YOTABITES

Big Data Summit KC 17

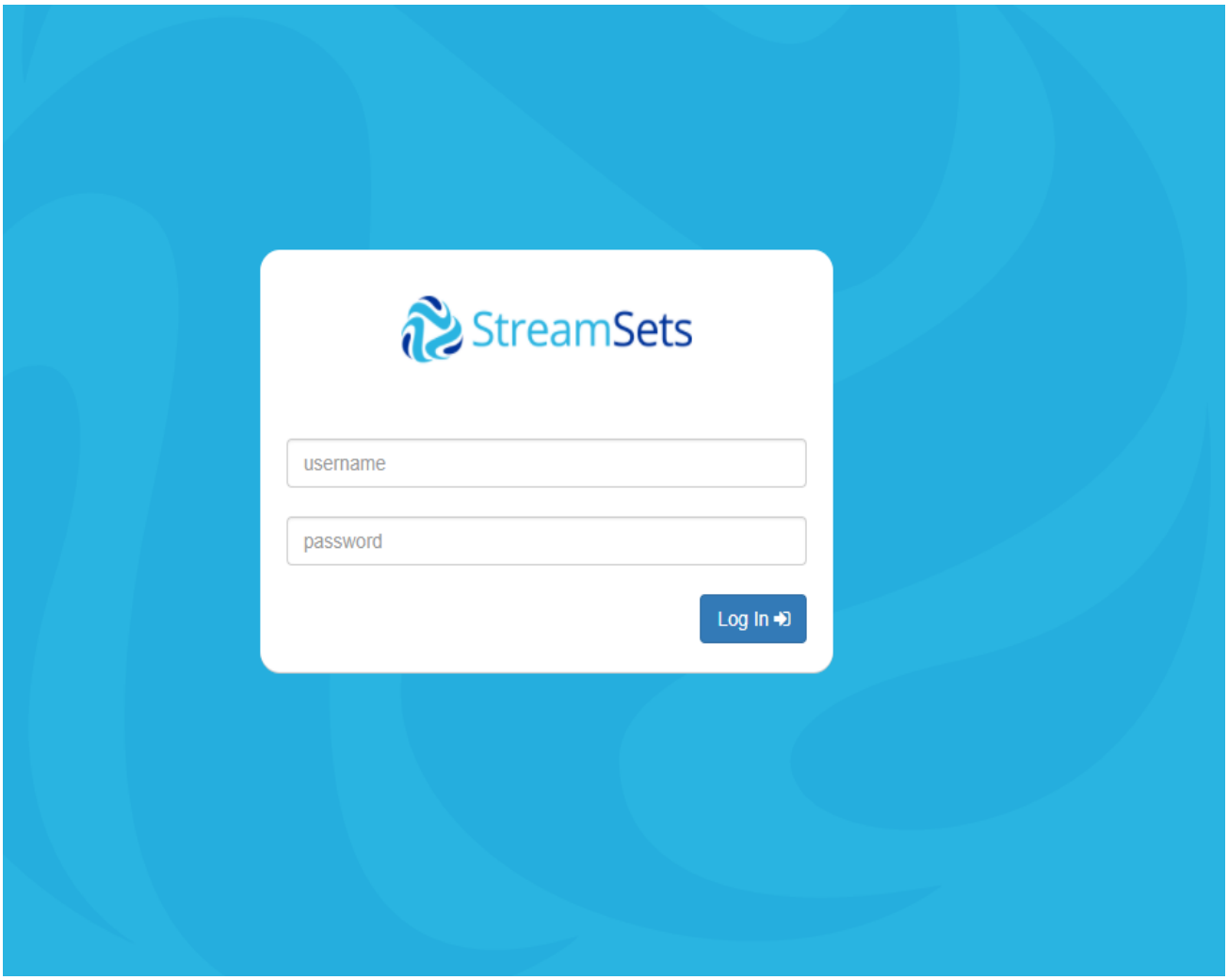
StreamSets WORKSHOP

Yotabites Consulting LLC | **888-441-8629** | **bigdata@yotabites.com**

BUILDING PIPELINES USING STREAMSETS

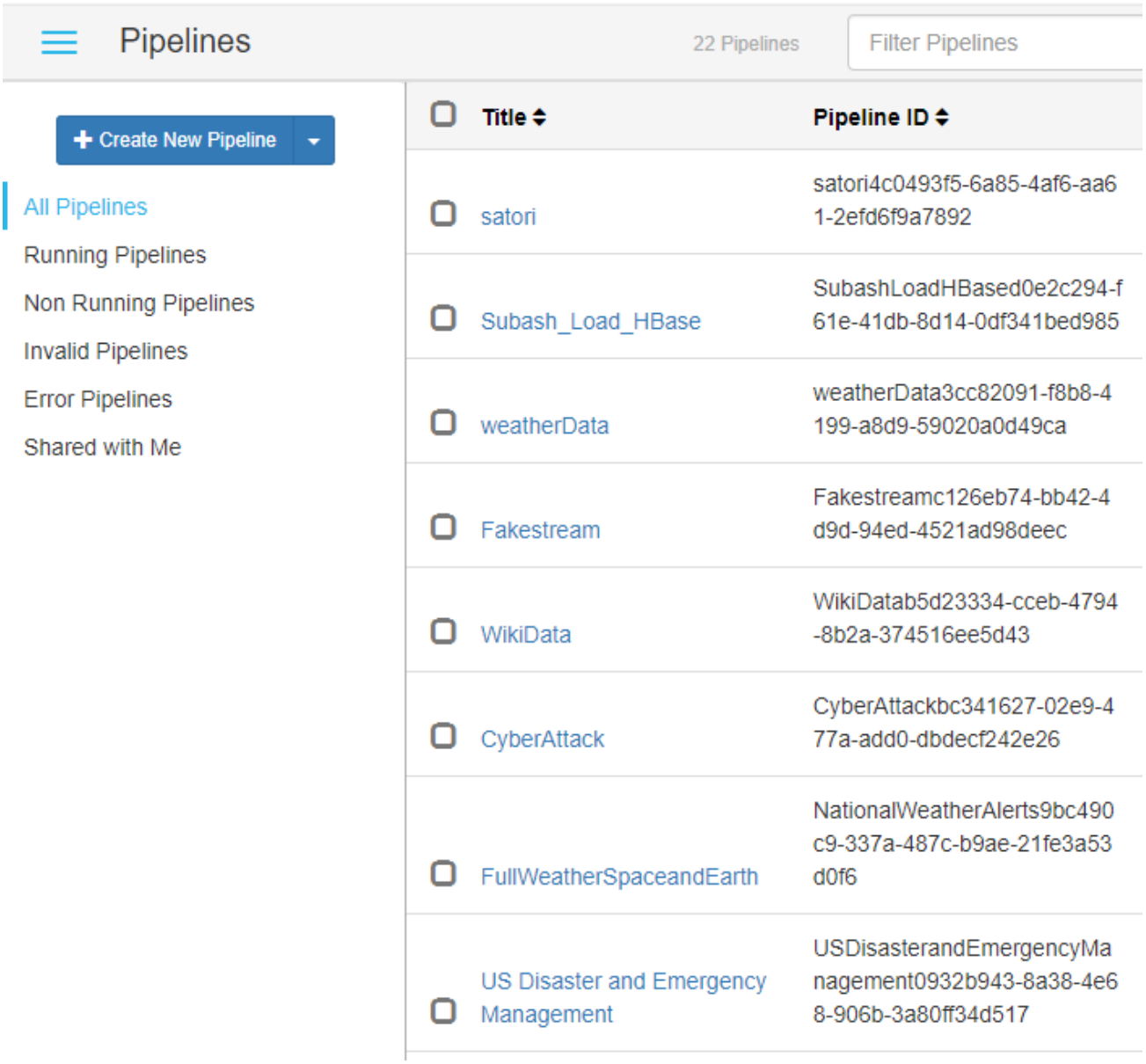
StreamSets – Web UI

1.1	Login to StreamSets.
1.1.1	Go to **http://ec2-34-203-42-12.compute-1.amazonaws.com:18630/** to access StreamSets Web UI in browser
1.1.2	Enter the **username** and **password** sent to your registered **Email id** and hit Log In.



