



Data Flow in Hadoop – Apache NiFi



About Me

- Jacob
- Work for III
- NYUST dept. IM
- Hadoop, Python, R, etc..



About Me

FB: 陳建融

Web: <http://chenjr-jacob.idv.tw/>

GitHub: <https://github.com/chenjr0719>

Mail: jacobchen@iii.org.tw

or: chenjr0719@gmail.com



Contents

1. Backgrounds

2. Hortonworks DataFlow (HDF)

3. Hands On



Backgrounds



Situation

- Sensor -> Database -> Backend



And More..

- App -> Database -> Backend
- Online Shop -> OLAP -> Backend



This is FLOW.



What is Data Flow

- Source
- Sink
- Between **Source** and **Sink** – **Pipeline**.



In real world

- The data from source is **Real Time**.
- And this also called **Streaming**.



Common data source

- Facebook, Twitter (SNS)
- Factory machine (Sensor)
- PTT, Dcard, News (Web)
- App
- IoT



Role of Hadoop

- Data Warehouse
- Process Platform
- Analysis Platform
- Most of them is about **Batch** processing.



But..

Streaming is more important than before.



Now..

- Many projects aim to deal with **Streaming**.
 - NiFi
 - Spark Streaming
 - Flink
 - Storm
- And more..



Hortonworks DataFlow (HDF)

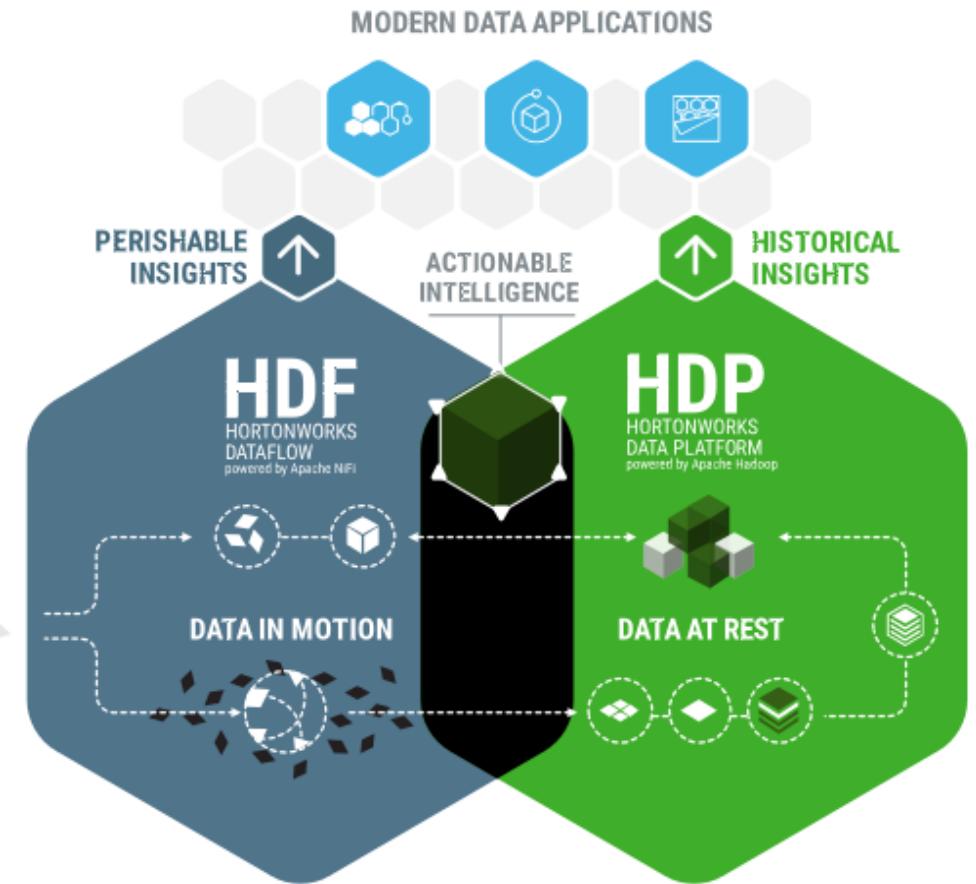


Hortonworks

- The company focuses on the development and support of **Apache Hadoop**.
- Hortonworks Data Platform (HDP®)
- Hortonworks DataFlow (HDF™)

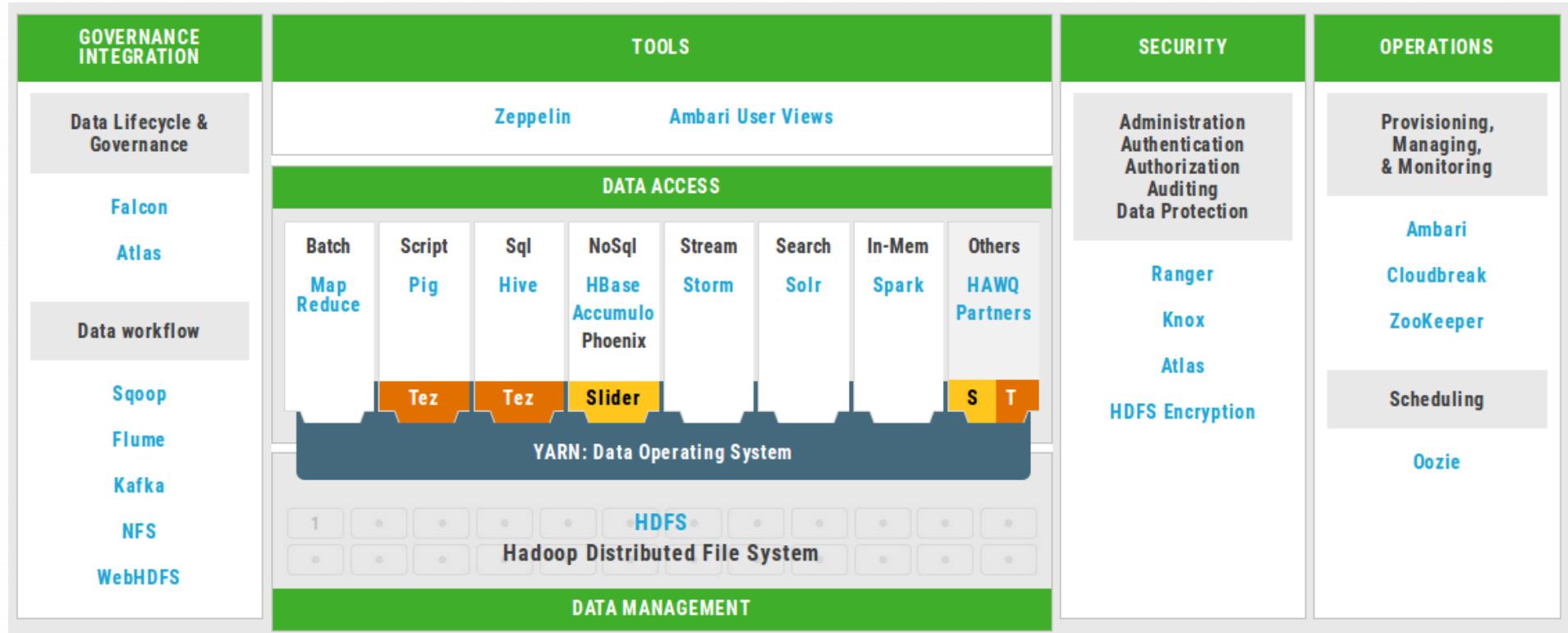


Hortonworks Solution





Hortonworks Overviews





- ZooKeeper
- Storm
- Ambari Infra
- Kafka
- NiFi
- <https://youtu.be/-2Zw3n1qQUA>



Apache NiFi

- An easy to use, powerful, and reliable system to process and distribute data.
- Web UI
- Highly configurable



Apache Zookeeper

- A centralized service.
- Providing distributed synchronization
- Providing group services



Storm

- Distributed realtime computation system.
- Streaming.
- Can be used with any programming language.



Ambari Infra

- Provide common shared services for stack components



Kafka

- Publish-subscribe messaging system.
- Fast
- Scalable



Hands On



Hands On

1. Install HDF
2. NiFi with Kafka
3. HDF to HDP
4. Let's Tweet
5. Pre-Process
6. ExecuteProcess
7. To Spark Streaming



Install HDF

```
sudo wget -nv http://public-repo-  
1.hortonworks.com/ambari/centos6/2.x/updates  
/2.4.0.1/ambari.repo -O  
/etc/yum.repos.d/ambari.repo
```

```
sudo yum -y install ambari-server
```



Install HDF

```
sudo ambari-server setup -s
```

```
sudo ambari-server install-mpack  
--mpack=http://public-repo-  
1.hortonworks.com/HDF/centos6/2.x/updates/2.  
0.0.0/tars/hdf_ambari_mp/hdf-ambari-mpack-  
2.0.0.0-579.tar.gz --purge --verbose
```

```
sudo ambari-server start
```



Install HDF

The screenshot shows the Ambari Cluster Install Wizard. The top navigation bar includes the Ambari logo, a user icon labeled "admin", and a "grid" icon. The left sidebar, titled "CLUSTER INSTALL WIZARD", lists the following steps: Get Started (selected), Select Version, Install Options, Confirm Hosts, Choose Services, Assign Masters, Assign Slaves and Clients, Customize Services, Review, Install, Start and Test, and Summary. The main content area is titled "Get Started" and contains the following text: "This wizard will walk you through the cluster installation process. First, start by naming your new cluster." Below this is a text input field with the placeholder "Name your cluster" and a link "Learn more". The input field contains the text "HDF". A green "Next →" button is located at the bottom right of the main panel.

Licensed under the Apache License, Version 2.0.
See third-party tools/resources that Ambari uses and their respective authors



Install HDF

The screenshot shows the Ambari Cluster Install Wizard on the 'Select Version' step. The left sidebar lists steps: Get Started, Select Version (highlighted), Install Options, Confirm Hosts, Choose Services, Assign Masters, Assign Slaves and Clients, Customize Services, Review, Install, Start and Test, and Summary.

