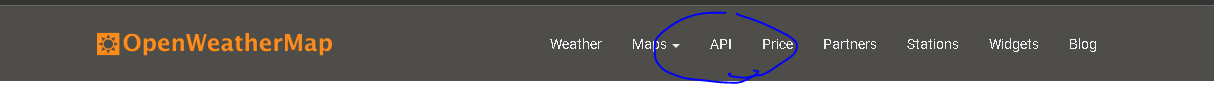
Objective: ETL weather data from openweather API

using NIFI

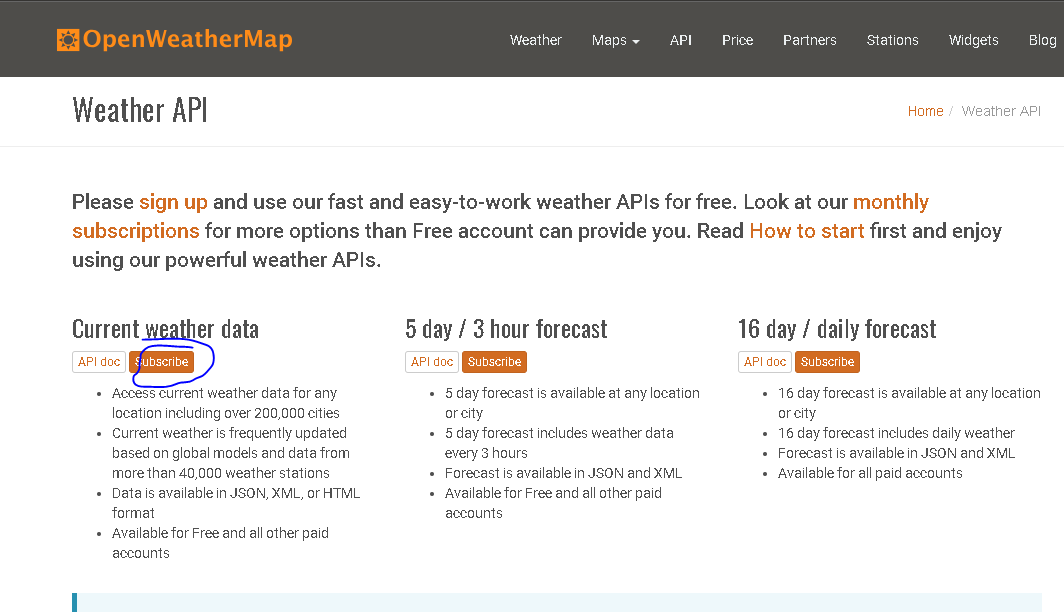
1. Obtain API key from <https://openweathermap.org>

Note: the layout of the site might change as time goes, therefore screenshot provided might not reflect current website’s layout.

1. Browse to [openweathermap.org](https://openweathermap.org) and find the API page.



Subscribe to the API



**2)** Follow the Sign up instruction and you will be provided with a API key that look like this

C47965fe3b09b7e80e5efe235406477o

Go back to the initial API page, click on the API doc to see how the API may be used.

For our purpose, we will do a simple example of obtaining current weather data of a area indicated by zip code.

The example of full api url is the following:

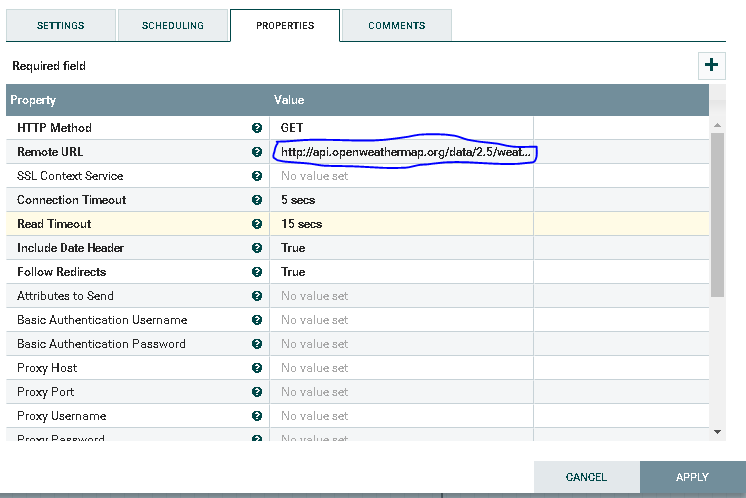
[http://api.openweathermap.org/data/2.5/weather?zip=10012,us&units=imperial&apikey={your](http://api.openweathermap.org/data/2.5/weather?zip=10012,us&units=imperial&apikey=%7Byour) api key}

