Data Flow Across Data Centers

**Table of Contents**

1 Overview 3

2 Business use case 3

3 Solution Architecture 3

4 Build the data flow 4

4.1 Set up Nifi data acquisition instance 5

4.1.1 Download nifi binaries from Apache nifi website 5

4.1.2 Untar or unzip the file based on if you are working with mac or windows 5

4.1.3 Make Configuration changes 5

4.1.4 Start Nifi acquisition instance 6

4.2 Set up Nifi data processing instance 6

4.2.1 Nifi installation and setup 6

4.3 Build Nifi data processing flow 7

4.3.1 Open Nifi Processing instance 7

4.3.2 Setup input port 7

4.3.3 Add process group 7

4.3.4 Setup process group 8

4.4 Build Nifi data acquisition flow 9

4.4.1 Open Nifi acquisition instance 9

4.4.2 Setup process group 10

5 Test Site to Site data flow 12

5.1 Start the Processing Instance 12

5.2 Start the Acquisition Instance 12

5.3 Move data 12

# **Overview**

Traditionally enterprises have been dealing with data flows or data movement within their data centers. But as the world has become more flattened and global presence of companies has become a norm, enterprises are faced with the challenge of collecting and connecting data from their global footprint. This problem was daunting NSA a decade ago and they came up with a solution for this using a product which was later named as Apache Nifi.

Apache nifi is a easy to use, powerful, and reliable system to process and distribute data. Within Nifi, as you will see, I will be able to build a global data flow **with minimal to no Coding**. You can learn the details about Nifi from [Apache Nifi](https://nifi.apache.org/) website. This is one of most well documented Apache projects.

The focus of this article to just look at one specific feature within Nifi that I believe no other software product does it as well as Nifi. And this feature is “site to site” protocol data transfer.

# **Business use case**

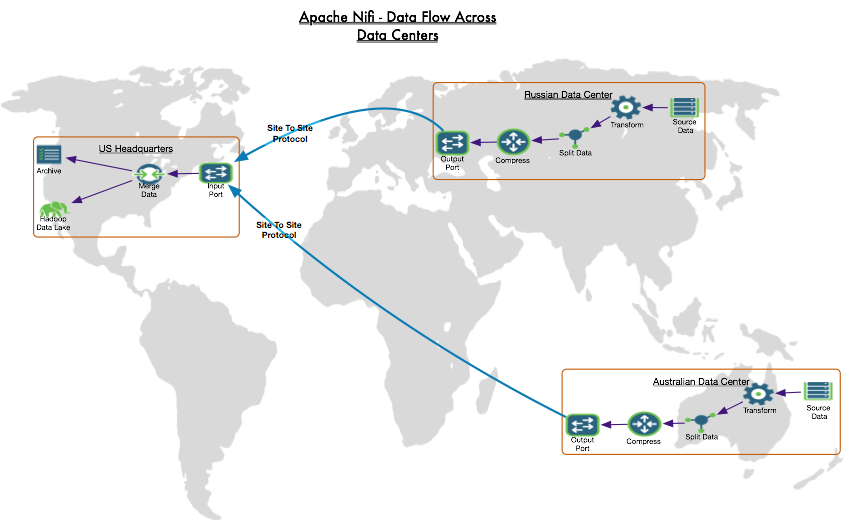
One of the classic business problem is to push data from a location that has a small IT footprint, to the main data center, where all the data is collected and connected. This small IT footprint could be a oil rig at the middle of the ocean, a small bank location at a remote mountain in a town, a sensor on a vehicle so on and so forth. So, your business wants a mechanism to push the data generated at various location to say Headquarters in a reliable fashion, with all the bells and whistles of an enterprise data flow which means maintain lineage, secure, provenance, audit, ease of operations etc.

The data that’s generated at my sources are of various formats such as txt, csv, json, xml, audio, image etc.. and they could of various size ranges from few MBs to GBs. I wanted to break these files into smaller chunks as I have a low bandwidth at my source data centers and want to stich them together at the destination and load that into my centralized Hadoop data lake.

# **Solution Architecture**

Apache Nifi (aka Hortonworks Data Flow) is a perfect tool to solve this problem. The overall architecture looks something like Fig 1.