The main panel title is 'Select Version'. It contains a note: 'Select the software version and method of delivery for your cluster. Using a Public Repository requires Internet connectivity. Using a Local Repository requires you have configured the software in a repository available in your network.' A dropdown menu shows 'HDF-2.0.0.0'. Below it is a table of services and their versions:

Service	Version
Ambari Infra	0.1.0
Ambari Metrics	0.1.0
Kafka	0.10.0
Log Search	0.5.0
NiFi	1.0.0.2.0
Ranger	0.6.0
Storm	1.0.1

Below the table, there are two radio buttons: 'Use Public Repository' (selected) and 'Use Local Repository'. The 'Repositories' section shows base URLs for various operating systems:

OS	Name	Base URL	Action
debian7	HDF-2.0	http://public-repo-1.hortonworks.com/HDF/debian7/2.x/u	- Remove
	HDP-UTILS-1.1.0.21	http://public-repo-1.hortonworks.com/HDP-UTILS-1.1.0.21	- Remove
redhat6	HDF-2.0	http://public-repo-1.hortonworks.com/HDF/centos6/2.x/u	- Remove
	HDP-UTILS-1.1.0.21	http://public-repo-1.hortonworks.com/HDP-UTILS-1.1.0.21	- Remove
redhat7	HDF-2.0	http://public-repo-1.hortonworks.com/HDF/centos7/2.x/u	- Remove
	HDP-UTILS-1.1.0.21	http://public-repo-1.hortonworks.com/HDP-UTILS-1.1.0.21	- Remove
suse11	HDF-2.0	http://public-repo-1.hortonworks.com/HDF/suse11sp3/2.	- Remove
	HDP-UTILS-1.1.0.21	http://public-repo-1.hortonworks.com/HDP-UTILS-1.1.0.21	- Remove



Install HDF

Ambari

admin

CLUSTER INSTALL WIZARD

Get Started

Select Version

Install Options

Confirm Hosts

Choose Services

Assign Masters

Assign Slaves and Clients

Customize Services

Review

Install, Start and Test

Summary

Install Options

Enter the list of hosts to be included in the cluster and provide your SSH key.

Target Hosts

Enter a list of hosts using the Fully Qualified Domain Name (FQDN), one per line. Or use [Pattern Expressions](#)

hdf.maas

Host Registration Information

Provide your [SSH Private Key](#) to automatically register hosts

[Browse...](#) id_rsa

-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEayKLs1G601Bbifa5pz371EzNzw4P5hqtrrkA30gFXNtLa
SWSX
+rt2rn8/vv1WhCMyhInu4WnRkTVhnuREh7hAN1n0AcDKMvUkMOV1I f2DnVAM

SSH User Account centos

SSH Port Number 22

Perform [manual registration](#) on hosts and do not use SSH

[← Back](#) [Register and Confirm →](#)

Licensed under the Apache License, Version 2.0.
See third-party tools/resources that Ambari uses and their respective authors



Install HDF

Ambari

CLUSTER INSTALL WIZARD

Get Started

Select Version

Install Options

Confirm Hosts

Choose Services

Assign Masters

Assign Slaves and Clients

Customize Services

Review

Install, Start and Test

Summary

Confirm Hosts

Registering your hosts.
Please confirm the host list and remove any hosts that you do not want to include in the cluster.

	Host	Progress	Status	Action
<input type="checkbox"/>	hdf.maas	<div style="width: 100%;"> </div>	Installing	Remove

Show: All (1) | [Installing \(1\)](#) | [Registering \(0\)](#) | [Success \(0\)](#) | [Fail \(0\)](#)

Show: 25 | 1 - 1 of 1 | [Back](#) | [Next](#)

Licensed under the Apache License, Version 2.0.
See third-party tools/resources that Ambari uses and their respective authors



Install HDF

Ambari

CLUSTER INSTALL WIZARD

Get Started

Select Version

Install Options

Confirm Hosts

Choose Services

Assign Masters

Assign Slaves and Clients

Customize Services

Review

Install, Start and Test

Summary

Confirm Hosts

Registering your hosts.
Please confirm the host list and remove any hosts that you do not want to include in the cluster.

<input type="checkbox"/> Host	Progress	Status	Action
<input type="checkbox"/> hdf.maas	<div style="width: 100%;">Success</div>	Success	<input type="button" value="Remove"/>

Show: All (1) | Installing (0) | Registering (0) | Success (1) | Fail (0)

Show: 25 1 - 1 of 1

Some warnings were encountered while performing checks against the 1 registered hosts above [Click here to see the warnings.](#)

[← Back](#) [Next →](#)

Licensed under the Apache License, Version 2.0.
See third-party tools/resources that Ambari uses and their respective authors



Install HDF

Ambari

admin

CLUSTER INSTALL WIZARD

Get Started

Select Version

Install Options

Confirm Hosts

Choose Services

Assign Masters

Assign Slaves and Clients

Customize Services

Review

Install, Start and Test

Summary

Choose Services

Choose which services you want to install on your cluster.

<input type="checkbox"/> Service	Version	Description
<input checked="" type="checkbox"/> ZooKeeper	3.4.6	Centralized service which provides highly reliable distributed coordination
<input checked="" type="checkbox"/> Storm	1.0.1	Apache Hadoop Stream processing framework
<input checked="" type="checkbox"/> Ambari Infra	0.1.0	Core shared service used by Ambari managed components.
<input type="checkbox"/> Ambari Metrics	0.1.0	A system for metrics collection that provides storage and retrieval capability for metrics collected from the cluster
<input checked="" type="checkbox"/> Kafka	0.10.0	A high-throughput distributed messaging system
<input type="checkbox"/> Log Search	0.5.0	Log aggregation, analysis, and visualization for Ambari managed services. This service is Technical Preview .
<input checked="" type="checkbox"/> NiFi	1.0.0.2.0	Apache NiFi is an easy to use, powerful, and reliable system to process and distribute data.

← Back Next →

Licensed under the Apache License, Version 2.0.
See third-party tools/resources that Ambari uses and their respective authors



Install HDF

Ambari

admin

CLUSTER INSTALL WIZARD

Get Started

Select Version

Install Options

Confirm Hosts

Choose Services

Assign Masters

Assign Slaves and Clients

Customize Services

Review

Install, Start and Test

Summary

Assign Masters

Assign master components to hosts you want to run them on.

ZooKeeper Server: hdf.maas (5.8 GB, 2 cores)

Nimbus: hdf.maas (5.8 GB, 2 cores)

DRPC Server: hdf.maas (5.8 GB, 2 cores)

Storm UI Server: hdf.maas (5.8 GB, 2 cores)

Infra Solr Instance: hdf.maas (5.8 GB, 2 cores)

Kafka Broker: hdf.maas (5.8 GB, 2 cores)

NiFi: hdf.maas (5.8 GB, 2 cores)

hdf.maas (5.8 GB, 2 cores)

ZooKeeper Server Nimbus DRPC Server
Storm UI Server Infra Solr Instance
Kafka Broker NiFi

← Back

Next →

Licensed under the Apache License, Version 2.0.
See third-party tools/resources that Ambari uses and their respective authors



Install HDF

Ambari

CLUSTER INSTALL WIZARD

Get Started

Select Version

Install Options

Confirm Hosts

Choose Services

Assign Masters

Assign Slaves and Clients

Customize Services

Review

Install, Start and Test

Summary

Assign Slaves and Clients

Assign slave and client components to hosts you want to run them on.
Hosts that are assigned master components are shown with *.
"Client" will install ZooKeeper Client and Infra Solr Client.

Host	all none	all none	all none
hdf.maas*	<input checked="" type="checkbox"/> Supervisor	<input checked="" type="checkbox"/> NiFi Certificate Authority	<input checked="" type="checkbox"/> Client

Show: 25 1 - 1 of 1

← Back Next →

Licensed under the Apache License, Version 2.0.
See third-party tools/resources that Ambari uses and their respective authors



Install HDF

The screenshot shows the Ambari Cluster Install Wizard at the 'Customize Services' step. The left sidebar lists the wizard steps: Get Started, Select Version, Install Options, Confirm Hosts, Choose Services, Assign Masters, Assign Slaves and Clients, **Customize Services**, Review, Install, Start and Test, and Summary. The 'Customize Services' step is currently selected. The main panel title is 'Customize Services' with the sub-instruction: 'We have come up with recommended configurations for the services you selected. Customize them as you see fit.' Below this, tabs for ZooKeeper, Storm, Ambari Infra, Kafka, NiFi (with a red notification badge '1'), and Misc are shown, with NiFi selected. A toolbar includes 'Group' dropdown (set to 'Default (1)'), 'Manage Config Groups', and a 'Filter...' dropdown. A section titled 'Advanced nifi-ambari-config' contains two input fields: 'Sensitive property values' (Type password) and 'encryption password' (Retype Password), both marked as 'This is required'. An attention message at the bottom states: '⚠ Attention: Some configurations need your attention before you can proceed. Showing properties with issues. [Show all properties](#)'.

Licensed under the Apache License, Version 2.0.
See third-party tools/resources that Ambari uses and their respective authors



Install HDF

The screenshot shows the Ambari Cluster Install Wizard at the 'Customize Services' step. The left sidebar lists the wizard steps: Get Started, Select Version, Install Options, Confirm Hosts, Choose Services, Assign Masters, Assign Slaves and Clients, **Customize Services**, Review, Install, Start and Test, and Summary. The 'Customize Services' step is currently selected. The main panel title is 'Customize Services' with the sub-instruction: 'We have come up with recommended configurations for the services you selected. Customize them as you see fit.' Below this, tabs for ZooKeeper, Storm, Ambari Infra, Kafka, NiFi, and Misc are shown, with NiFi selected. A toolbar includes 'Group', 'Default (1)', 'Manage Config Groups', and a 'Filter...' dropdown. A section titled 'Advanced nifi-ambari-config' contains two redacted sensitive property values for 'encryption password'. A green success message at the bottom states: 'All configurations have been addressed.' with a link to 'Show all properties'. Navigation buttons 'Back' and 'Next >' are at the bottom.