Above url has three parameters - zip, unit and apikey. zip indicating zip code of the area you wish to fetch data from, it should be in format of {zip code},{country code}. The units of ‘imperial’ indicating that the temperature data is in Farenheit, Lastly, input your api key.

Below is a example of json file return by the API:;

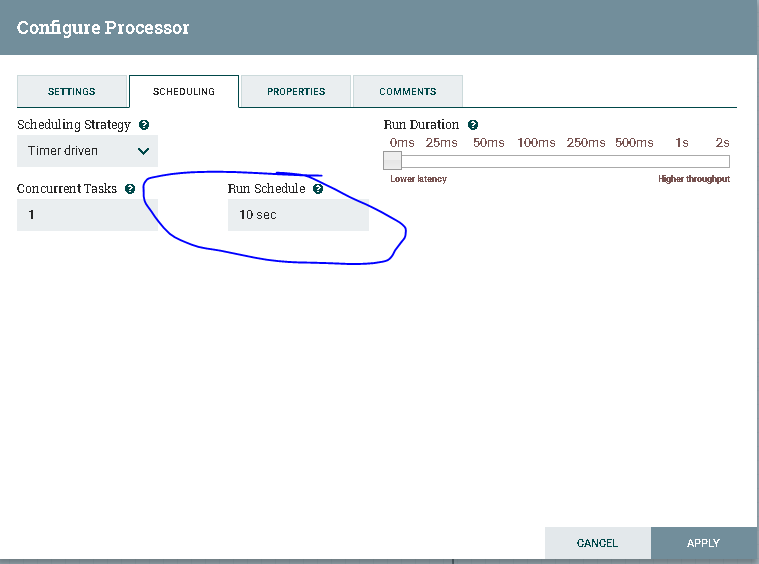
{"coord":{"lon":-73.99,"lat":40.73},"weather":[{"id":500,"main":"Rain","description":"light rain","icon":"10d"},{"id":701,"main":"Mist","description":"mist","icon":"50d"}],"base":"stations","main":{"temp":59.09,"pressure":1023,"humidity":96,"temp\_min":55.04,"temp\_max":64.4},"visibility":16093,"wind":{"speed":8.05,"deg":70},"rain":{"1h":0.51},"clouds":{"all":90},"dt":1536585840,"sys":{"type":1,"id":2121,"message":0.0048,"country":"US","sunrise":1536575538,"sunset":1536621144},"id":420027051,"name":"New York","cod":200}

II. Nifi

1. In the nifi canvas, add an ‘**InvokeHTTP**’ processor. Configure the properties of the processor 

Input your api url in the **Remote URL** property.

Go to **SCHEDULING**, set the **Run Schedule** to **10** second. It is to avoid excessive api call.

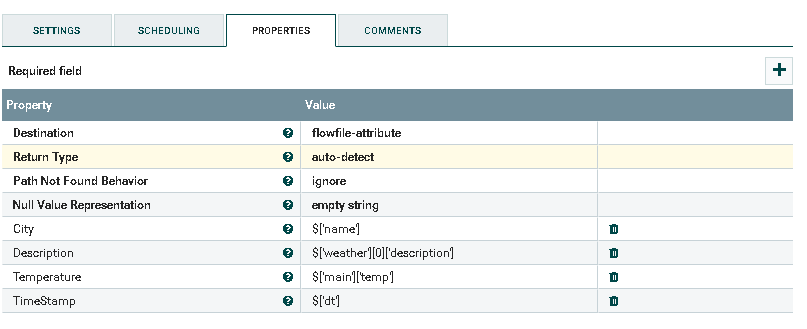


Click **APPLY** to finish configuring the **InvokeHTTP**.