We have a Australian & Russian data center from where we want to move the data to US Headquarters. We will have what we call edge instance of nifi that will be sitting in Australian & Russian data center, that will act as a data acquisition points. We will then have a Nifi processing cluster in US where we will receive and process all these data coming from global location. We will build this end to end flow without any coding but rather by just a drag and drop GUI interface.



# **Build the data flow**

Here are the high level steps to build the overall data flow.

Step1) Setup a Nifi instance at Australian data center that will act as data acquisition instance. I will create a local instance of Nifi that will act as my Australian data center.

Step2) Setup Nifi instance on a CentOS based virtual machine that will act as my Nifi data processing instance. This could be cluster of Nifi as well but, in my case it will be just a single instance.

Step3) Build Nifi data flow for the processing instance. This will have an input port that will indicate that this instance can accept data from other Nifi instances.

Step4) Build Nifi data for the data acquisition instance. This will have a “remote process group” that will talk to the Nifi data processing instance via site-to-site protocol.

Step5) Test out the overall flow.

## Set up Nifi data acquisition instance

You will use your local machine as your data acquisition instance. Detailed steps for installation could be found at the Nifi website (https://nifi.apache.org/docs.html)

Here is a more summarized version on how to install and configure nifi.

### Download nifi binaries from Apache nifi website

<https://nifi.apache.org/download.html>

### Untar or unzip the file based on if you are working with mac or windows

For mac

tar -xvzf nifi-0.3.0-bin.tar.gz

For windows

Double click on the zip file

### Make Configuration changes

You will be able to find the configuration file under <installdir>/conf

Like for example in my case it is

/Users/csankaraiah/Documents/nifi/nifi-0.3.0/conf

You have to make two changes to the nifi.properties files. Here are the two sections that you need to change.

***# Site to Site properties***

*nifi.remote.input.socket.host=*

*nifi.remote.input.socket.port=9998*

*nifi.remote.input.secure=false*

***# web properties #***

*nifi.web.war.directory=./lib*

*nifi.web.http.host=*

*nifi.web.http.port=9999*

*nifi.web.https.host=*

*nifi.web.https.port=*

*nifi.web.jetty.working.directory=./work/jetty*

*nifi.web.jetty.threads=200*

You need to pick a port number for Site to Site properties (any port that is not currently used by other applications) and also set the secure value to false.

*nifi.remote.input.socket.port=9998*

*nifi.remote.input.secure=false*

You need to pick a port number for web properties (any port that is not currently used by other applications)

*nifi.web.http.port=9999*

### Start Nifi acquisition instance

In order to start nifi you need to go to the bin location (<installdir>/bin) and use nifi.sh script or the bat file based upon the OS.

For Mac (example)

/Users/csankaraiah/Documents/nifi/nifi-0.3.0/bin/nifi.sh start

For Windows

Double click on bat file under bin location.

## Set up Nifi data processing instance

We will use a HDP (Hortonworks Data Platform) sandbox that is running inside a VM as the place where Nifi data processing instance will be installed. I am using HDP sandbox because my objective to store the data eventually into hadoop data lake. Here are the details on how to have HDP sandbox running in your machine

<http://hortonworks.com/products/hortonworks-sandbox/#install>

I could very well install this Nifi instance on any other machine that does not have Hadoop running on it. You need to ensure that where ever you install this instance, your nifi acquisition instance build under section 4.1 can talk to it.

### Nifi installation and setup

You follow the same steps as mentioned under section 4.1

You need to ensure that you use the same site to site port that you used for the data acquisition instance. However, your web port could be a different value

*# Site to Site properties*

*nifi.remote.input.socket.host=*

*nifi.remote.input.socket.port=9998*

*nifi.remote.input.secure=false*

*# web properties #*

*nifi.web.war.directory=./lib*

*nifi.web.http.host=*

*nifi.web.http.port=9090*

*nifi.web.https.host=*

*nifi.web.https.port=*

*nifi.web.jetty.working.directory=./work/jetty*

*nifi.web.jetty.threads=200*

## Build Nifi data processing flow

### Open Nifi Processing instance

Once you have started the nifi instance build under section 4.2, Open up Nifi processing instance on a browser. My processing instance is running on host 172.16.149.157 and I used the port 9090 under the nifi.properties files for the web port. So here is web link for my nifi instance