Licensed under the Apache License, Version 2.0.
See third-party tools/resources that Ambari uses and their respective authors



Install HDF

Ambari

admin

CLUSTER INSTALL WIZARD

- Get Started
- Select Version
- Install Options
- Confirm Hosts
- Choose Services
- Assign Masters
- Assign Slaves and Clients
- Customize Services
- Review
- Install, Start and Test
- Summary

Customize Services

We have come up with recommended configurations for the services you selected. Customize them as you see fit.

ZooKeeper Storm Ambari Infra Kafka NiFi Misc

Group Default (1) Manage Config Groups remote

Advanced nifi-bootstrap-env

Advanced nifi-properties

nifi.cluster.node.read.timeout	5 sec
nifi.remote.input.host	
nifi.remote.input.http.enabled	<input checked="" type="checkbox"/>
nifi.remote.input.http.transaction.ttl	30 sec
nifi.remote.input.secure	<code>{{nifi_ssl_enabled}}</code>
nifi.remote.input.socket.port	8850

All configurations have been addressed.

Back Next

Licensed under the Apache License, Version 2.0.
See third-party tools/resources that Ambari uses and their respective authors



Install HDF

Ambari admin

Configurations

Some service configurations are not configured properly. We recommend you review and change the highlighted configuration values. Are you sure you want to proceed without correcting configurations?

Type	Service	Property	Value	Description
Warning	NiFi	nifi.node.ssl.isenabled	false	For NiFi Certificate Authority to be useful, ssl should be enabled Check flag to enable SSL. A few additional properties are also required (depending on your setup). Assuming NiFi's Certificate Authority (CA) was installed: If Ranger auth will be used, only 'Nifi CA Token' is required - otherwise 'Nifi CA Token', 'Initial Admin Identity', 'Node Identities' are all required. Assuming CA is not installed: If Ranger auth will be used, only Truststore/Keystore paths/type/passwords should be set - otherwise Truststore/Keystore paths/type/passwords as well as 'Initial Admin Identity', 'Node Identities' are required.
Error	NiFi	nifi.toolkit.tls.token		If NiFi Certificate Authority is used, nifi.toolkit.tls.token must be set This is a token that will be used by the NiFi Certificate Authority to verify the identity of NiFi nodes before issuing them certificates and by the NiFi nodes to verify the identity of the NiFi Certificate Authority. If relying on NiFi Certificate Authority, set this to a long, random value. For security purposes, password changes will not be shown in configuration version comparisons

Cancel **Proceed Anyway**

nifi.cluster.node.read.timeout: 5 sec
nifi.remote.input.host:
nifi.remote.input.http.enabled: C
nifi.remote.input.http.transaction.ttl: 30 sec
nifi.remote.input.secure: {{nifi_ssl_enabled}}
nifi.remote.input.socket.port: 8850

~ Back Next ~

Licensed under the Apache License, Version 2.0.
See third-party tools/resources that Ambari uses and their respective authors



Install HDF

Ambari

admin

CLUSTER INSTALL WIZARD

- Get Started
- Select Version
- Install Options
- Confirm Hosts
- Choose Services
- Assign Masters
- Assign Slaves and Clients
- Customize Services
- Review
- Install, Start and Test**
- Summary

Install, Start and Test

Please wait while the selected services are installed and started.

3 % overall

Host	Status	Message
hdf.maas	3%	Waiting to install DRPC Server

1 of 1 hosts showing - Show All

Show: 25 1 - 1 of 1

Next →

Licensed under the Apache License, Version 2.0.
See third-party tools/resources that Ambari uses and their respective authors



Install HDF

Ambari

admin

CLUSTER INSTALL WIZARD

- Get Started
- Select Version
- Install Options
- Confirm Hosts
- Choose Services
- Assign Masters
- Assign Slaves and Clients
- Customize Services
- Review
- Install, Start and Test**
- Summary

Install, Start and Test

Please wait while the selected services are installed and started.

100 % overall

Host	Status	Message
hdf.maas	100%	Success

1 of 1 hosts showing - Show All

Show: 25 1 - 1 of 1

Successfully installed and started the services.

Next →

Licensed under the Apache License, Version 2.0.
See third-party tools/resources that Ambari uses and their respective authors



Install HDF

The screenshot shows the Ambari Cluster Install Wizard Summary page. The top navigation bar includes the Ambari logo, a user icon labeled "admin", and a grid icon. The left sidebar lists the steps of the wizard: Get Started, Select Version, Install Options, Confirm Hosts, Choose Services, Assign Masters, Assign Slaves and Clients, Customize Services, Review, Install, Start and Test, and Summary. The "Summary" button is highlighted with a dark grey background. The main content area is titled "Summary" and contains a message: "Here is the summary of the install process." Below this, it states: "The cluster consists of 1 hosts", "Installed and started services successfully on 1 new host", "Master services installed", "All services started", "All tests passed", and "Install and start completed in 12 minutes and 31 seconds". A green "Complete →" button is located at the bottom right of the summary box.

Licensed under the Apache License, Version 2.0.
See third-party tools/resources that Ambari uses and their respective authors



Result

The screenshot shows the Ambari Metrics dashboard interface. At the top, there is a navigation bar with the Ambari logo, the text "HDF 0 ops 0 alerts", and links for Dashboard, Services, Hosts, Alerts, Admin, and a user dropdown for "admin". On the left, a sidebar lists checked services: ZooKeeper, Storm, Ambari Infra, Kafka, and NiFi, with an "Actions" button below it. The main area has tabs for Metrics (selected), Heatmaps, and Config History, and dropdowns for Metric Actions and Last 1 hour. Below these are five cards: Memory Usage, Network Usage, CPU Usage, Cluster Load, and Supervisors Live, each showing "No Data Available". The Supervisors Live card displays "1/1" in green. At the bottom, there is a footer note about Apache License Version 2.0 and third-party tools.

Licensed under the Apache License, Version 2.0.
See third-party tools/resources that Ambari uses and their respective authors



Result

Storm UI

Cluster Summary

Version	Supervisors	Used slots	Free slots	Total slots	Executors	Tasks
1.0.1.2.0.0.0-579	1	0	2	2	0	0

Nimbus Summary

Host	Port	Status	Version	UpTime
hdf.maas	6627	Offline	Not applicable	Not applicable
HDF.maas	6627	Leader	1.0.1.2.0.0.0-579	7m 24s

Showing 1 to 2 of 2 entries

Topology Summary

Name	Owner	Status	Uptime	Num workers	Num executors	Num tasks	Replication count	Assigned Mem (MB)	Scheduler Info
No data available in table									

Showing 0 to 0 of 0 entries

Supervisor Summary

Host	Id	Uptime	Slots	Used slots	Used Mem (MB)	Version
HDF.maas	d58e60d2-709f-4542-b230-9782b2ca7511	5m 14s	2	0	0	1.0.1.2.0.0.0-579

Showing 1 to 1 of 1 entries

Nimbus Configuration

Show	entries	Search:
Key	Value	
backpressure.disruptor.high.watermark	0.9	



Result

Solr

Dashboard

No cores available
Go and create one

Instance

Start 11 minutes ago

Versions

Spec	Version	Commit	Date
solr-spec	5.5.2	5.5.2 8e5d40b22a3968df065dfc078ef81cbb031f0e4a	- sarowe - 2016-06-21 11:44:11
solr-impl	5.5.2 8e5d40b22a3968df065dfc078ef81cbb031f0e4a	- sarowe - 2016-06-21 11:44:11	
lucene-spec	5.5.2	5.5.2 8e5d40b22a3968df065dfc078ef81cbb031f0e4a	- sarowe - 2016-06-21 11:38:23
lucene-impl	5.5.2 8e5d40b22a3968df065dfc078ef81cbb031f0e4a	- sarowe - 2016-06-21 11:38:23	

JVM

Runtime: Oracle Corporation Java HotSpot(TM) 64-Bit Server VM (1.8.0_77 25.77-b03)

Processors: 2

Args:

```
-DSTOP.KEY=solrrocks
-DSTOP.PORT=7886
-Dcom.sun.management.jmxremote
-Dcom.sun.management.jmxremote.authenticate=false
-Dcom.sun.management.jmxremote.local.only=false
-Dcom.sun.management.jmxremote.port=18886
-Dcom.sun.management.jmxremote.rmi.port=18886
-Dcom.sun.management.jmxremote.ssl=false
-Djetty.home=/usr/lib/ambari-infra-solr/server
-Djetty.port=8886
-Dlog4j.configuration=file:/etc/ambari-infra-solr/conf/log4j.properties
-Dsolr.install.dir=/usr/lib/ambari-infra-solr
-Dsolr.solr.home=/opt/ambari_infra_solr/data
-Duser.timezone=UTC
-DzkClientTimeout=60000
-DzkHost=hdf.maas:2181/infra-solr
-XX:+CMSParallelRemarkEnabled
-XX:+CMSScavengeBeforeRemark
-XX:+ParallelRefProcEnabled
-XX:+PrintGCApplicationStoppedTime
-XX:+PrintGCDetails
-XX:+PrintGCTimeStamps
-XX:+PrintHeapAtGC
-XX:+PrintTenuringDistribution
-XX:+UseCMSInitiatingOccupancyOnly
-XX:+UseConcMarkSweepGC
-XX:+UseParNewGC
-XX:CMSInitiatingOccupancyFraction=50
-XX:CMSMaxAbortablePrecleanTime=6000
```

System

Physical Memory: 90.3% (5.27 GB / 5.84 GB)

Swap Space: NaN% (0.00 MB / 0.00 MB)

File Descriptor Count: 2.6% (105 / 4096)

JVM-Memory: 3.2% (62.23 MB / 1.92 GB)

Try New UI



Result

The screenshot shows the Apache NiFi user interface. The top navigation bar includes icons for file operations (New, Open, Save, Print, etc.), a search bar, and a timestamp of 09:46:52 UTC. Below the header are two side panels: 'Navigate' and 'Operate'. The 'Navigate' panel contains a search bar and a large empty box for listing components. The 'Operate' panel displays a 'NiFi Flow Process Group' named '41d13a78-0157-1000-44ab-3a8e7d1f3537' with various management icons (gear, lightning bolt, play, etc.) and a 'DELETE' button.