Now the **InvokeHTTP** will return the whole json file.

2) Add another processor - **EvaluateJsonPath**,

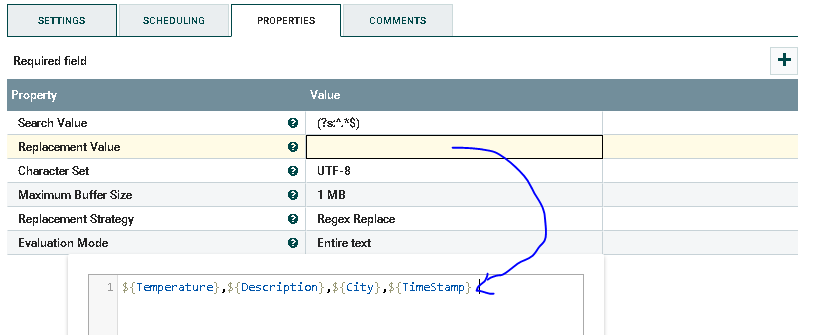
Configure the properties as follow;



The function of this processor is evaluate the content of the Flowfile using jsonpath and assign the result to Flowfile attribute.

3) Add a third processor - **ReplaceText.**

This processor allow you to access the attributes of the flowfile and write the value to as content. Change the properties as follow;

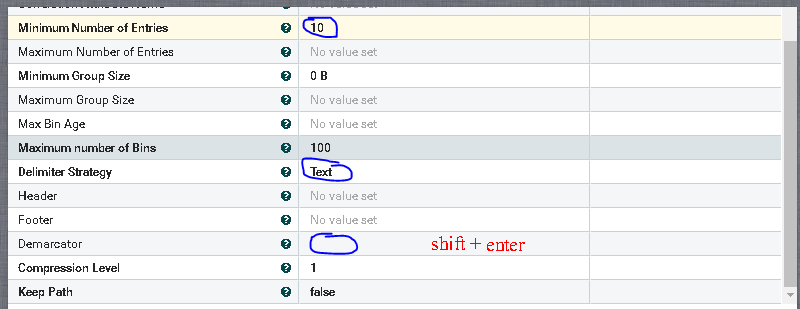


Note: ${attribute\_name} will return the attribute by the name of attribute\_name.

Everything in the content that match the regex in the ‘**Search Value**’ will be replaced by the string in the ‘**Replacement Value**’.

Now the content of the file will be a simple comma delimited file.

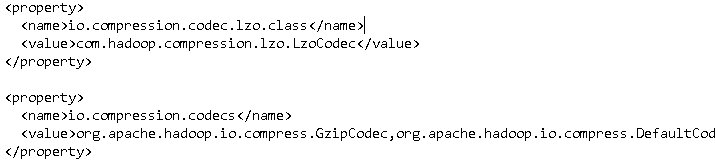
4) For each API call, our replace processor will produce one file. And the content of such file has only one comma delimited line. If we load the files directly to HDFS, we will have excessive numbers of small file. To avoid this, add **MergeContent** processor, and configure it as following;



**Minimum Number of Entries** tell the processor to wait for certain number of input files and merge them as one. In the **Demarcator**, enter new line by press shift + enter such that each entries of the file is a new line.

5) Lastly add **PutHDFS** processor. Before this processor will works, one must perform the following configuration step:

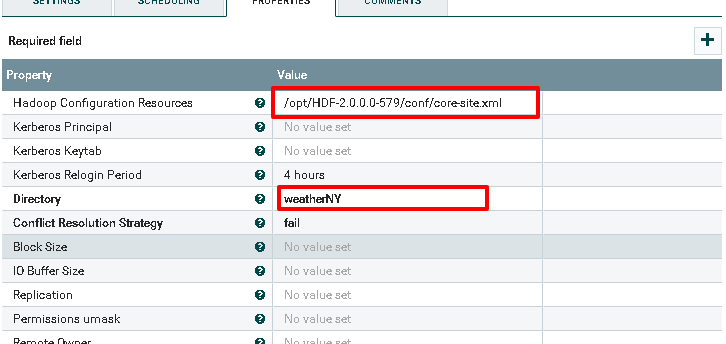
1. Using WinSCP, find **core-site.xml** in local(linux) machine. It is likely located at /etc/hadoop/conf/
2. Make a copy of this xml file to the /opt/HDF-2.0.0.0-579/conf/ or any nifi owned path. Open XML file at this location, and find below two properties;