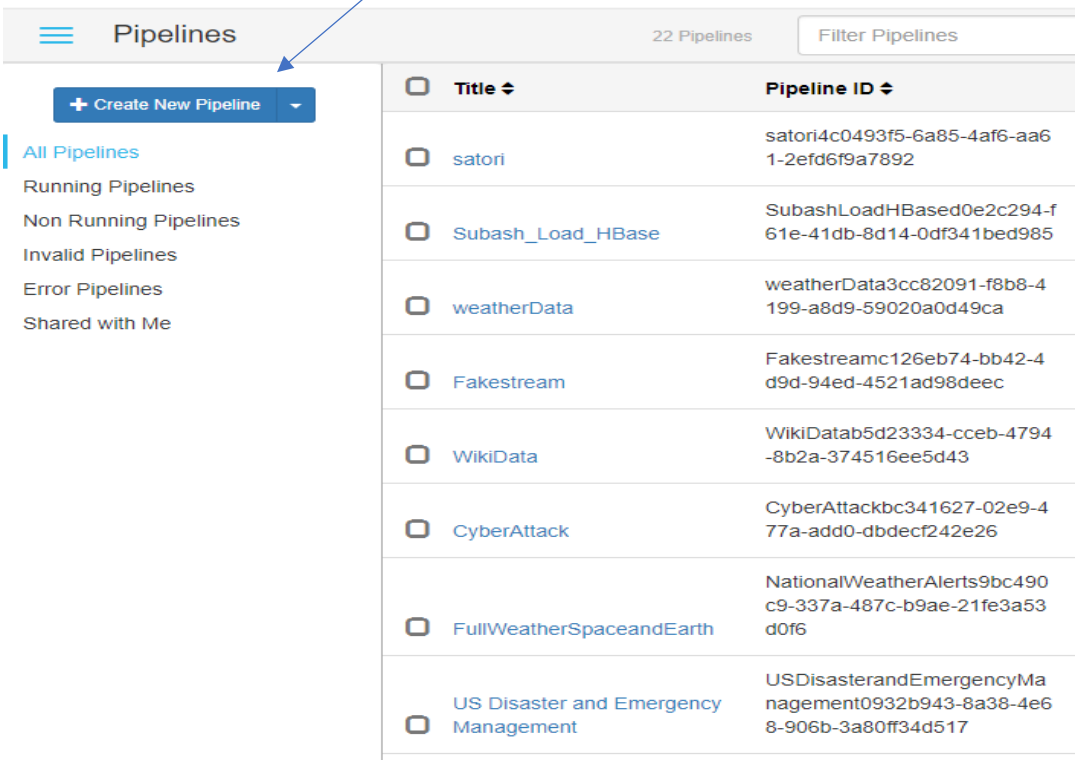
Yotabites

1.2	Once your login is successful, you will see the following landing page.

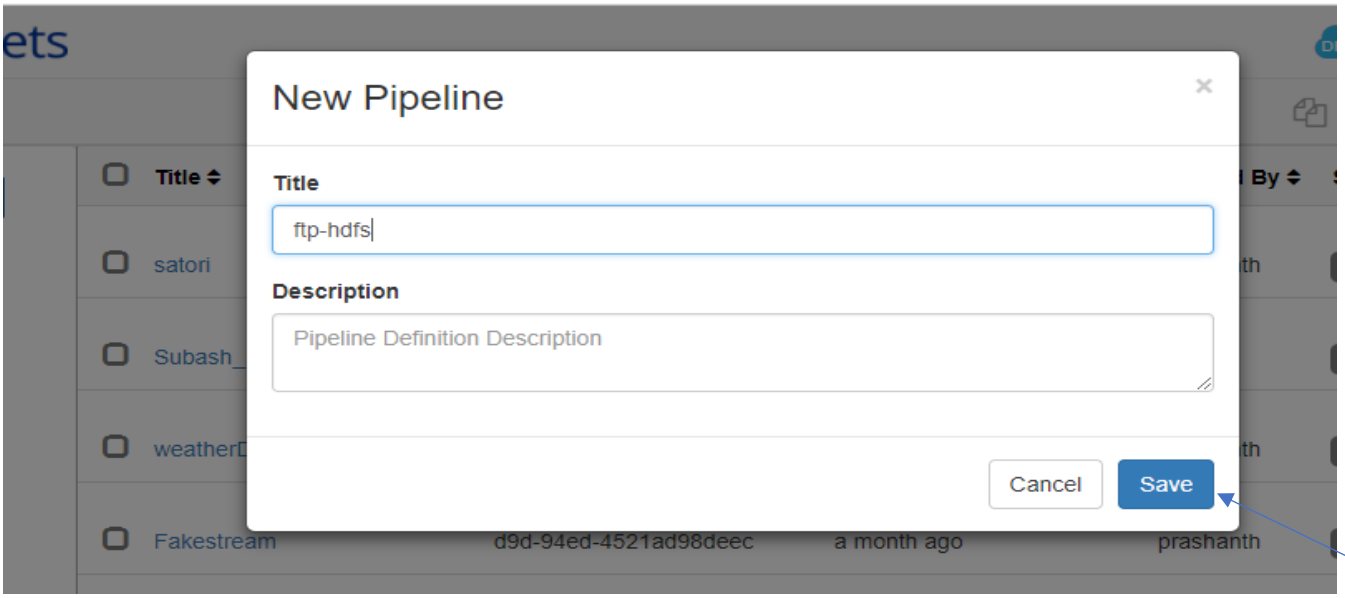


SFTP-HDFS

2.1	Click on **create new pipeline**

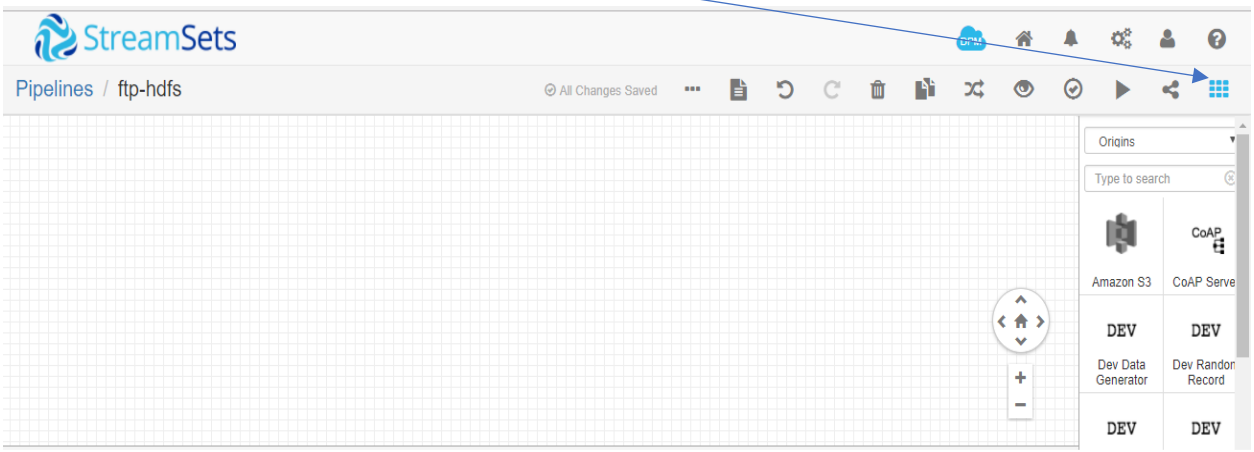


2.2	Input your **project title** and description in the **following window** and click save

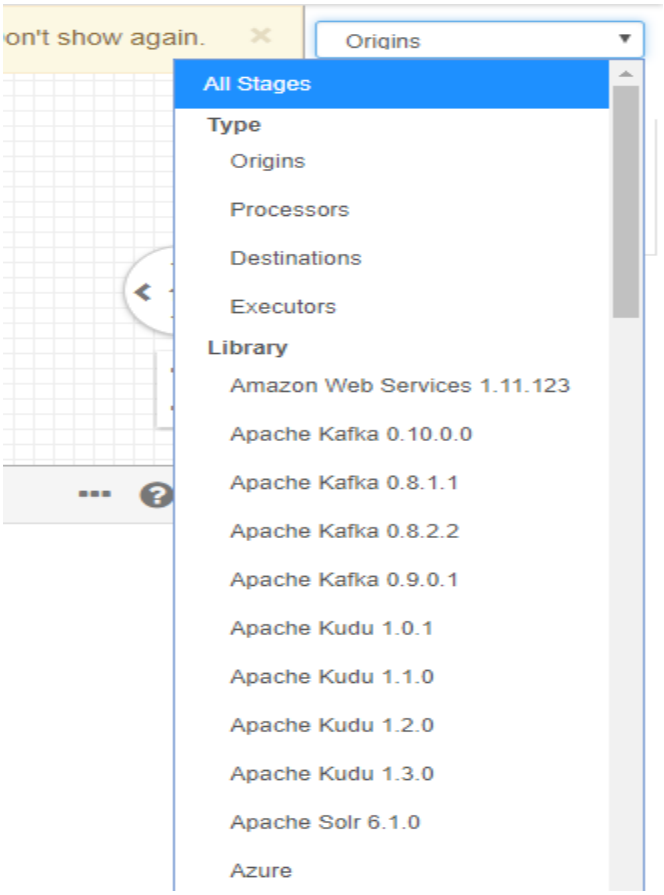


Yotabites

2.3	Now click on the **Stage library** option on the top right corner



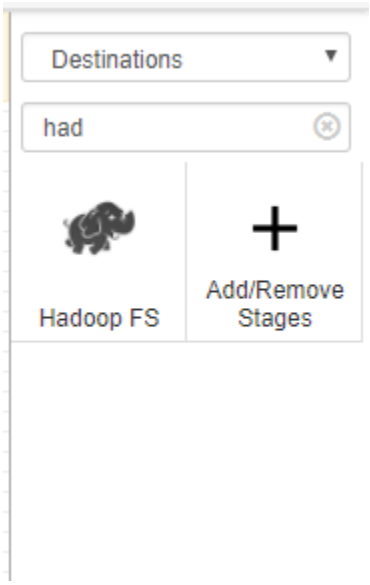