NiFi with Kafka

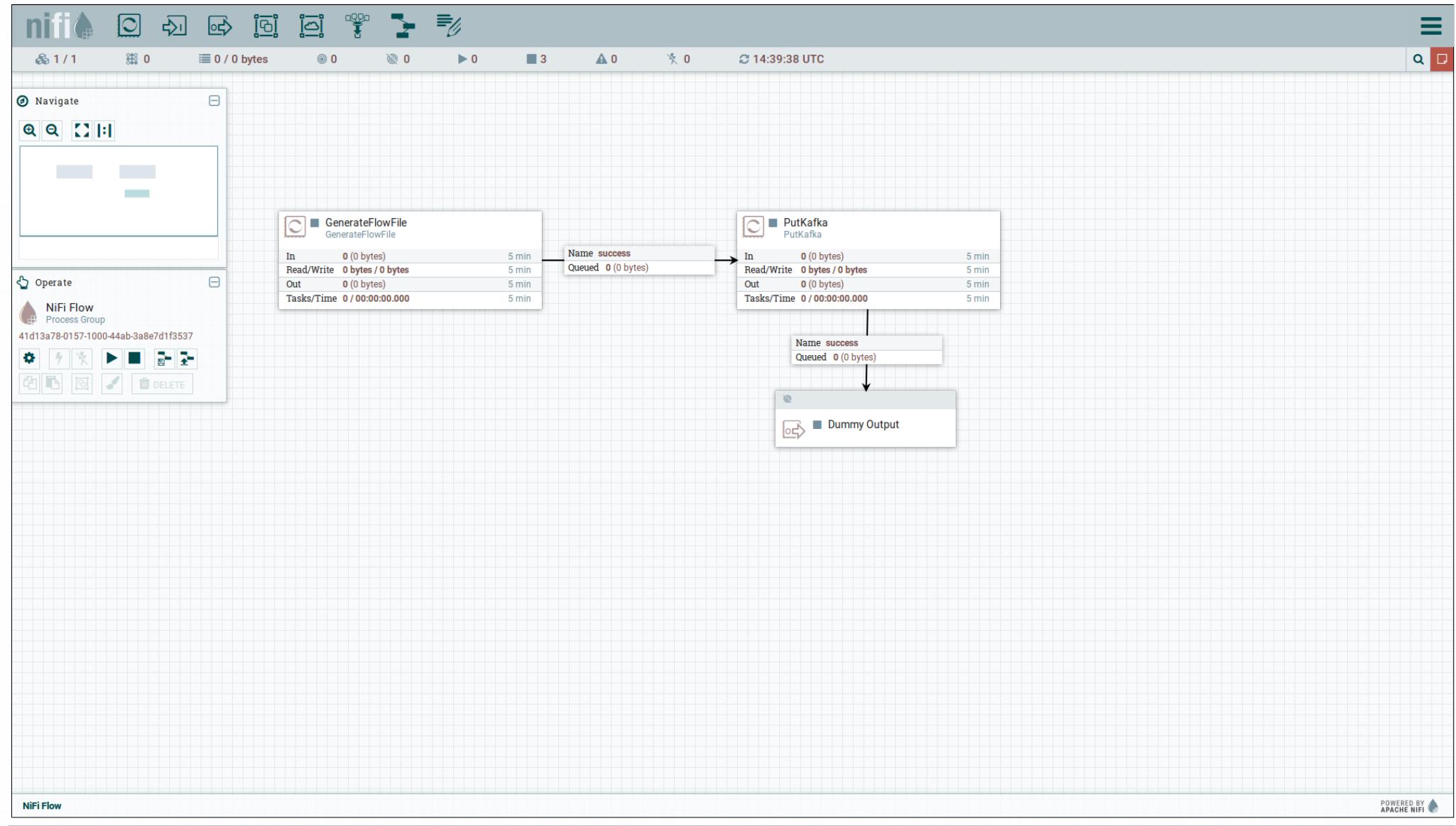


Create Kafka topic

```
cd /usr/hdf/2.0.0.0-579/kafka/bin/  
./kafka-topics.sh --create --zookeeper  
localhost:2181 --replication-factor 1 --partitions  
2 --topic test  
./kafka-topics.sh --list --zookeeper  
localhost:2181
```



NiFi





NiFi

The screenshot shows the Apache NiFi web interface. On the left, there's a sidebar with navigation links like 'Navigate', 'Operate', and a list of processors including 'GenerateFlowFile'. The main area displays a 'Configure Processor' dialog for a 'GenerateFlowFile' processor. The dialog has tabs for 'SETTINGS', 'SCHEDULING', 'PROPERTIES', and 'COMMENTS', with 'PROPERTIES' selected. A table lists properties with their values:

Property	Value
File Size	1 KB
Batch Size	1
Data Format	Text
Unique FlowFiles	false

The 'File Size' value '1 KB' is highlighted with a red box. At the bottom of the dialog are 'CANCEL' and 'APPLY' buttons.



NiFi

The screenshot shows the Apache NiFi user interface. On the left, there's a navigation sidebar with 'nifi' logo, search, and 'Operate' section containing a 'PutKafka Processor' card. The main area displays a 'Configure Processor' dialog for a 'PutKafka' processor named 'GenerateFlowFile'. The dialog has tabs for 'SETTINGS', 'SCHEDULING', 'PROPERTIES' (selected), and 'COMMENTS'. The 'PROPERTIES' tab shows a table of configuration properties:

Property	Value
Known Brokers	hdf.maas:6667
Topic Name	test
Partition Strategy	Round Robin
Partition	No value set.
Kafka Key	No value set.
Delivery Guarantee	Best Effort
Message Delimiter	Shift+enter
Max Buffer Size	5 MB
Max Record Size	1 MB
Communications Timeout	30 secs
Batch Size	16384
Queue Buffering Max Time	No value set.
Compression Codec	None
Client Name	NiFi

At the bottom of the dialog are 'CANCEL' and 'APPLY' buttons. The background shows a grid-based workspace with other processors and connections.

NiFi

The screenshot shows the Apache NiFi web interface. On the left, there's a sidebar with navigation links like 'Navigate', 'Operate', and a specific entry for 'PutKafka Processor' with ID 41da7c4c-0157-1000-0000-0000694d9c5a. Below these are various operational icons. The main area is titled 'Configure Processor' for a 'PutKafka' processor. The processor has been named 'PutKafka'. It is currently 'Enabled'. Under the 'COMMENTS' tab, there's a section for 'Auto Terminate Relationships' where the 'failure' checkbox is checked and highlighted with a red box. There are two options: 'failure', which describes routing failed flowfiles to a relationship, and 'success', which describes routing successful flowfiles to a relationship. At the bottom right of the configuration window are 'CANCEL' and 'APPLY' buttons. The status bar at the bottom of the screen shows 'NiFi Flow' and 'POWERED BY APACHE NIFI'.



Check result

```
./kafka-console-consumer.sh --zookeeper  
localhost:2181 --topic test --from-beginning
```



HDF to HDP



Install NiFi on HDP

```
sudo yum -y install git
```

```
sudo git clone https://github.com/abajwa-hw/ambari-nifi-service.git /var/lib/ambari-server/resources/stacks/HDP/2.4/services/NIFI
```

```
sudo ambari-server restart
```



Install NiFi on HDP

Ambari HDP 0 ops 0 alerts

Dashboard Services Hosts Alerts Admin admin

Actions ▾

Metrics Heatmaps Config History

Metric Actions ▾ Last 1 hour ▾

HDFS Disk Usage 48% DataNodes Live 1/1 HDFS Links NameNode Secondary NameNode 1 DataNodes More... CPU Usage No Data Available Cluster Load No Data Available NameNode Heap 5% NameNode RPC 0.20 ms NameNode CPU WIO n/a

NameNode Uptime 41.1 min ResourceManager Heap 10% ResourceManager Uptime 38.7 min NodeManagers Live 1/1 YARN Memory 0%

YARN Links ResourceManager 1 NodeManagers More...

Licensed under the Apache License, Version 2.0.
See third-party tools/resources that Ambari uses and their respective authors



Install NiFi on HDP

Ambari HDP 0 ops 0 alerts Dashboard Services Hosts Alerts Admin admin

Metrics Heatmaps Config History Metric Actions Last 1 hour

HDFS Disk Usage 48% DataNodes Live 1/1 HDFS Links NameNode Secondary NameNode 1 DataNodes More...

CPU Usage No Data Available Cluster Load No Data Available NameNode Heap 5% NameNode RPC 0.20 ms NameNode CPU WIO n/a

NameNode Uptime 41.1 min ResourceManager Heap 10% ResourceManager Uptime 38.7 min NodeManagers Live 1/1 YARN Memory 0%

Actions + Add Service Start All Stop All Restart All Required

Licensed under the Apache License, Version 2.0.
See third-party tools/resources that Ambari uses and their respective authors

hdp.maas:8080/#

The screenshot shows the Ambari HDP dashboard with various service status indicators. On the left, a sidebar lists services: HDFS, YARN, MapReduce2, Tez, Hive, Pig, ZooKeeper, Spark, Zeppelin Notebook, NiFi, and Slider. Below the sidebar is an 'Actions' dropdown menu with options: Start All, Stop All, and Restart All Required. The '+ Add Service' button is highlighted with a red box. The main dashboard area displays metrics for HDFS Disk Usage (48%), DataNodes Live (1/1), HDFS Links (NameNode, Secondary NameNode, 1 DataNodes), CPU Usage (No Data Available), Cluster Load (No Data Available), NameNode Heap (5%), NameNode RPC (0.20 ms), NameNode CPU WIO (n/a), NameNode Uptime (41.1 min), ResourceManager Heap (10%), ResourceManager Uptime (38.7 min), NodeManagers Live (1/1), and YARN Memory (0%). At the bottom, there is a note about the Apache License and a footer URL hdp.maas:8080/#.