<http://172.16.149.157:9090/nifi/>

### Setup input port

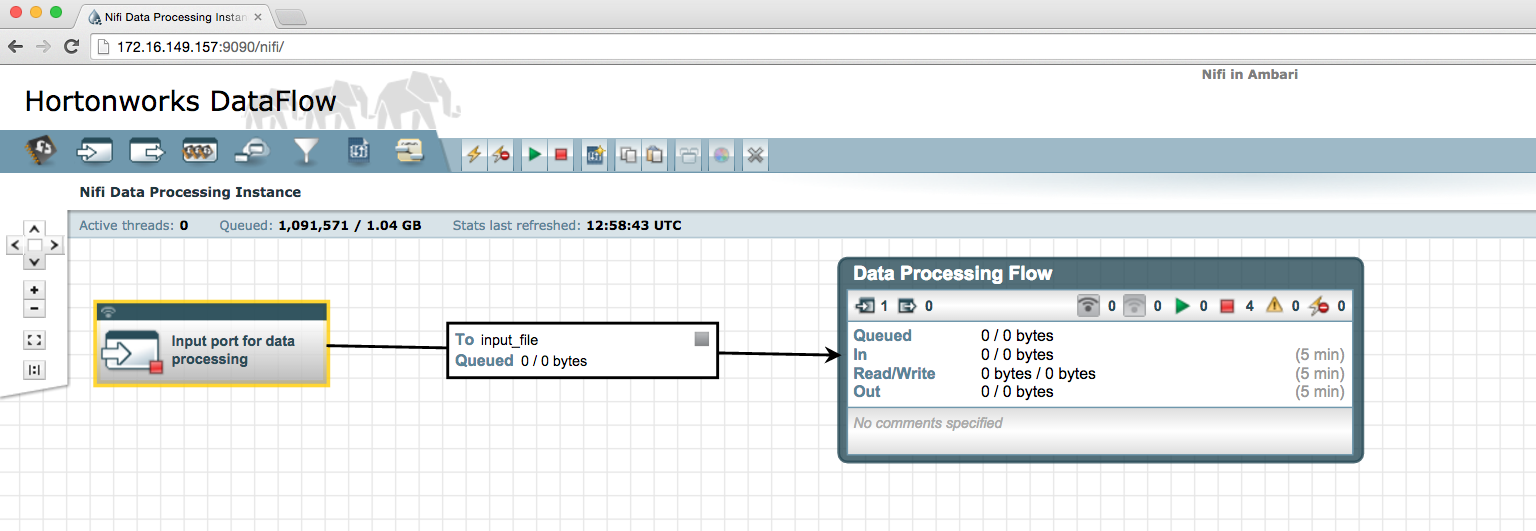
Drag the input port icon which is second from the left under Hortonworks DataFlow title. Name it as “input port for data processing”

### Add process group

Drag the process group icon which is 4th from the left and name it as “Data processing Flow”