2.4	In the dropdown list select the **origins**



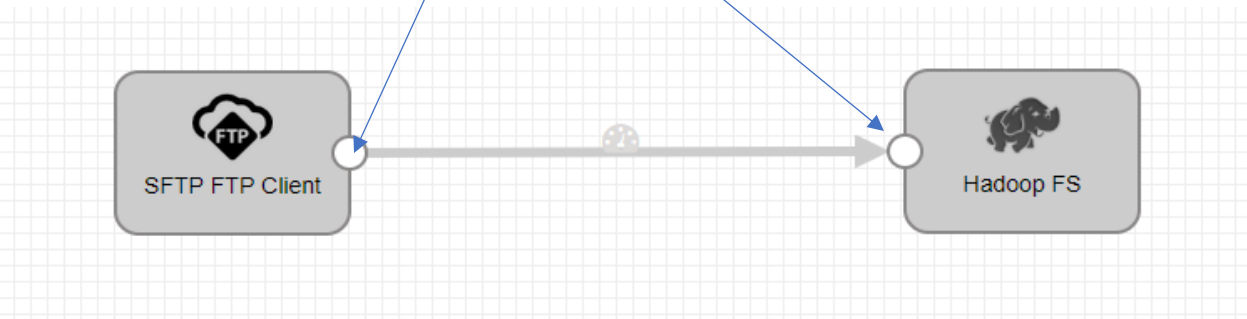
Yotabites

2.5	Now search for **SFTP FTP Client** and click once on the SFTP FTP Client option.

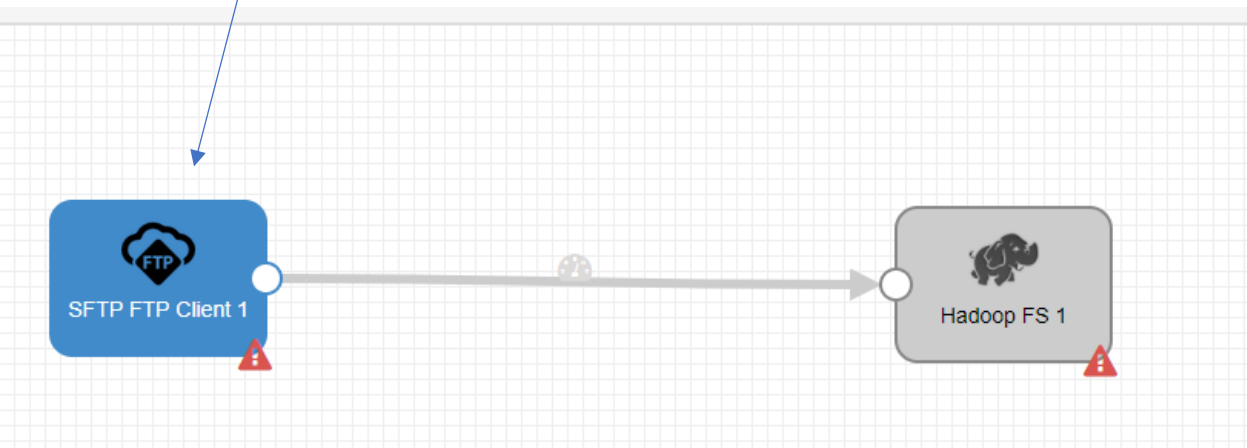
2.6	Now select **Destinations** in the dropdown list and select **Hadoop Fs** option and click once.



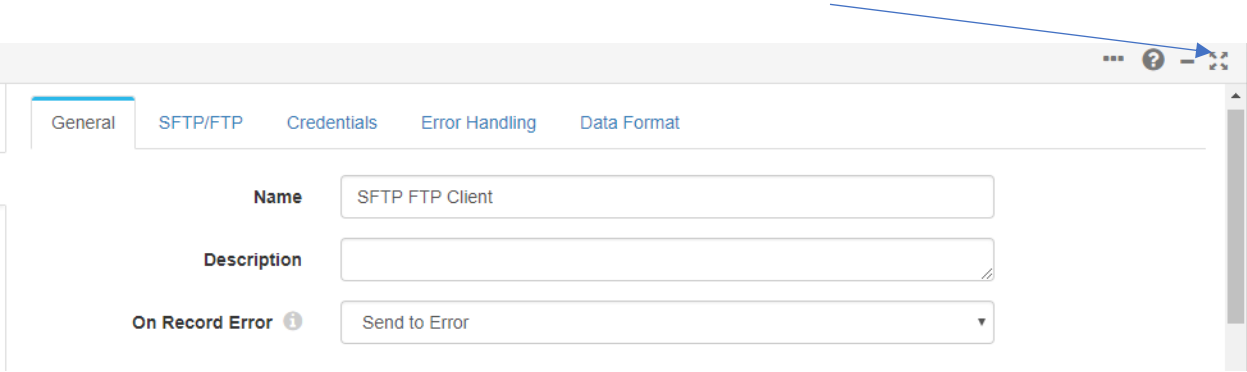
2.7	Now connect the **origin** with **destination** by left clicking on the circle on the origin and drag the arrow to connect to circle on the destination