Install NiFi on HDP

The screenshot shows the Ambari 'Add Service Wizard' interface. The top navigation bar includes 'Ambari', 'HDP', 'Dashboard', 'Services', 'Hosts', 'Alerts', 'Admin', and a user dropdown. The main window title is 'Add Service Wizard'. A sub-header indicates 'Core shared service used by Ambari managed components'. Below is a list of services:

Service Name	Version	Description
Ambari Metrics	0.1.0	A system for metrics collection that provides storage and retrieval capability for metrics collected from the cluster
Atlas	0.7.0	Atlas Metadata and Governance platform
Kafka	0.10.0	A high-throughput distributed messaging system
Knox	0.9.0	Provides a single point of authentication and access for Apache Hadoop services in a cluster
Log Search	0.5.0	Log aggregation, analysis, and visualization for Ambari managed services. This service is Technical Preview .
Ranger	0.6.0	Comprehensive security for Hadoop
Ranger KMS	0.6.0	Key Management Server
SmartSense	1.3.0.0-1	SmartSense - Hortonworks SmartSense Tool (HST) helps quickly gather configuration, metrics, logs from common HDP services that aids to quickly troubleshoot support cases and receive cluster-specific recommendations.
<input checked="" type="checkbox"/> Spark	1.6.2	Apache Spark is a fast and general engine for large-scale data processing.
<input type="checkbox"/> Spark2	2.0.0	Apache Spark 2.0 is a fast and general engine for large-scale data processing. This service is Technical Preview .
<input checked="" type="checkbox"/> Zeppelin Notebook	0.6.0	A web-based notebook that enables interactive data analytics. It enables you to make beautiful data-driven, interactive and collaborative documents with SQL, Scala and more.
<input type="checkbox"/> Mahout	0.9.0	Project of the Apache Software Foundation to produce free implementations of distributed or otherwise scalable machine learning algorithms focused primarily in the areas of collaborative filtering, clustering and classification
<input checked="" type="checkbox"/> NiFi	1.0.0-DEMO	Apache NiFi is an easy to use, powerful, and reliable system to process and distribute data. This service is for demo purposes only and not officially supported
<input checked="" type="checkbox"/> Slider	0.91.0	A framework for deploying, managing and monitoring existing distributed applications on YARN.

A 'Next →' button is located at the bottom right of the wizard window. At the very bottom of the page, there is a note: 'See third-party tools/resources that Ambari uses and their respective authors'.



Create NiFi user for HDFS

```
sudo su hdfs -c 'hadoop fs -mkdir /user/nifi'  
sudo su hdfs -c 'hadoop fs -chown nifi:hdfs  
/user/nifi'
```



NiFi to HDFS

The screenshot shows the Apache NiFi user interface. The main window displays a grid of available components: 0 inputs, 0 outputs, 0 processors, 0 controllers, and 0 sensors. The timestamp in the top right corner is 08:11:48 UTC. On the left, there's a sidebar with 'Operate' and 'NiFi Flow Process Group' sections, along with various configuration and monitoring icons. A search bar is located at the top right. The central part of the screen is occupied by a modal dialog titled 'Add Remote Process Group'. This dialog has fields for 'URL' (containing 'http://hdf.maas:9090/nifi/'), 'Transport Protocol' (set to 'RAW'), and other optional settings like 'HTTP Proxy Server Hostname', 'HTTP Proxy Server Port', 'HTTP Proxy User', 'HTTP Proxy Password', 'Communications Timeout' (set to '30 sec'), and 'Yield Duration' (set to '10 sec'). At the bottom of the dialog are 'CANCEL' and 'ADD' buttons.



NiFi to HDFS

The screenshot shows the Apache NiFi user interface. In the center, there is a card for a remote port named "http://hdf.maas:9090/nifi". The card displays statistics: Sent 0 (0 bytes) → 0 and Received 0 → 0 (0 bytes). Below the statistics, it says "No comments specified". A context menu is open over this card, with the "Enable transmission" option highlighted by a red box.

- Configure
- Remote ports
- Enable transmission**
- Disable transmission
- Status History
- Refresh
- Go to
- Center in view
- Copy
- Delete

At the bottom left of the interface, the text "NiFi Flow" is visible. At the bottom right, it says "POWERED BY APACHE NIFI" with the Apache logo.



NiFi to HDFS

The screenshot shows the Apache NiFi user interface. The top navigation bar includes icons for file operations (New, Open, Save, etc.) and a search bar. The main workspace displays a process group titled "http://hdf.maas:9090/nifi". On the left, the "Operate" sidebar shows a single remote process group named "http://hdf.maas:9090/nifi" with ID "46a6b722-0157-1000-43ae-3400bfad2ee0". The central workspace contains a single node labeled "http://hdf.maas:9090/nifi". A context menu is open over this node, listing options: "View configuration", "Remote ports" (which is highlighted with a red box), "Enable transmission", "Disable transmission", "Status History", "Refresh", "Go to", "Center in view", and "Copy". The bottom status bar indicates "NiFi Flow" and "POWERED BY APACHE NIFI".



NiFi to HDFS

The screenshot shows the Apache NiFi user interface. A central modal dialog is open, titled "Remote Process Group Ports". The dialog displays configuration details for a port named "http://hdf.maas:9090/nifi".
Input ports: Name - http://hdf.maas:9090/nifi
Output ports:

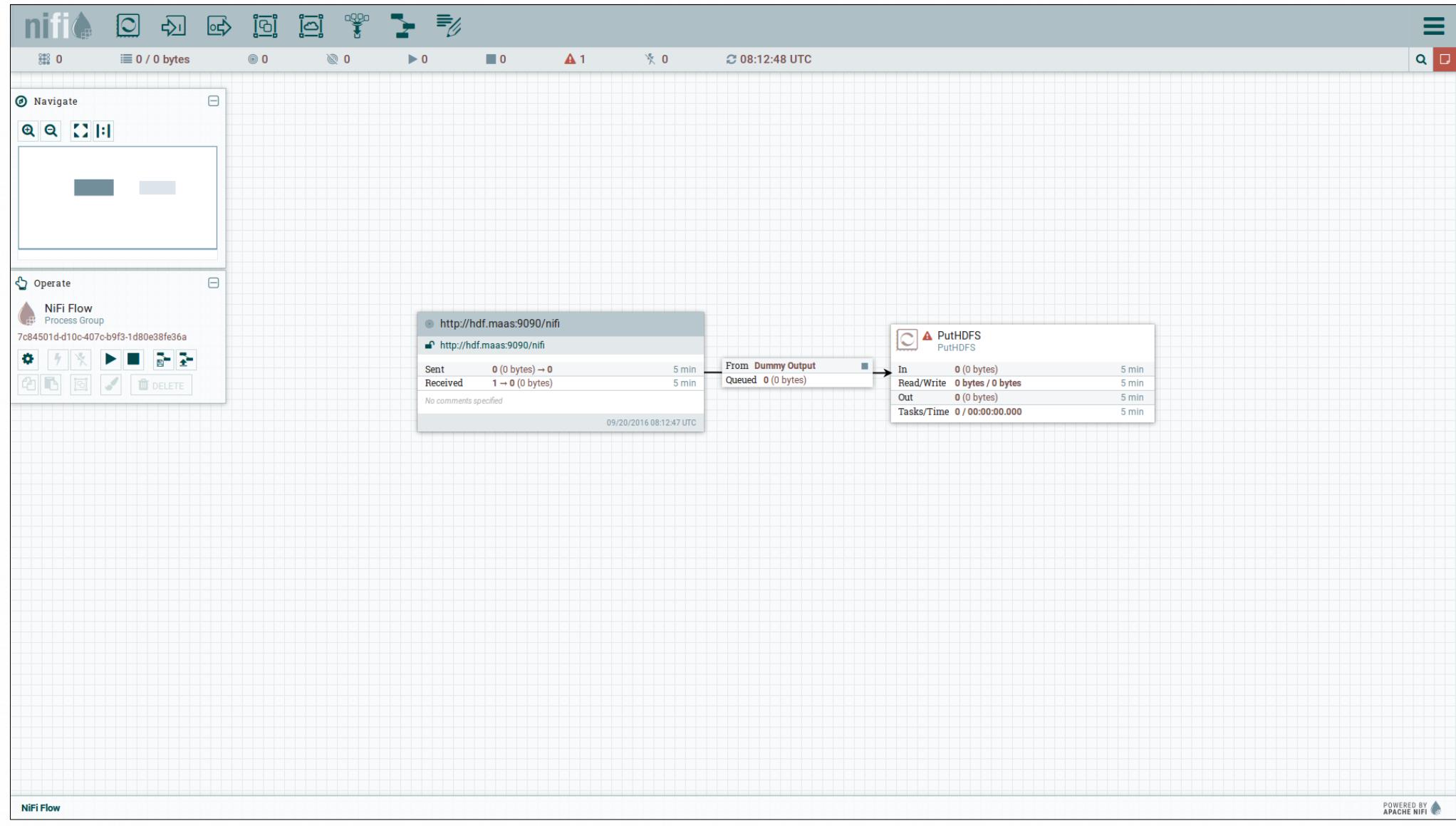
- Dummy Output (selected, highlighted with a red box)
- No description specified.
- Concurrent Tasks: 1
- Compressed: No

At the bottom right of the modal is a "CLOSE" button.

The background shows the main NiFi interface with various components and a sidebar labeled "Operate".