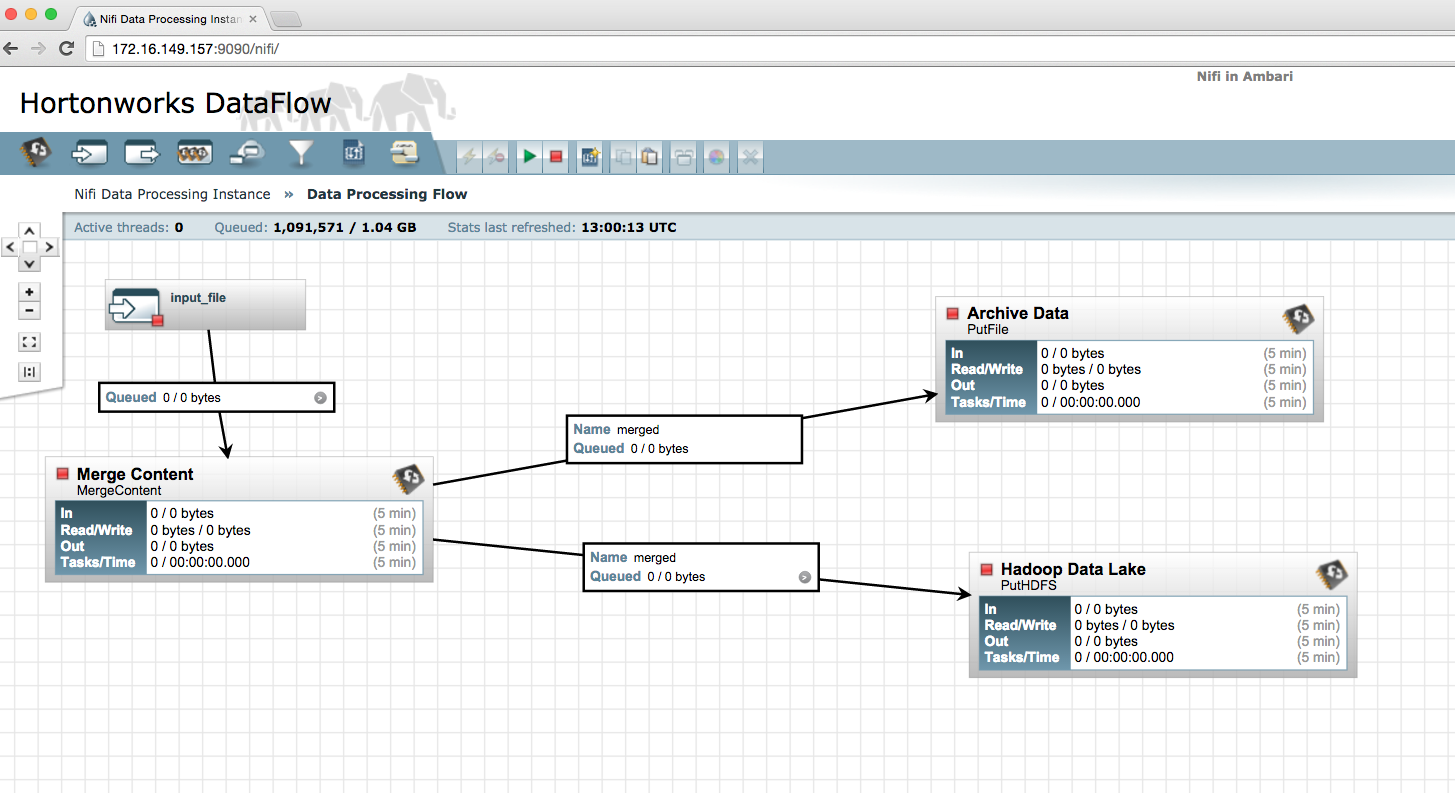
This process subgroup will have bunch of processors within them that will take the input data, merge it and load it into archive & hadoop data lake.

Drag the link from input port to the new process subgroup that you added



### Setup process group

Double click on “Data Processing Flow” subgroup and build this data flow.