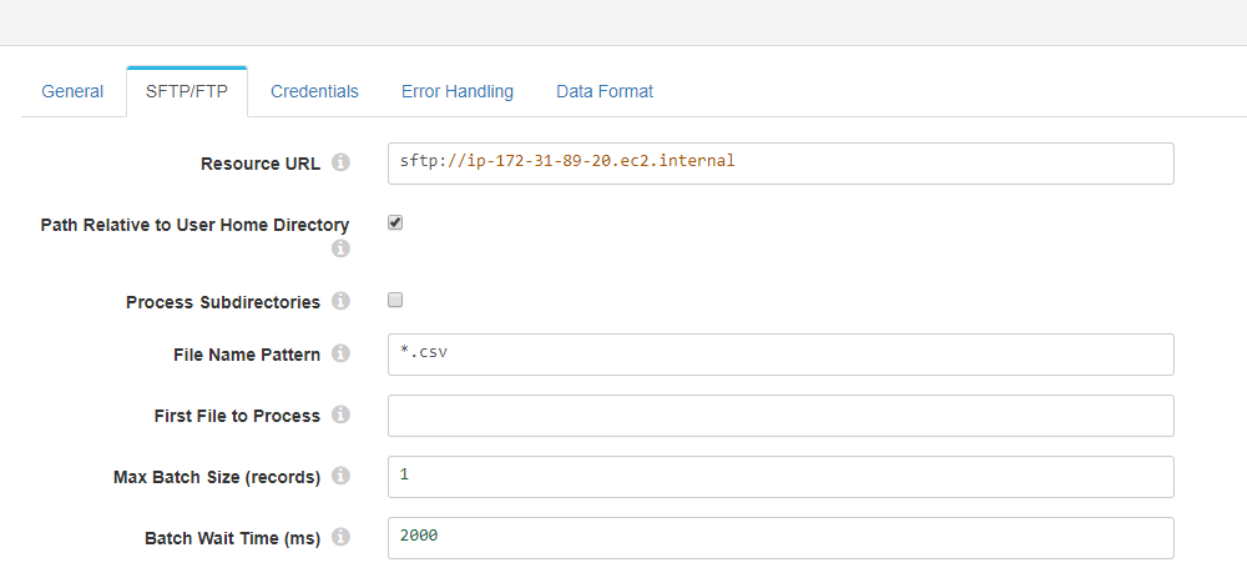
2.8	Now **click** on the **SFTP FTP** Client to provide the configuration to the origin



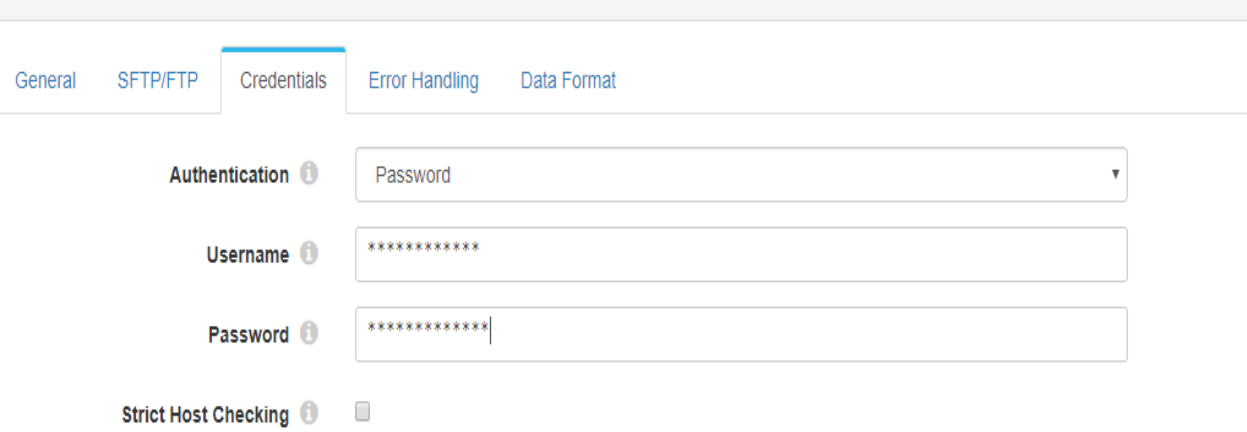
2.9	Enlarge the configuration window by clicking on the **enlarge** option



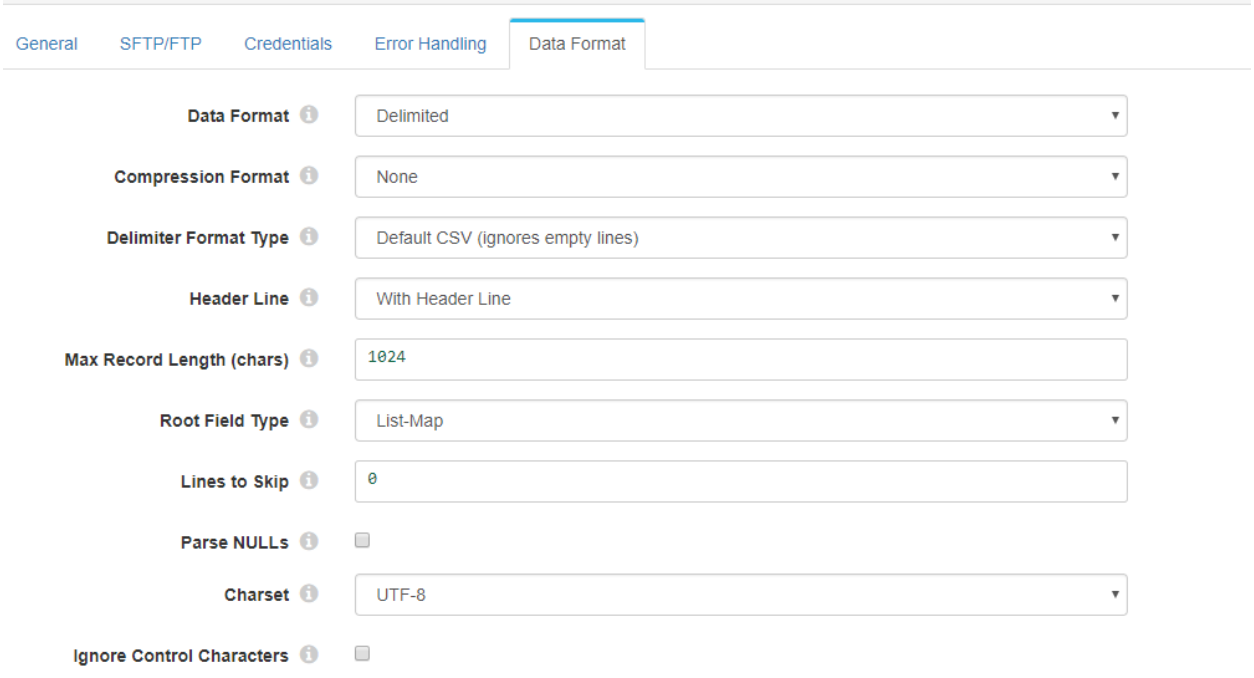
2.10	Now click on the **SFTP/FTP** option and fill the **Resource URL** and **File Name Pattern**



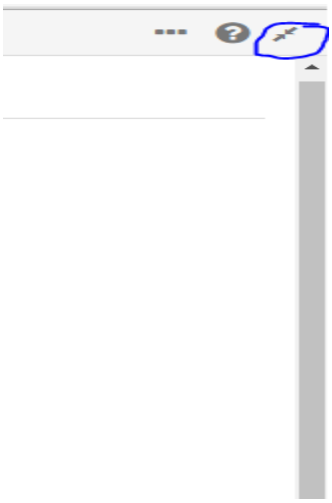
2.11	Select **Credentials** window. In the Authentication drop down list select **Password** option and enter your **username** and **password**



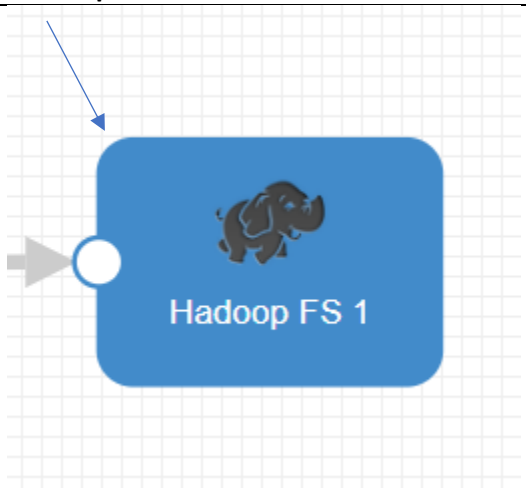
2.12	Now select the **Data Format** and input the configuration as specified in the below picture.