NiFi to HDFS





NiFi to HDFS

The screenshot shows the Apache NiFi user interface. On the left, the 'Operate' sidebar displays a 'PutHDFS Processor' with the identifier '46b58b93-0157-1000-c5d5-067ce26a7060'. Below it are standard file operations like Open, Save, Copy, Paste, and Delete. The main workspace is titled 'Configure Processor' and shows the 'PROPERTIES' tab selected. A modal dialog is open over the properties table, specifically for the 'Hadoop Configuration Resources' property. This dialog contains a single input field with the value '/etc/hadoop/conf/core-site.xml', which is highlighted with a red rectangle. Below the input field is a checkbox labeled 'Set empty string'. At the bottom of the dialog are 'CANCEL' and 'OK' buttons. In the background, the properties table lists various HDFS configuration options such as Block Size, IO Buffer Size, Replication, Permissions umask, Remote Owner, Remote Group, and Compression codec, all currently set to 'No value set' or 'NONE'. The bottom right of the dialog has 'CANCEL' and 'APPLY' buttons.



NiFi to HDFS

The screenshot shows the Apache NiFi user interface. On the left, the sidebar displays the NiFi Flow with a single active processor, a PutHDFS Processor. The main area is titled "Configure Processor" for this specific instance. The processor has been configured with the following properties:

Property	Value
Hadoop Configuration Resources	/etc/hadoop/conf/core-site.xml
Kerberos Principal	No value set
Kerberos Keytab	No value set
Kerberos Relogin Period	5 min
Directory	test/
Conflict Resolution Strategy	Set empty string
Block Size	5 min
IO Buffer Size	5 min
Replication	5 min
Permissions umask	5 min
Remote Owner	5 min
Remote Group	5 min
Compression codec	NONE

A modal dialog box is open over the configuration screen, specifically for the "Directory" property. This dialog contains a text input field with the value "test/" and a checkbox labeled "Set empty string". The "OK" button is highlighted with a red border.

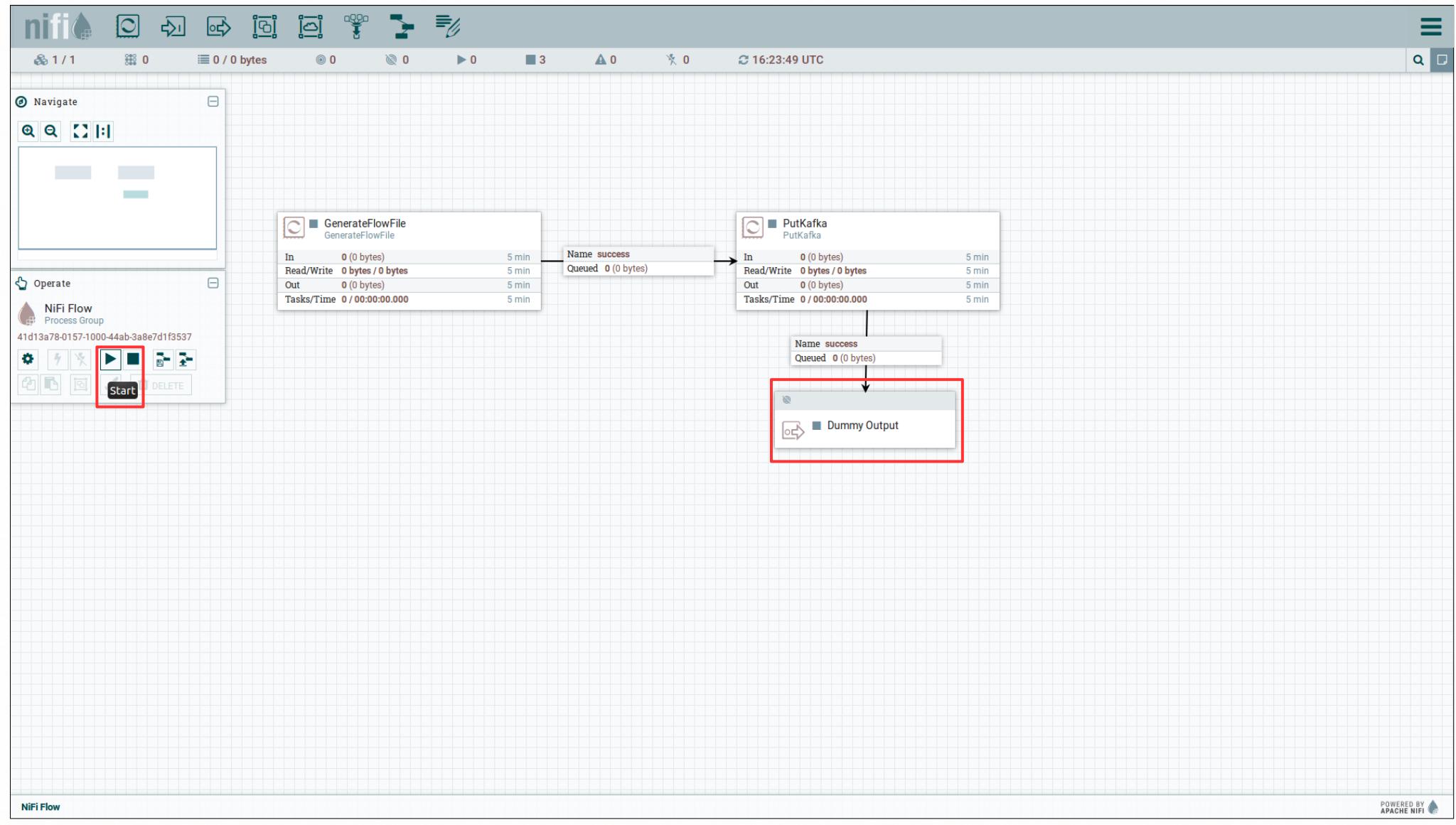


NiFi to HDFS

The screenshot shows the Apache NiFi user interface. On the left, there's a sidebar with various icons and a list of processors. A specific processor, "PutHDFS Processor" (ID: 46b58b93-0157-1000-c5d5-067ce26a7060), is selected and highlighted with a red box. The main panel displays the "Configure Processor" dialog for this selected processor. The dialog has four tabs: SETTINGS (selected), SCHEDULING, PROPERTIES, and COMMENTS. In the SETTINGS tab, the processor is named "PutHDFS", is marked as "Enabled", and has an "Id" of "46b58b93-0157-1000-c5d5-067ce26a7060". It is of type "PutHDFS" and has a "Penalty Duration" of "30 sec" and a "Yield Duration" of "1 sec". The "Bulletin Level" is set to "WARN". In the COMMENTS tab, there is a section titled "Auto Terminate Relationships" which is currently disabled. There are two checkboxes: "failure" (which is checked) and "success" (which is also checked). Both checkboxes have descriptive text below them: "failure" says "Files that could not be written to HDFS for some reason are transferred to this relationship" and "success" says "Files that have been successfully written to HDFS are transferred to this relationship". At the bottom of the dialog are "CANCEL" and "APPLY" buttons.