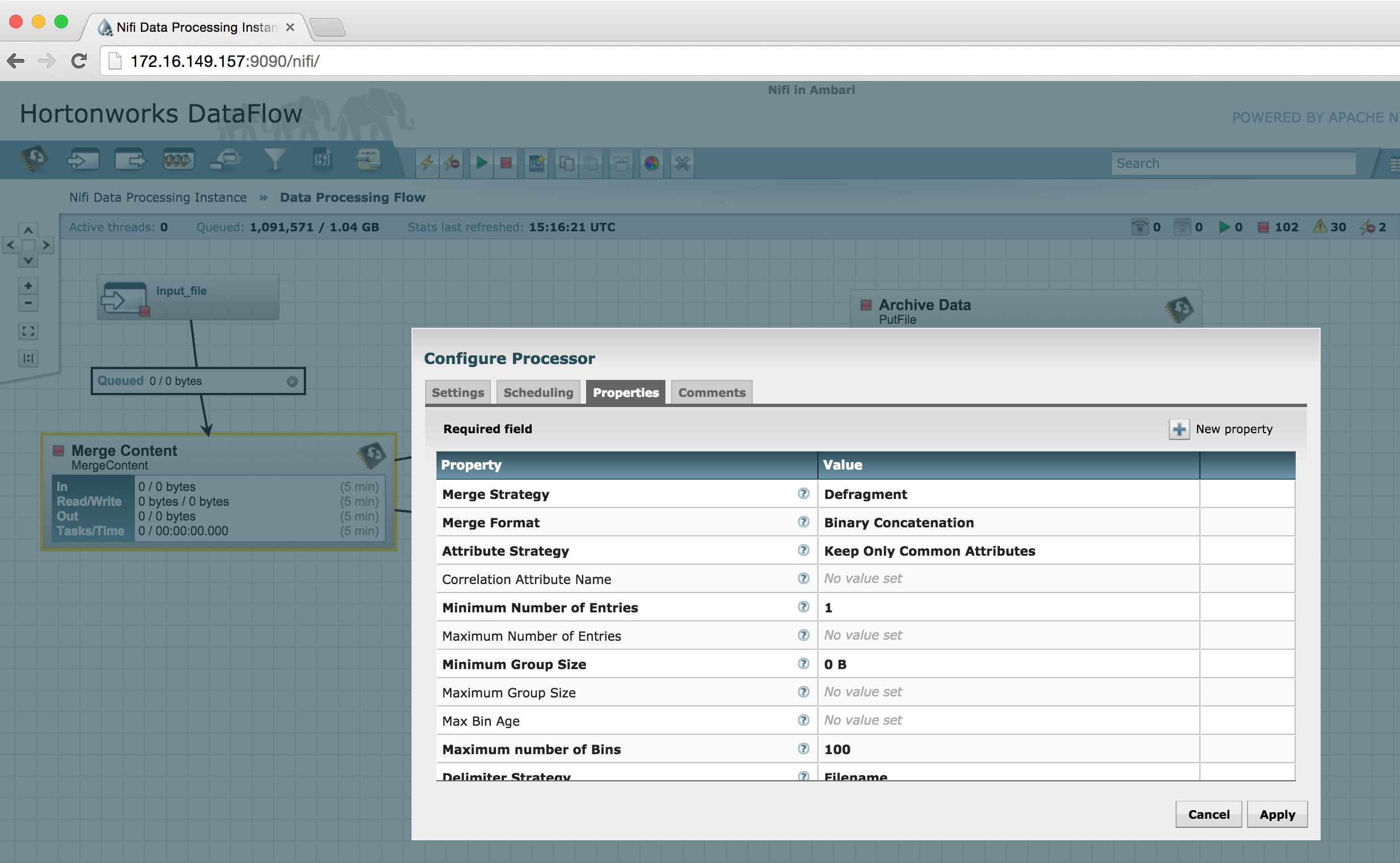
Here are the various processors that are used for the setup along with their configuration

Input\_file

This processor will take the input for the process group. You have to only specify the name as part of the configuration

Merge Content processor

This processor merges the fragmented files back to their original state. Attached is the screen shot of this processor.



PutHDFS Processor ( Hadoop data lake)

This processor loads the data into HDFS in a directory /tmp/tgtfiles

PutFiles Processor ( Archive data)