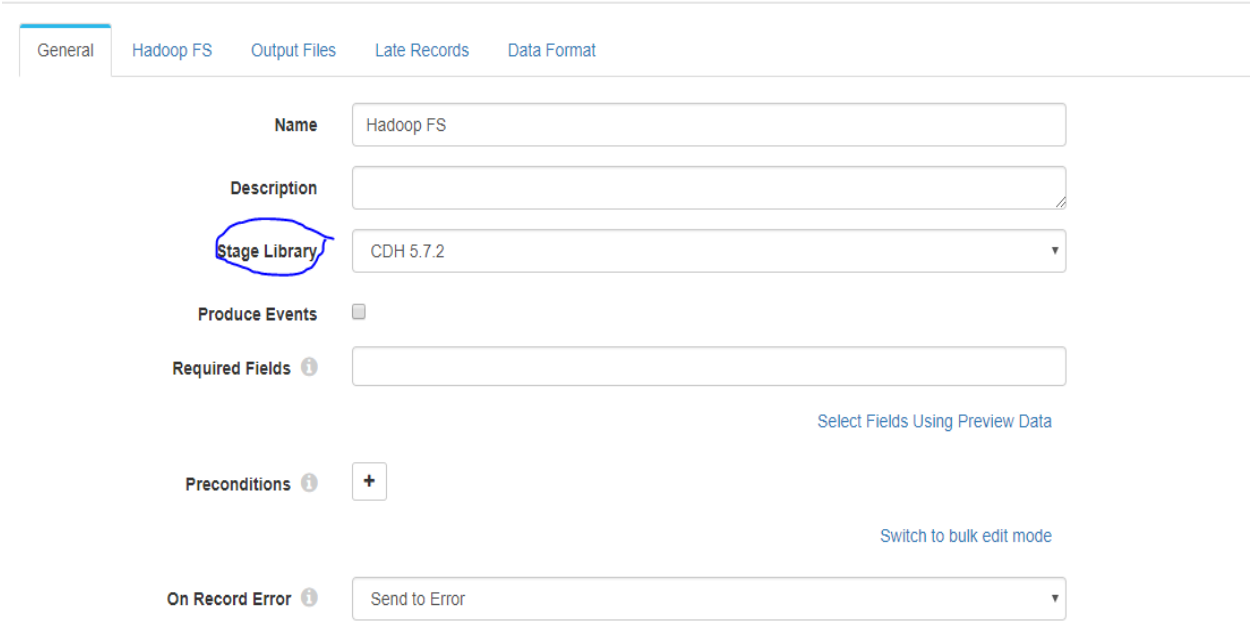
2.13	Now minimize the configuration window by clicking the **minimize** option on the top right of window



2.14	Now click on the **Hadoop FS** destination

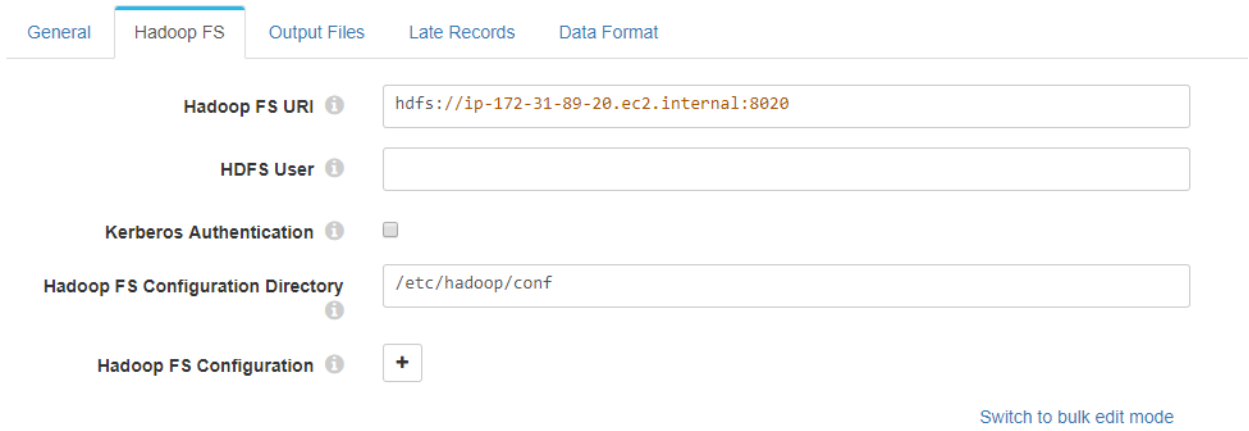


2.15	Click on General and Select the **Stage library** as per your CDH version

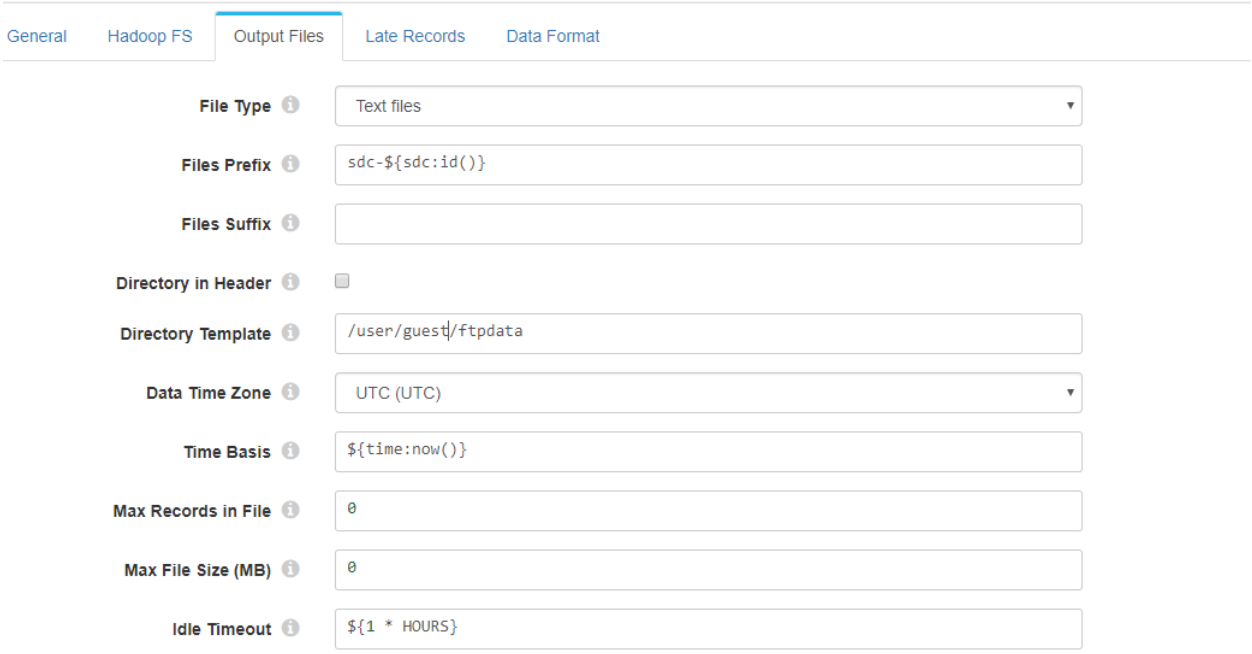


Yotabites

2.16	Now input the **URI to connect** to HDFS and Conf directory location(conf directory contain files like core-site.xml….etc)