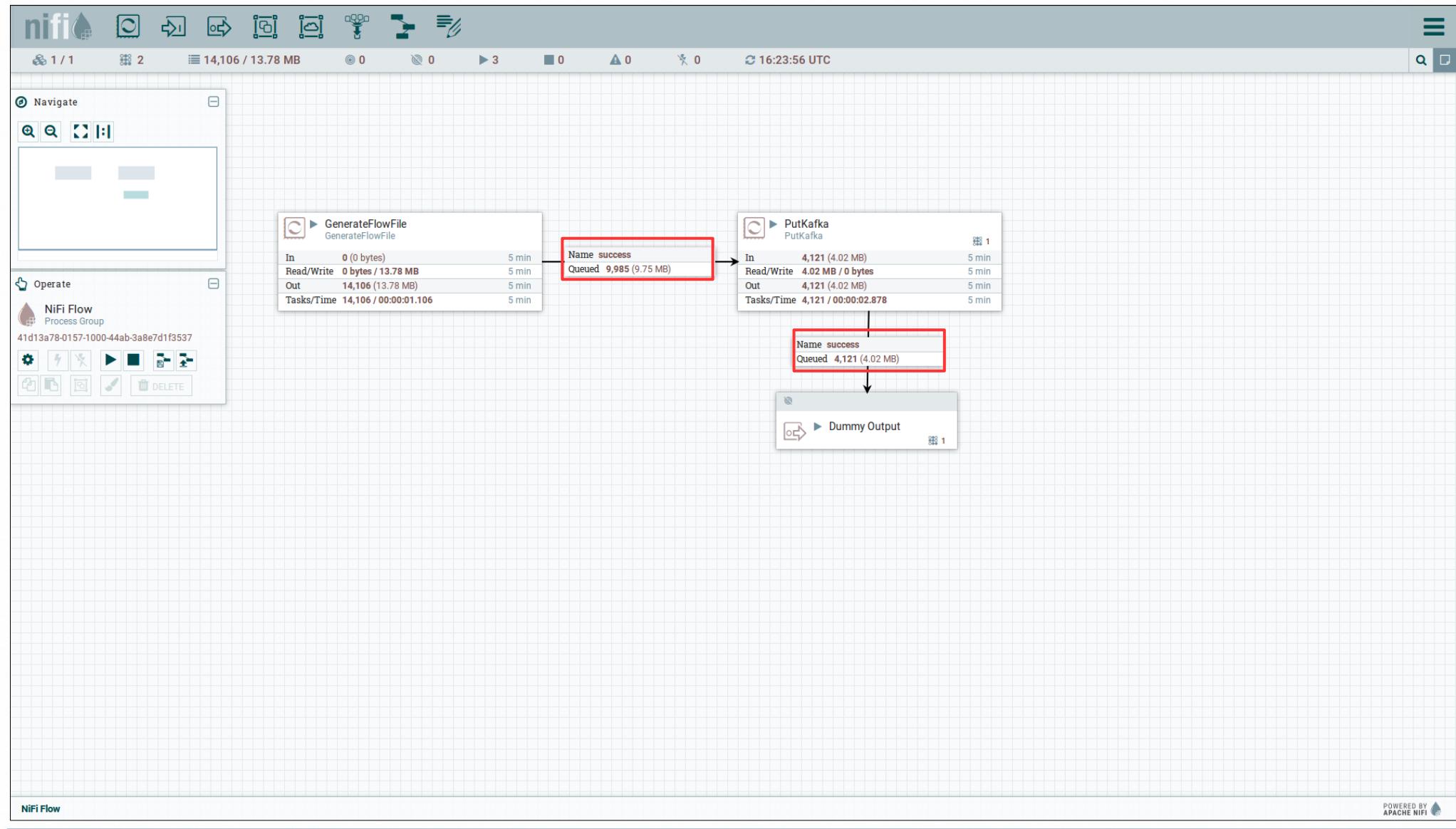


NiFi to HDFS



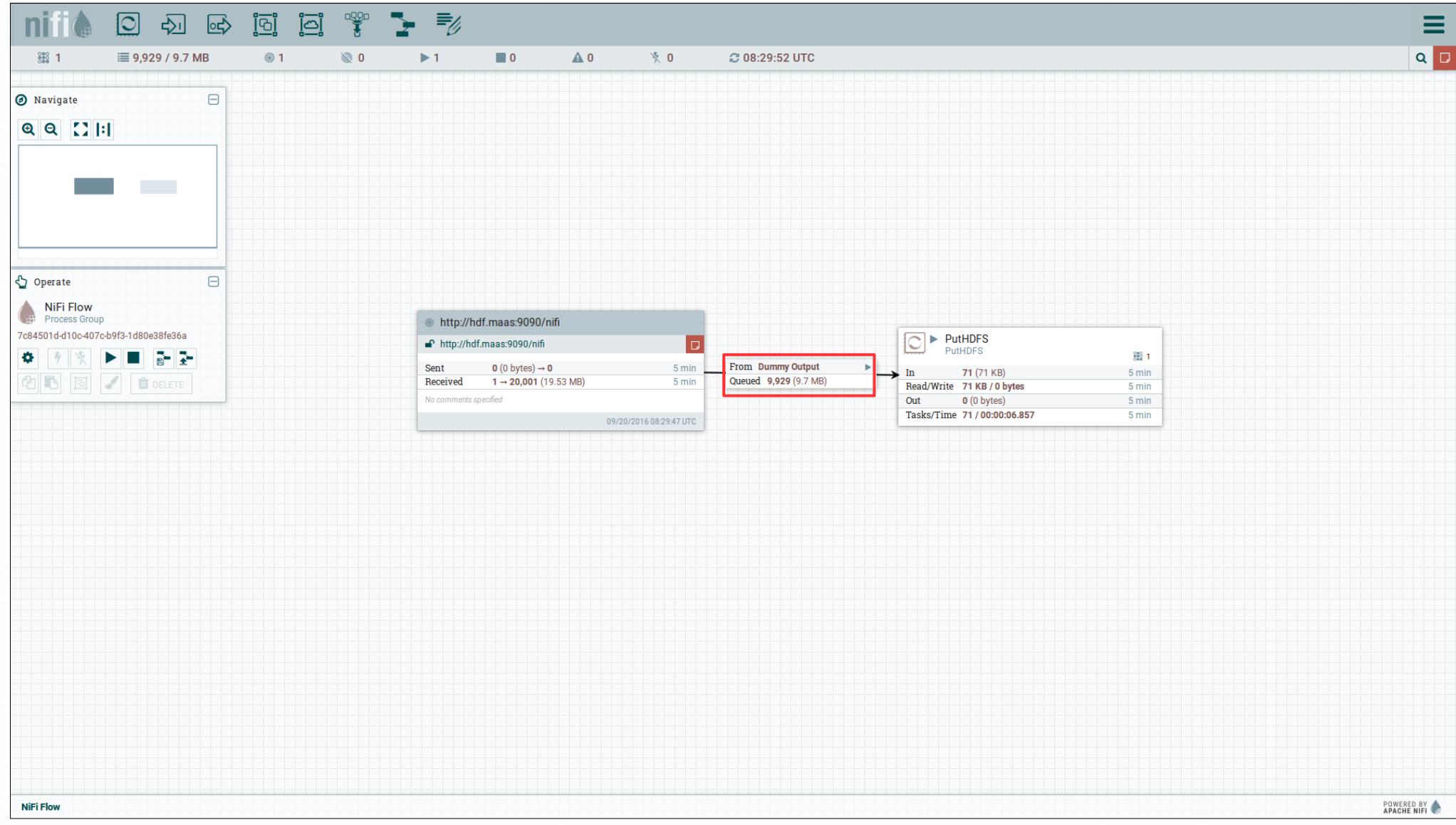


NiFi to HDFS





NiFi to HDFS





NiFi to HDFS

Hadoop Overview Datanodes Datanode Volume Failures Snapshot Startup Progress Utilities ▾

Browse Directory

/user/nifi/test

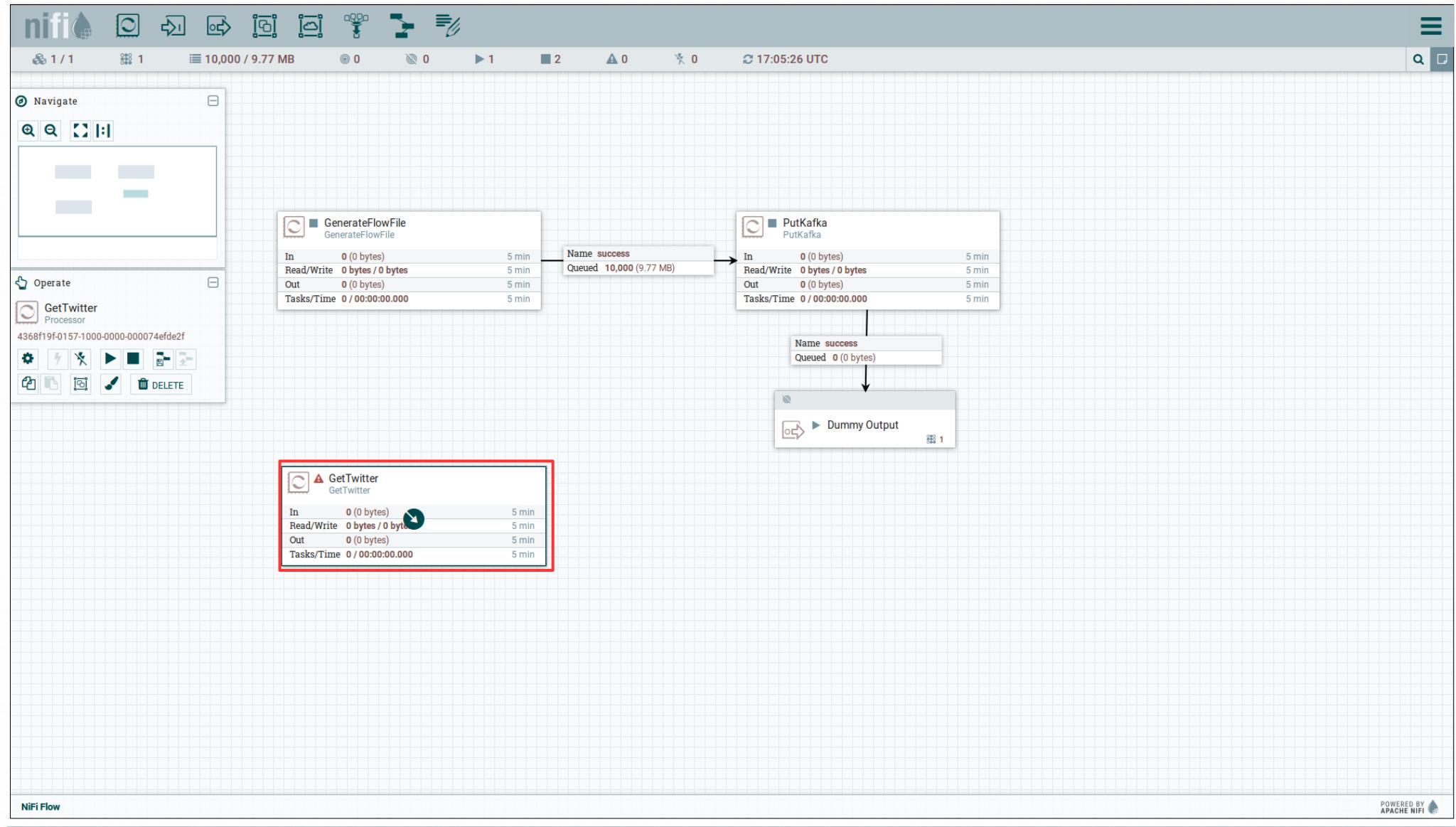
Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
-rW-r--r--	nifi	hdfs	0 B	2016/9/20 下午4:30:28	3	128 MB	.25893215913542
-rW-r--r--	nifi	hdfs	1 KB	2016/9/20 下午4:29:46	3	128 MB	25893126264585
-rW-r--r--	nifi	hdfs	1 KB	2016/9/20 下午4:29:46	3	128 MB	25893126472345
-rW-r--r--	nifi	hdfs	1 KB	2016/9/20 下午4:29:46	3	128 MB	25893126647523
-rW-r--r--	nifi	hdfs	1 KB	2016/9/20 下午4:29:47	3	128 MB	25893126812560
-rW-r--r--	nifi	hdfs	1 KB	2016/9/20 下午4:29:47	3	128 MB	25893127043208
-rW-r--r--	nifi	hdfs	1 KB	2016/9/20 下午4:29:47	3	128 MB	25893127264449
-rW-r--r--	nifi	hdfs	1 KB	2016/9/20 下午4:29:47	3	128 MB	25893127430802
-rW-r--r--	nifi	hdfs	1 KB	2016/9/20 下午4:29:47	3	128 MB	25893127605095
-rW-r--r--	nifi	hdfs	1 KB	2016/9/20 下午4:29:47	3	128 MB	25893127877337
-rW-r--r--	nifi	hdfs	1 KB	2016/9/20 下午4:29:47	3	128 MB	25893128050828
-rW-r--r--	nifi	hdfs	1 KB	2016/9/20 下午4:29:47	3	128 MB	25893128226566
-rW-r--r--	nifi	hdfs	1 KB	2016/9/20 下午4:29:47	3	128 MB	25893128430393
-rW-r--r--	nifi	hdfs	1 KB	2016/9/20 下午4:29:47	3	128 MB	25893128601091
-rW-r--r--	nifi	hdfs	1 KB	2016/9/20 下午4:29:48	3	128 MB	25893128796478
-rW-r--r--	nifi	hdfs	1 KB	2016/9/20 下午4:29:48	3	128 MB	25893128935796
-rW-r--r--	nifi	hdfs	1 KB	2016/9/20 下午4:29:48	3	128 MB	25893129231018
-rW-r--r--	nifi	hdfs	1 KB	2016/9/20 下午4:29:48	3	128 MB	25893129417335
-rW-r--r--	nifi	hdfs	1 KB	2016/9/20 下午4:29:48	3	128 MB	25893129620044
-rW-r--r--	nifi	hdfs	1 KB	2016/9/20 下午4:29:48	3	128 MB	25893129768489
-rW-r--r--	nifi	hdfs	1 KB	2016/9/20 下午4:29:48	3	128 MB	25893130030348
-rW-r--r--	nifi	hdfs	1 KB	2016/9/20 下午4:29:48	3	128 MB	25893130221410