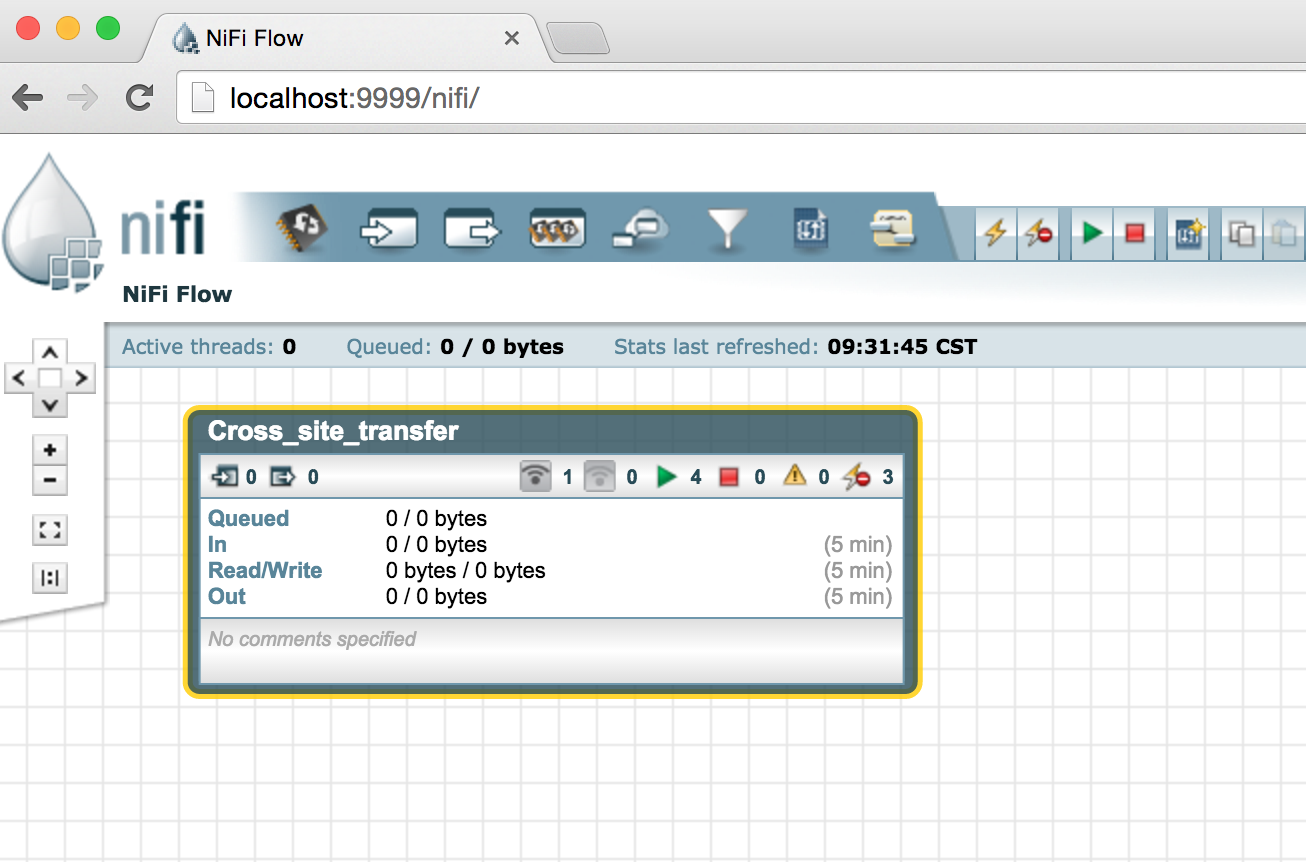
This processor loads the data into a regular directory /tmp/nifitgt