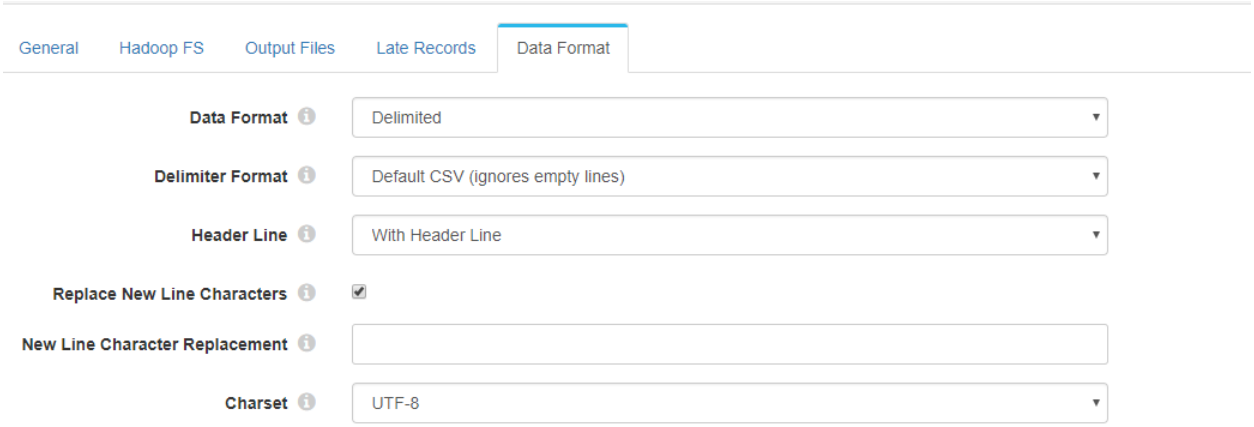


2.17	Now click on **output files** and input the path where you want yours records to be stored.

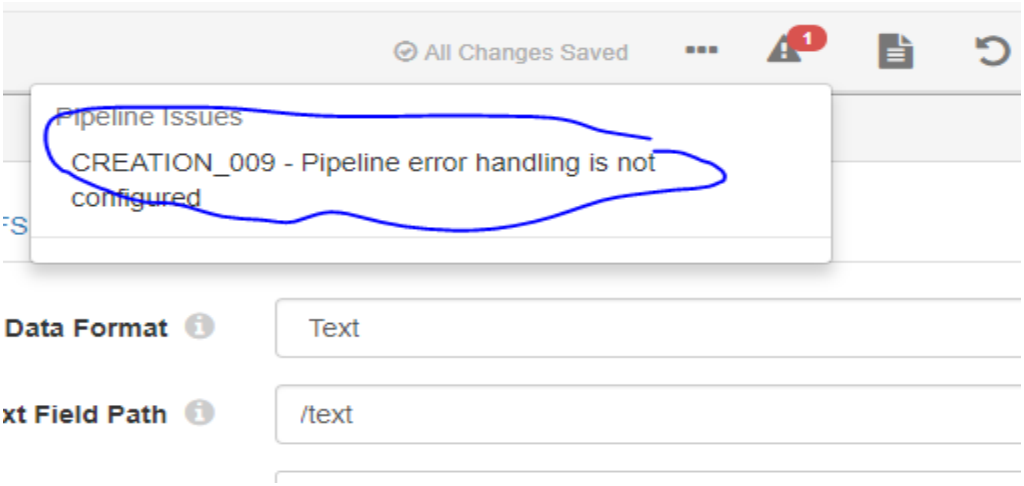


Yotabites

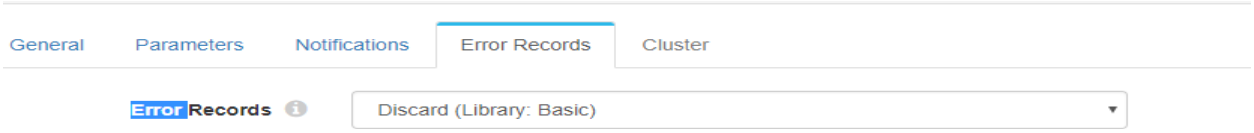
2.18	Now click on the **Data Format** and select the format of data coming from origin.In this pipeline we are using CSV.



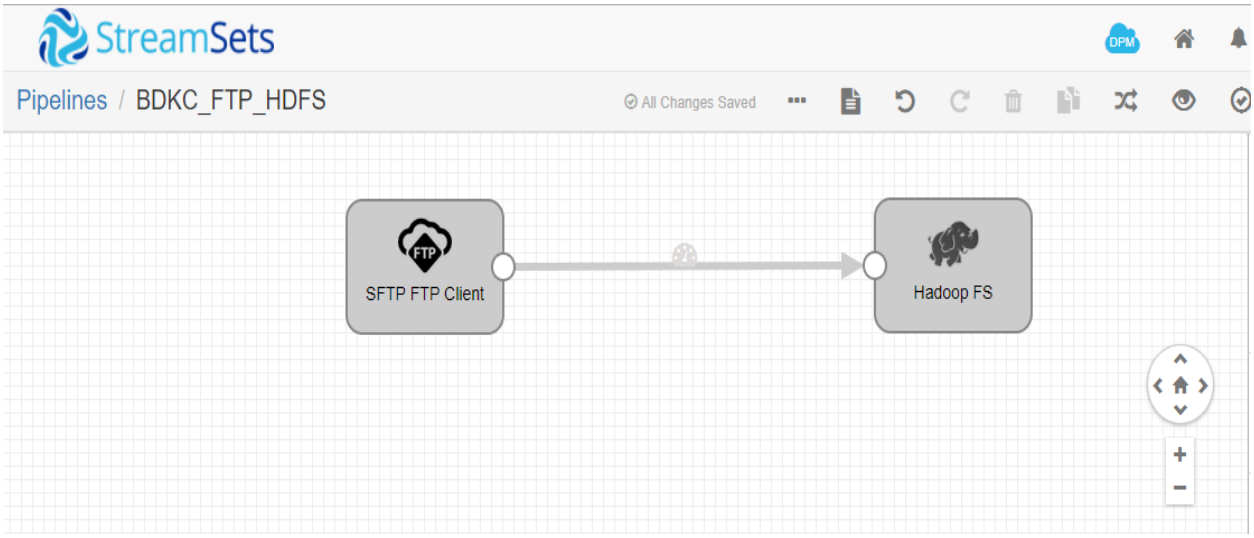
2.19	Now minimize configuration window and correct the following error by **clicking on the error**.



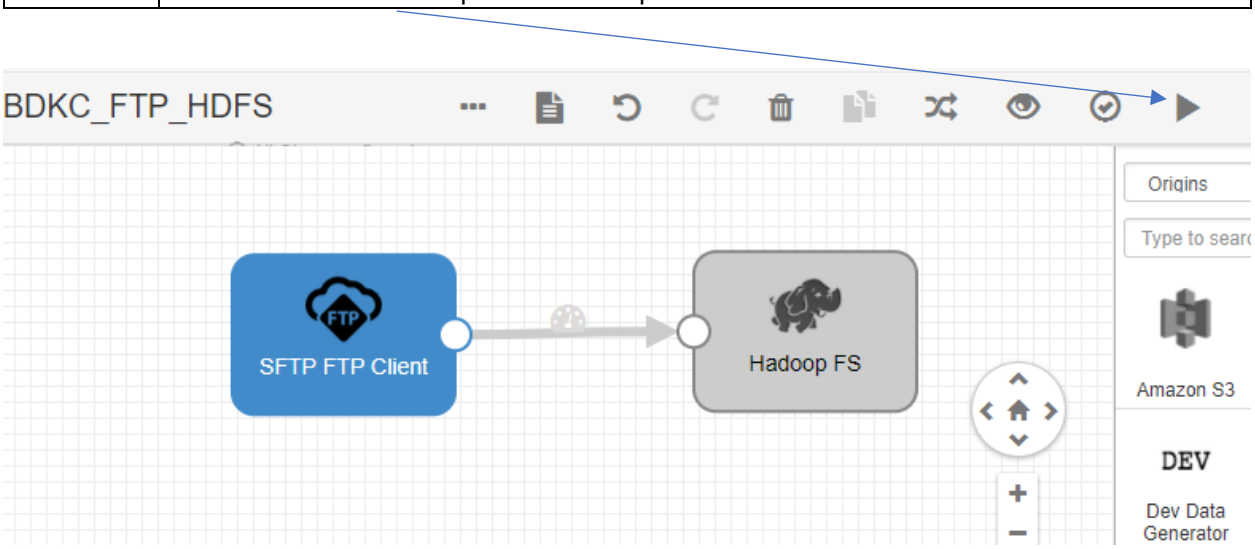
2.20	Select **Discard** option from the drop-down list.



