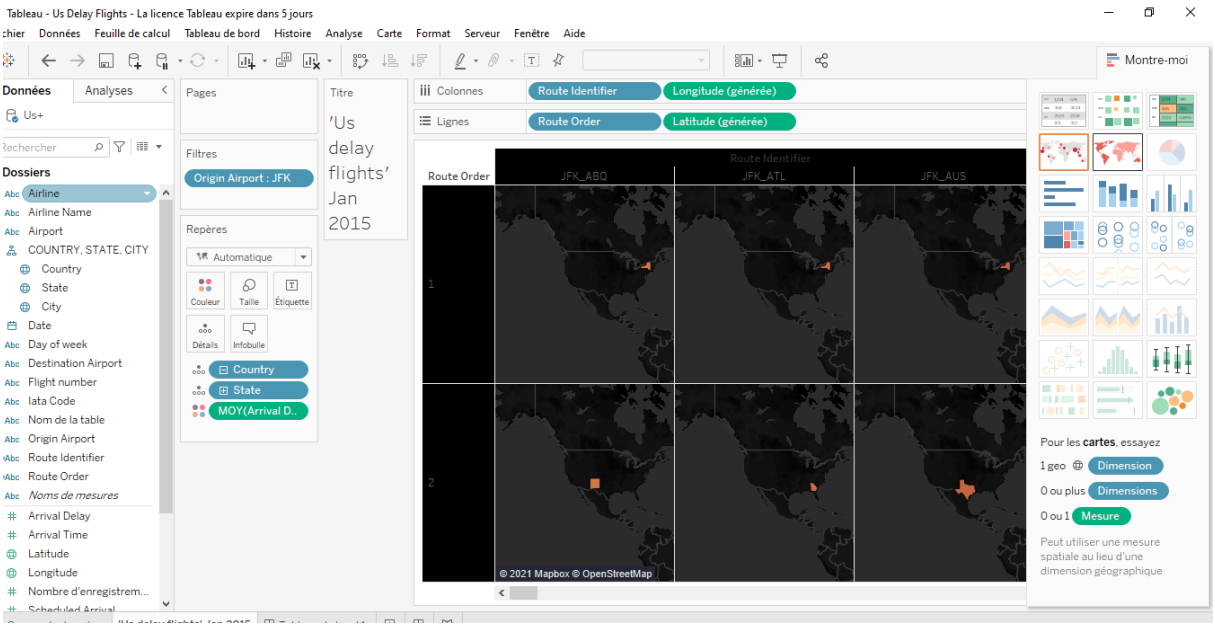


Tableau - Us Delay Flights - La licence Tableau expire dans 5 jours
Données
Analyses
Pages
Filtres
Origin Airport : JFK
Repères
Automatique
Couleur
Taille
Étiquette
Détails
Infobulle
Country
State
MOY(Arrival D..
Titre
'Us delay flights' Jan 2015
Colonnes
Route Identifier
Longitude (générée)
Lignes
Route Order
Latitude (générée)
Route Identifier
JFK_ABQ
JFK_ATL
JFK_AUS
© 2021 Mapbox © OpenStreetMap
Montre-moi
Pour les cartes, essayez
1 geo Dimension
0 ou plus Dimensions
0 ou 1 Mesure
Peut utiliser une mesure spatiale au lieu d'une dimension géographique