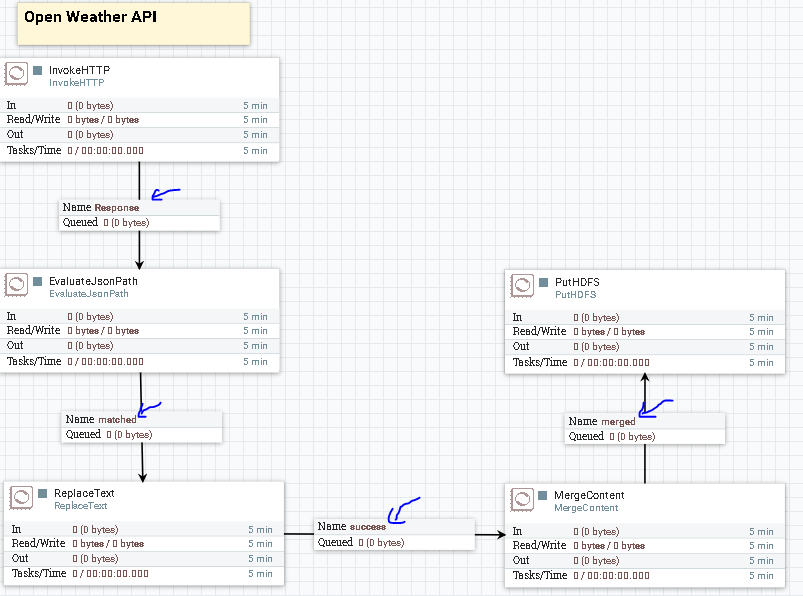
Delete these two properties and save;

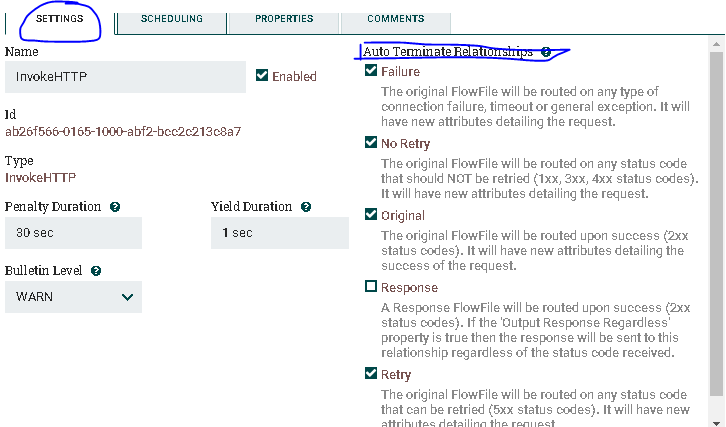
Copy the full path of this file.

   c) In **PutHDFS** processor, input the path of your xml file, and name an output directory;



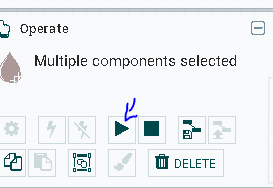
6) Now, connect all processor by relationship indicated as follows;



Check all other relationship under Auto Terminate Relationships in the configuration 🡪 setting: 

Do this for each processor.

Select all processor by pressing ctrl+A, and click run.

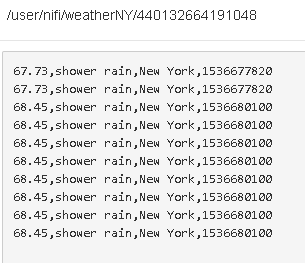


You should see **InvokeHttp** output a file at your specified rate. And **mergeContent** will accumulate 10 file before it send out to **PutHDFS** as one file.

Stop all processor, when you see some byte is written out by **PutHDFS.**

In hdfs: /user/nifi , you should see a new directory ‘weatherNY’ created.