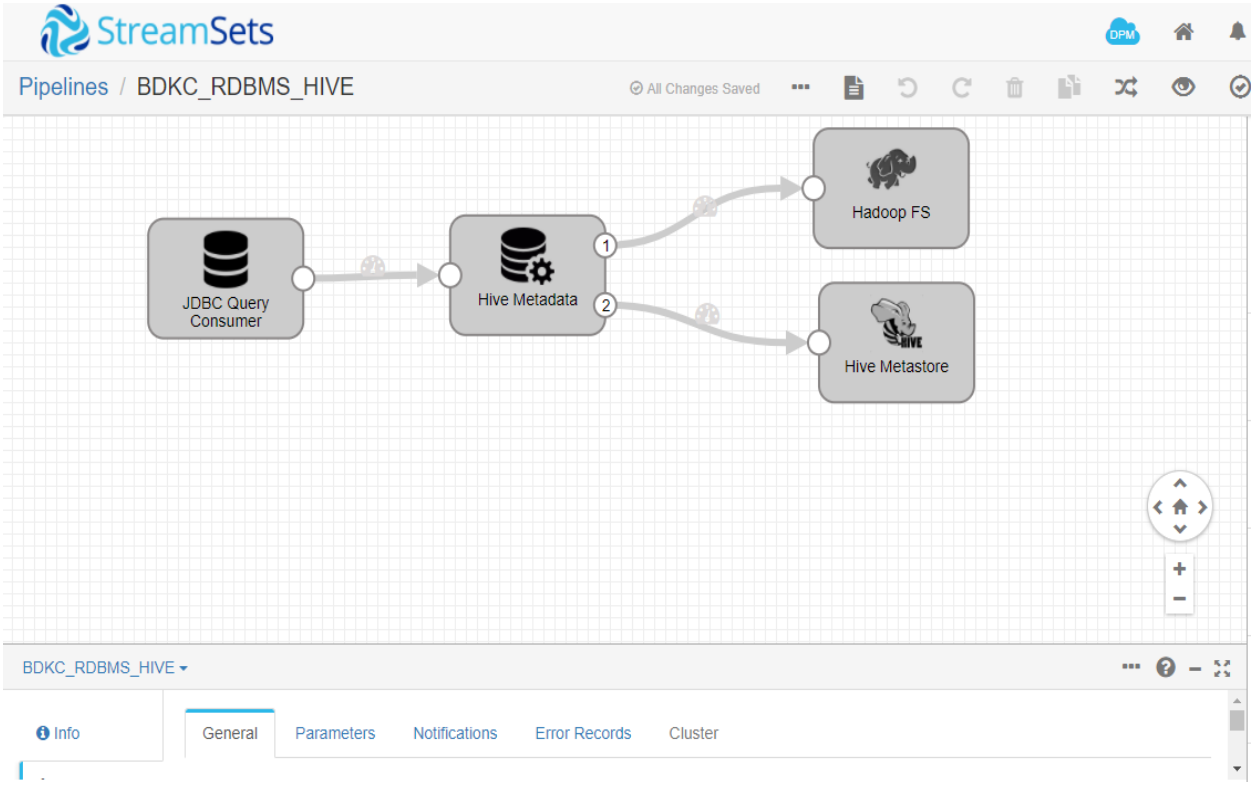
2.21	Once all the configurations are successfully validated. We will the following pipeline.



2.22	Now Click on the **start** option on the top left of the taskbar.



RDBMS-HIVE



Yotabites

	JDBC Consumer – Origin
Window Name	**Configuration**
JDBC	**JDBC Connection String** = *jdbc:mysql://ip-172-31-89-20.ec2.internal:3306/employee* **Incremental Mode** = *Check the box* **SQL Query** = *select * from tbldesignations where ID > ${OFFSET} ORDER BY ID* **Initial Offset** = 0 **Offset Column** = *ID (needs to be unique and keeps incrementing)* **Root Field Type** = *ListMap* **Note**: *Any key which is incrementing in an order can be taken to compare the offsets. We have used incrementing column as primary key for our convenience*.
Credentials	Username=*demo* Password=*demo*
Advanced	*Create JDBC Namespace Headers = Check the box*

	Hive Metadata – Processor
Window Name	**Configuration**
General	**Stage Library** = *CDH 5.9.2*
Hive	**JDBC URL** = *jdbc:hive2://ip-172-31-89-20.ec2.internal:10000/default* **JDBC Driver Name** = *org.apache.hive.jdbc.HiveDriver* **Hadoop Conf Directory** = */etc/hive/conf*

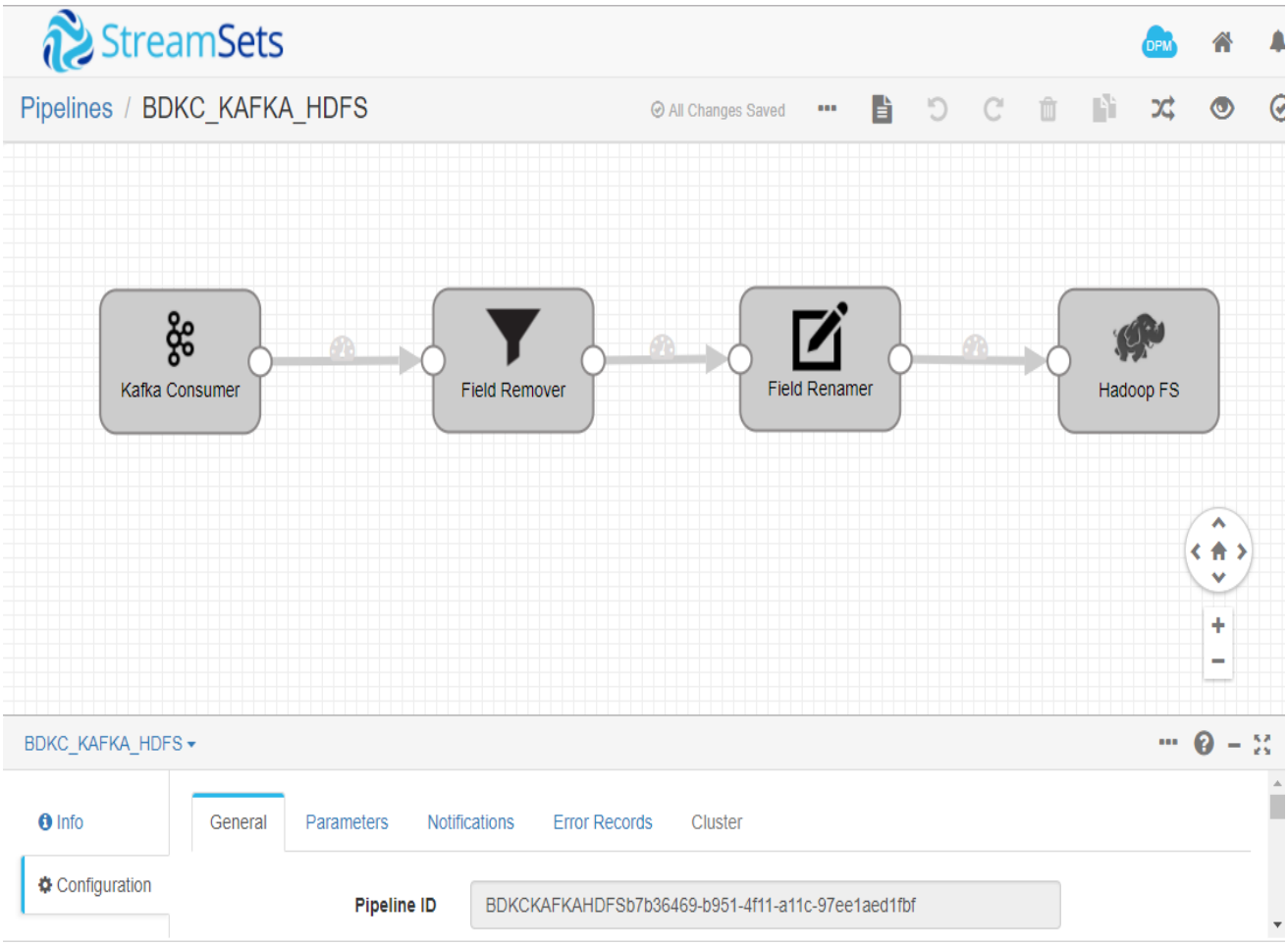
	Hive Metadata - Processor
Table	**Database Expression** = *${record:attribute('default')* **Table Name** = *${record:attribute('jdbc.tables')}*
Data Format	**Data Format** = *Avro*