Attached is the screen shot of this processor

## Build Nifi data acquisition flow

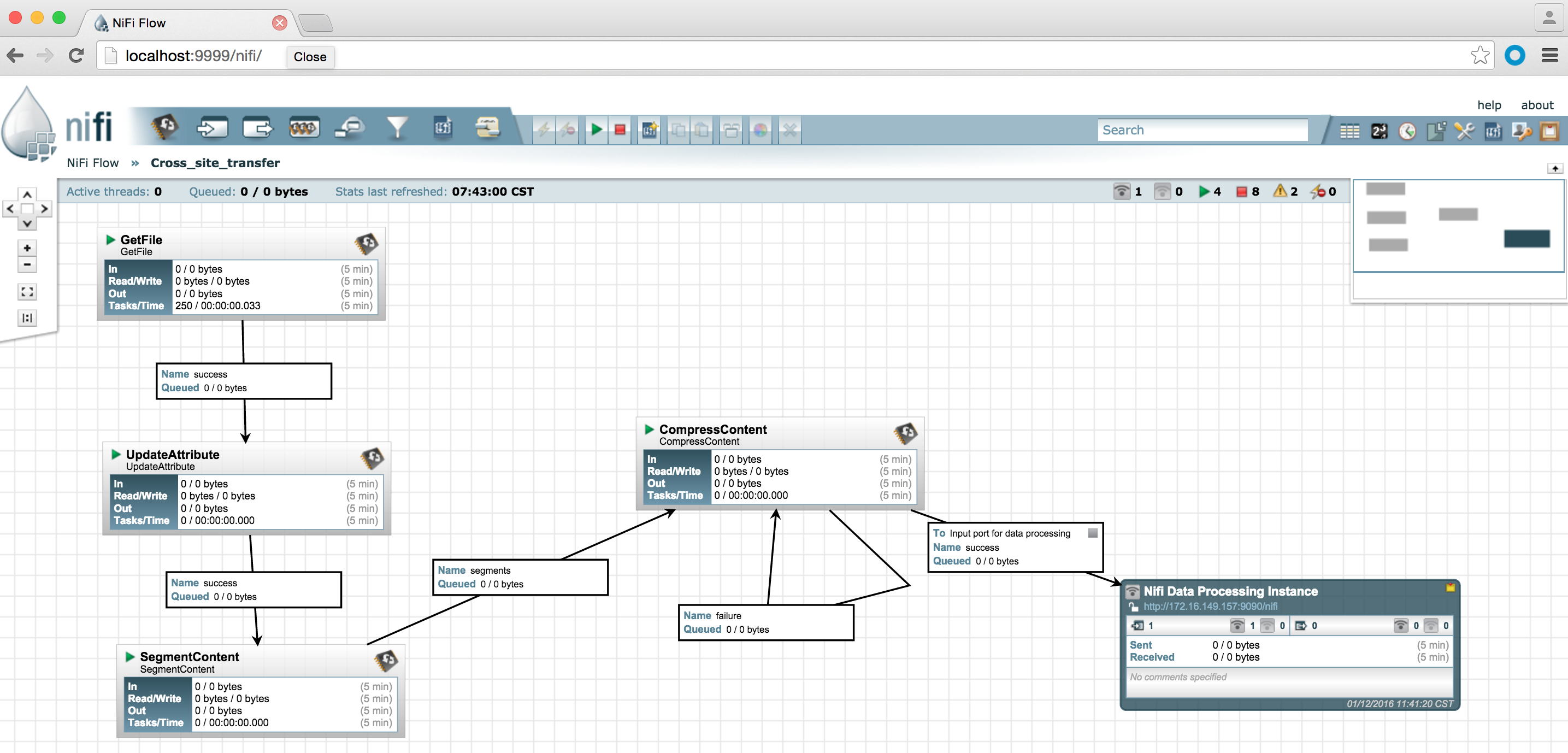
### Open Nifi acquisition instance

My acquisition instance is a local instance so I am using localhost as my server and my web port is 9999. Once you have opened the instance, add a process group and have named it as “Core\_site\_transfer”.



### Setup process group

Double click on “Corss\_site\_transfer” subgroup and build this data flow.