Open any file inside, you should have;



Most of the data are same because the free API from openweather.org actually only update every 30 min.

1. Publish data to Kafka topic

Kafka is often integrated in NIFI to publish the message to a topic which then can be consumed by other tool such as storm.

Port 2181 ZooKeeper

Port 6667 Kafka Broker

Your port might be different. Confirm by going to Ambari UI dashboard, click on kafka icon and double cheCk the Configs.

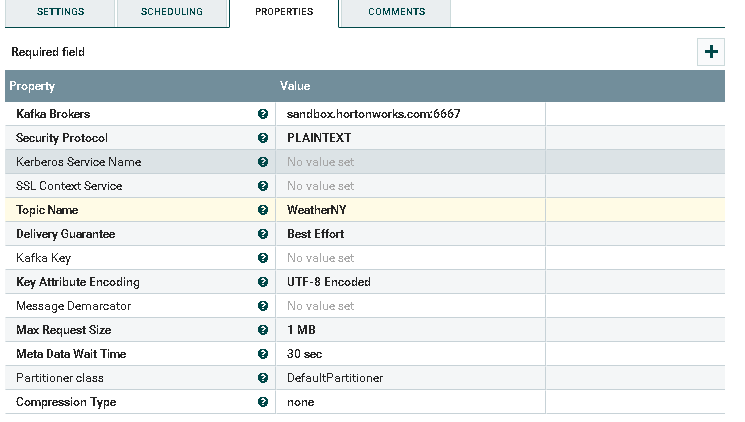
* + 1. Start the kafka server from ambari UI.
    2. From your command line interface run the following command to create a kafka topic;

/usr/hdp/current/kafka-broker/bin/kafka-topics.sh --create --zookeeper localhost:2181 --replication-factor 1 --partitions 1 --topic WeatherNY

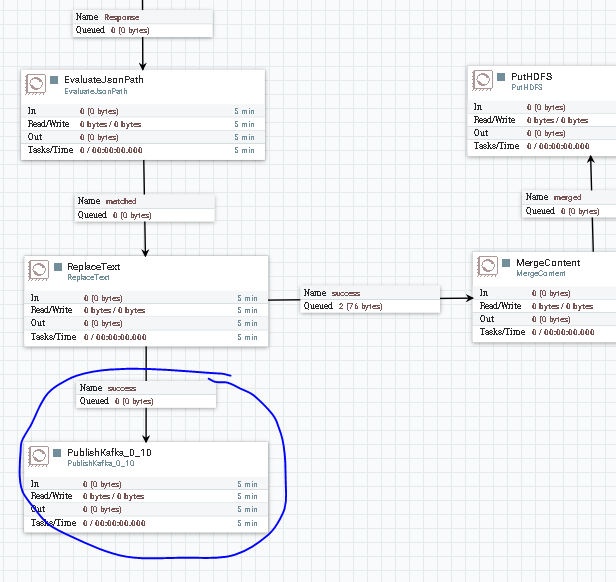
3) Check your created topic by running the following command;

/usr/hdp/current/kafka-broker/bin/kafka-topics.sh --list --zookeeper localhost:2181

4) Go back to NIFI canva, add processor – **PublishKafka\_0\_10**



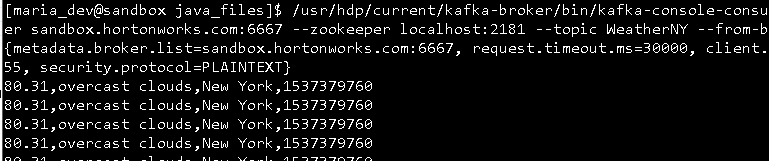
5) Make connection from **ReplaceText**



5) When you run the NIFI, the output data from **replaceText** will be stored in kafka log. You can access to the log in /kafka-logs/WeatherNY-0.

6) To see the published message in a more dynamic way, we can start a console consumer by running following command;

/usr/hdp/current/kafka-broker/bin/kafka-console-consumer.sh --bootstrap-server sandbox.hortonworks.com:6667 --zookeeper localhost:2181 --topic WeatherNY --from-beginning



You will see new row pops up as Nifi calling the API.