	Hadoop FS – Destination
Window Name	**Configuration**
General	**Stage Library** = *Select the library with respect to CDH version*
Hadoop FS	**HadoopFS URI** = *hdfs:// ip-172-31-89-20.ec2.internal:8020* **HadoopFS Configuration Directory** = */etc/hadoop/conf*
Output Files	**File Type** = *Text Files* **Directory in Header** = *Check the box* **Max Records in File** = 1 **Use Roll Attribute** = *Check the box* **Roll Attribute Name** = *roll*

Data Format	**Data Format** = *Avro* **Avro Schema Location** = *In Record Header*

	Hive Metastore- Destination
Window Name	**Configuration**
Hive	**JDBC URL** = *jdbc: hive2:// ip-172-31-89-20.ec2.internal:10000/default* **JDBC Driver Name** = *org.apache.hive.jdbc.HiveDriver* **Hadoop Conf Directory** = */etc/hive/conf*
Advanced	**Stored as Avro** = *check the box*

KAFKA – HDFS



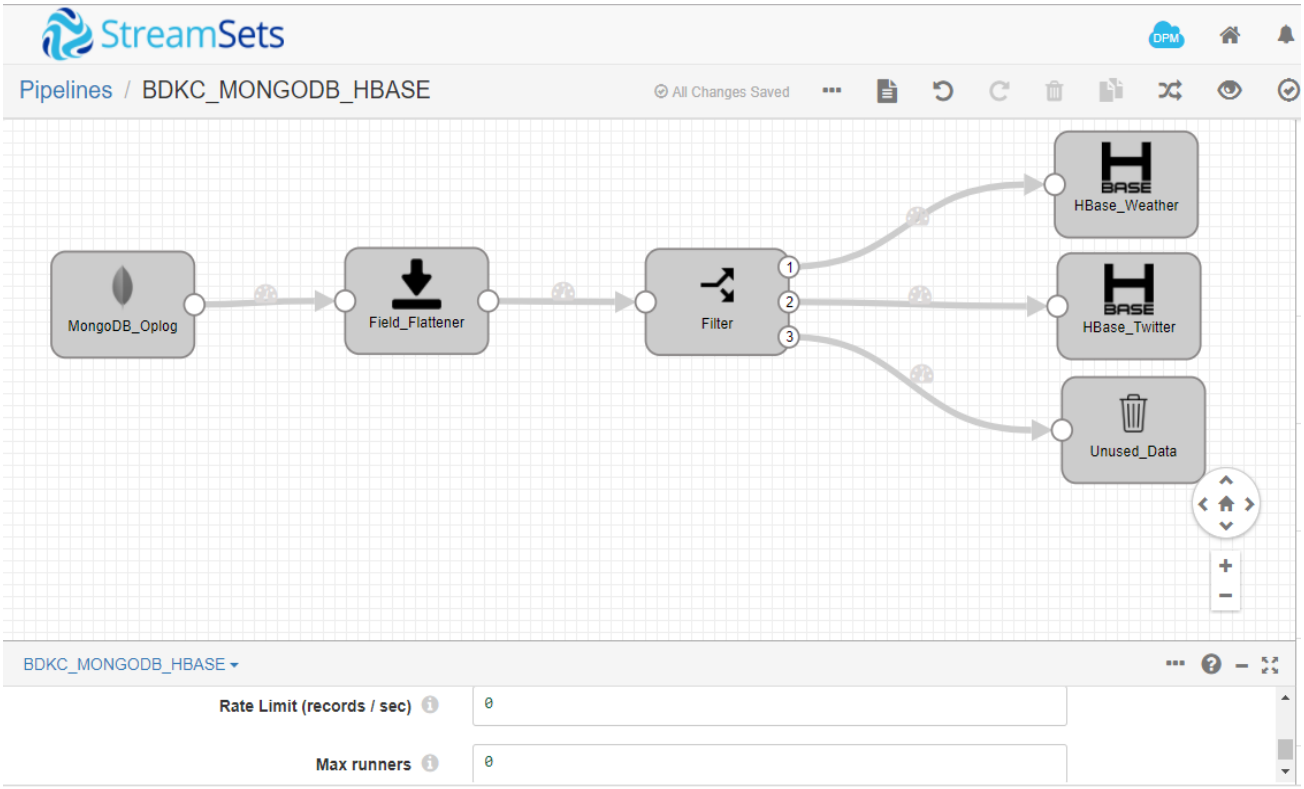
	Kafka Consumer – Origin
Window Name	**Configuration**
General	**Stage Library** = *Apache Kafka 0.10.0.0*
Kafka	**Broker URI** =*ip-172-31-89-20.ec2.internal:9092* **Zookeeper URI** = *ip-172-31-89-52.ec2.internal:2181/Kafka* **Topic** = *meetup*
Data Format	**Data Format** = *JSON* **Json Content** = *Multiple Json Objects*

	Field Remover – Processor
Window Name	**Configuration**
Remove/Keep	**Fields** = */member/photo ; /event/event_url*

	Field Renamer – Processor
Window Name	**Configuration**
Rename	**Source Field Expression** = */event/time* **Target Field Expression** = */event/mtime*

	HDFS – Destination
Window Name	**Configuration**
General	**Stage Library** = *CDH 5.9.2*
Hadoop FS	**HadoopFS URI** = *hdfs:// ip-172-31-89-20.ec2.internal:8020* **HadoopFS Configuration Directory** = */etc/hadoop/conf*
Data Format	**Data Format** = *Json* **Json Content** = *Multiple Json Object*

MongoDB – HBase



	MongoDB Oplog – Origin
Window Name	**Configuration**
MongoDB	**Connection String** = *mongodb://172.31.89.20:27017/streamset* **Collection** = *oplog.rs*
Credentials	**Authentication Type** = *None*

	Field Flattener – Processor
Window Name	**Configuration**
Flatten	**Flatten** = *Flatten entire record* **Name separator** = .

	Stream Selector – Processor
Window Name	**Configuration**
Conditions	**Condition** = *1 ${record:attribute('ns') =="streamset.weather"}* *2 ${record:attribute('ns') =="streamset.twitter"}* *3 default* Note: We have taken weather and twitter data. It can be replaced by any data streams. Use "+" on the right to add more conditions.

	HBase – Destination(Weather)
Window Name	**Configuration**
Hbase	**Zookeeper Quorum** = *172.31.89.52,172.31.81.155,172.31.87.163* **Zookeeper Client Port** = *2181* **Zookeeper Parent Znode** = */hbase* **Table Name** = *streamset_weather* **Rowkey** = */o.place* **Storage Type** =*Text* **Fields** = *Map the Field Path to Column with a proper syntax* Ex : Field path = /o.time_zone Column = Column_family:column_qualifier Storage = Text

	HBase – Destination(Twitter)
Window Name	**Configuration**
Hbase	**Zookeeper Quorum** = *172.31.89.52,172.31.81.155,172.31.87.163* **Zookeeper Client Port** = *2181* **Zookeeper Parent Znode** = */hbase* **Table Name** = *streamset_twitter* **Rowkey** = */o.screen_name* **Storage Type** =*Text* **Fields** = *Map the Field Path to Column with a proper syntax* Ex: Field path = /o. time_zone Column = Column_family: column_qualifier Storage = Text

	Trash- Destinations
Window Name	**Configuration**
General	**Name** = *Unused data* **Description** = *we send all the remaining data after filter to this pipeline*.