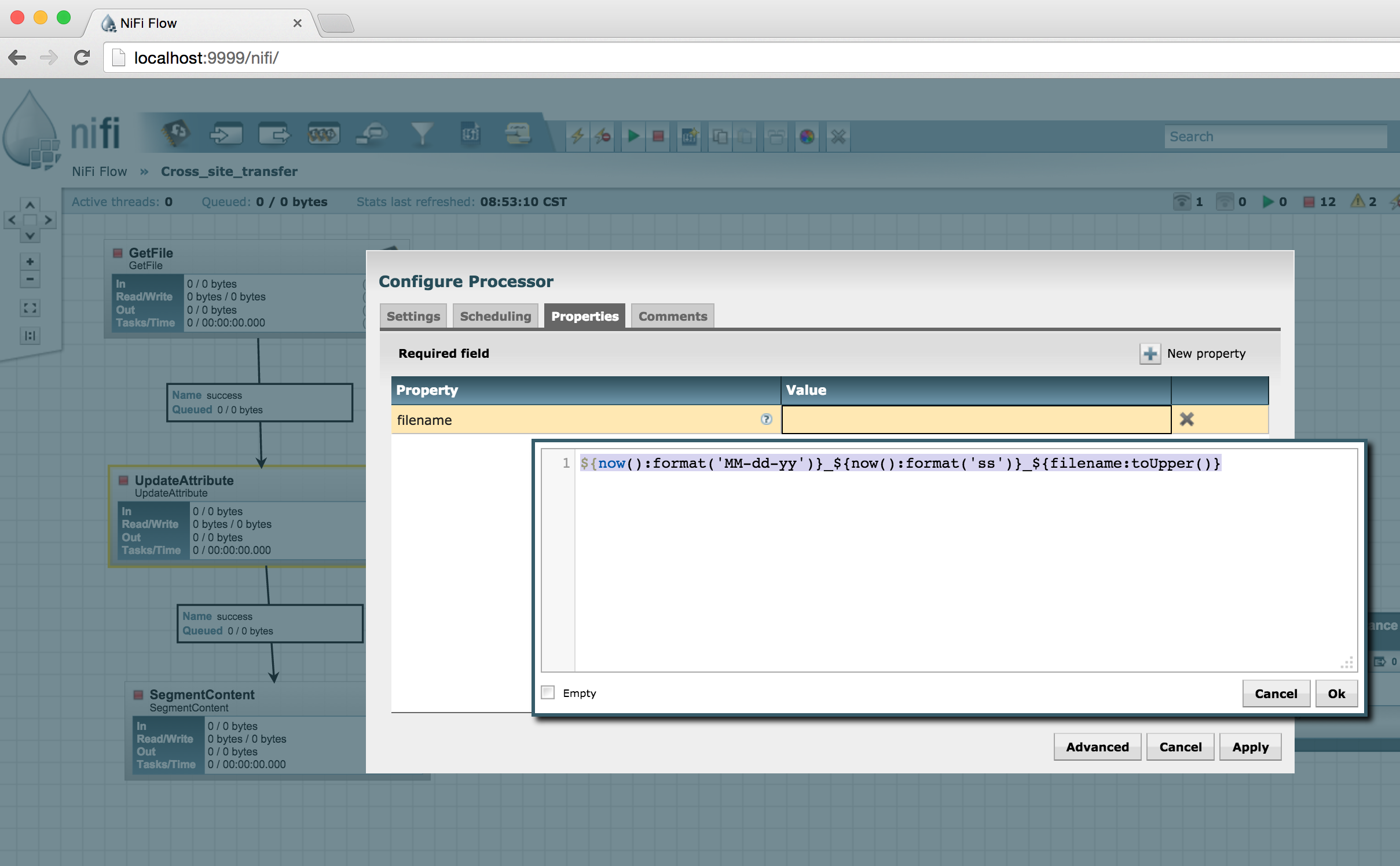
Here is a short description of each of the processor within this process and a screenshot of the configuration.

GetFile Processor

This processor is used to read data from a location. You have to specify the source directory for this processor.

UpdateAttribute processor

This processor is used to append data and time to the filename as it flows through the Nifi data flow.



SegmentContent

This processor splits the files into 100 KB segments. This is really handy if you are dealing with file of large size .This enables you to break those bigger files to smaller segments and then reassemble it at the destination using a nifi merge content processor.

Compress Content Processor

This processor compresses those segmented files from previous process and makes them further smaller in size so that its easier to move the data from source to destination that have low bandwidth. We will be using gzip for compression.

Remote Process group

This processor is used to communicate with the remote cluster to which data will be moved to.

So if you have multiple data centers this is the processor that you will be using to communicate between various instances of nifi.

In our case this processor is where we will provide the URL for the nifi processing instance that we setup under previous section.

# **Test Site to Site data flow**

As have build this acquisition instance and processing instance, it time now to test this out end to end. Here are the steps we are going to execute

Start the Processing Instance

Start the Acquisition instance

Move data

## Start the Processing Instance

Login to the processing instance and click on the start button available at the top ribbon.

## Start the Acquisition Instance

Login to the acquisition instance and click on the start button avaiablle at the top ribbon.

Also ensure that you click on the Remoet processor group and select “Enable Transmission”.

## Move data

Load the data to source directory and check the results in destination HDFS and archive location.

Here is the source data flow with files that was moved from source

here is the target data flow screen shot.