Let's Tweet



Twitter to HDF





Twitter to HDF

Screenshot of the Apache NiFi interface showing the configuration of a Twitter processor.

The main window displays two processors:

- GenerateFlowFile**: A blue square icon. Statistics: In 0 (0 bytes), Read/Write 0 bytes / 0 bytes, Out 0 (0 bytes), Tasks/Time 0 / 00:00:00.000.
- GetTwitter**: An orange triangle icon. Statistics: In 0 (0 bytes), Read/Write 0 bytes / 0 bytes, Out 0 (0 bytes), Tasks/Time 0 / 00:00:00.000.

A modal dialog titled "Configure Processor" is open, specifically for the "GetTwitter" processor. The "SETTINGS" tab is selected. A sub-modal dialog titled "Filter Endpoint" is displayed over the main configuration area, with the "OK" button highlighted by a red box.

The "Properties" table in the main configuration area includes the following entries:

Property	Value
Twitter Endpoint	Filter Endpoint
Consumer Key	No value set
Consumer Secret	No value set
Access Token	No value set
Access Token Secret	No value set
Languages	No value set
Terms to Filter On	No value set
IDs to Follow	No value set
Locations to Filter On	No value set



Twitter to HDF

NiFi Flow interface showing a Twitter to HDF pipeline.

The pipeline consists of two main components:

- GetTwitter Processor**: Fetches data from Twitter. Its configuration shows:
 - Twitter Endpoint: Filter Endpoint (redacted)
 - Consumer Key: UeNDek54HsVf1u9vN4xalLmIJ
 - Consumer Secret: Sensitive value set
 - Access Token: 584996438-YEe4MxKAsx0dVWlsSLz080Ydid7el...
 - Access Token Secret: Sensitive value set
 - Languages: No value set
 - Terms to Filter On: No value set
 - IDs to Follow: No value set
 - Locations to Filter On: No value set
- GenerateFlowFile Processor**: Converts Twitter data into flow files. Its configuration shows:
 - In: 0 (0 bytes)
 - Read/Write: 0 bytes / 0 bytes
 - Out: 0 (0 bytes)
 - Tasks/Time: 0 / 00:00:00.000

The central window is the **Configure Processor** dialog for the selected **GetTwitter** processor. It displays the **PROPERTIES** tab with the following table:

Property	Value
Twitter Endpoint	Filter Endpoint
Consumer Key	UeNDek54HsVf1u9vN4xalLmIJ
Consumer Secret	Sensitive value set
Access Token	584996438-YEe4MxKAsx0dVWlsSLz080Ydid7el...
Access Token Secret	Sensitive value set
Languages	No value set
Terms to Filter On	No value set
IDs to Follow	No value set
Locations to Filter On	No value set

Buttons at the bottom of the dialog are **CANCEL** and **APPLY**.



Twitter to HDF

The screenshot shows the Apache NiFi user interface. On the left, there's a sidebar with navigation links like 'Navigate', 'Operate', and a specific node named 'GetTwitter Processor'. The main workspace contains two nodes: a 'GenerateFlowFile' node at the top and a 'GetTwitter' node below it. A flow path connects them. A context menu is open over the 'GetTwitter' node, with the 'Configure' option selected, which has triggered the 'Configure Processor' dialog.

Configure Processor

The dialog has four tabs: SETTINGS (selected), SCHEDULING, PROPERTIES, and COMMENTS.

Required field

Property	Value
Twitter Endpoint	Filter Endpoint
Consumer Key	UeNDek54HsVf1u9vN4xalLmIJ
Consumer Secret	Sensitive value set
Access Token	584996438-YEe4MxKASx0dVWlslz08OYdid7el...
Access Token Secret	Sensitive value set

Properties

- Languages: google, apple, ~~facebook~~, amazon
- Terms to Filter On: (highlighted in yellow)
- IDs to Follow:
- Locations to Filter On:

Buttons

- Set empty string
- CANCEL
- OK
- CANCEL
- APPLY



Twitter to HDF

NiFi Flow

POWERED BY APACHE NIFI

The screenshot shows the Apache NiFi user interface. A central modal window titled "Configure Processor" is open, specifically for a "PutKafka" processor. The "PROPERTIES" tab is selected. In the "Required field" section, the "Topic Name" property is listed with a value of "tweet". This value is highlighted with a red rectangular box. Below the topic name, there is a checkbox labeled "Set empty string". At the bottom of the modal are "CANCEL" and "OK" buttons. The background of the NiFi interface shows other processors like "GenerateFlowFile" and "GetTwitter", along with their respective status metrics.

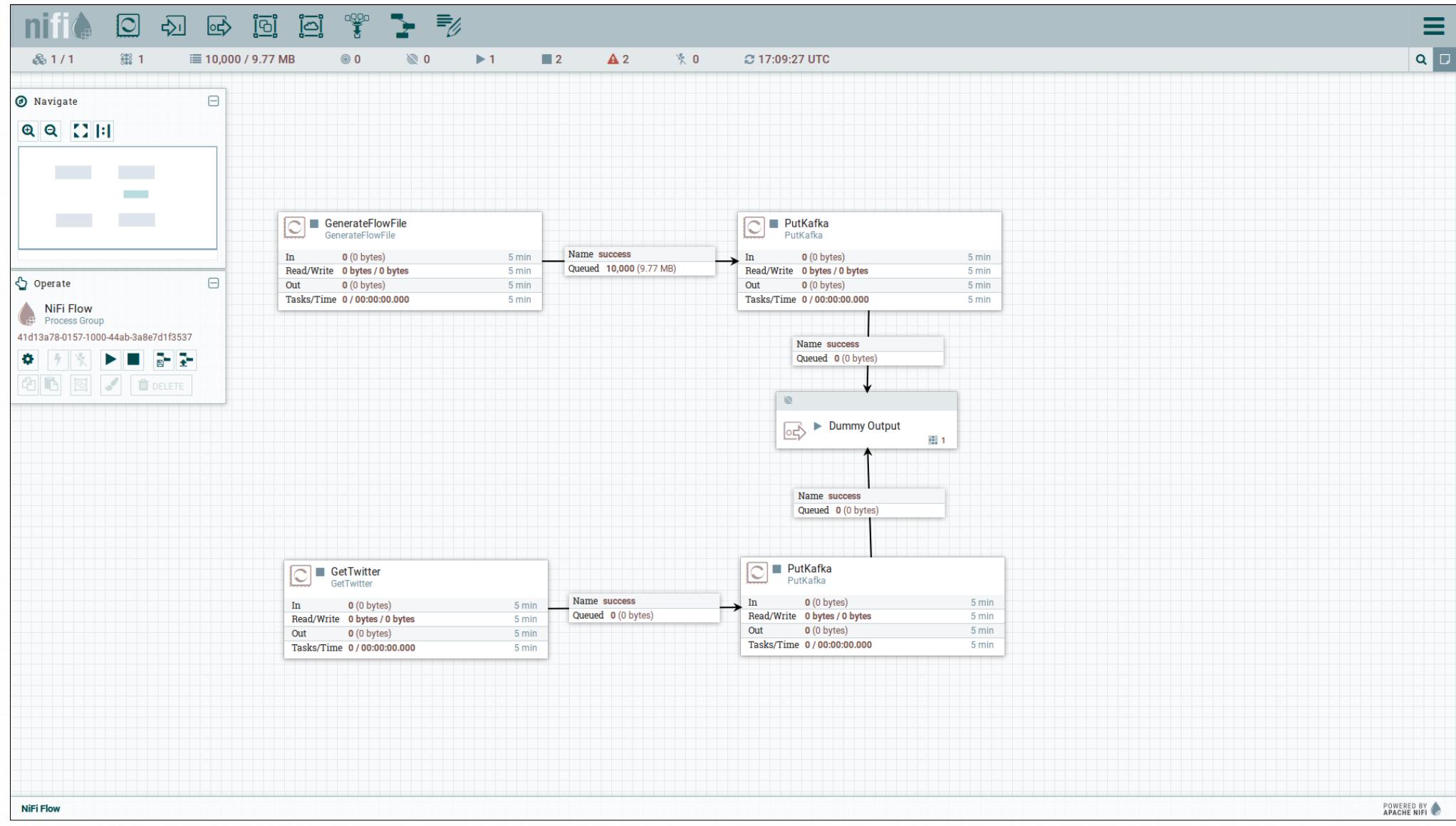


Twitter to HDF

```
cd /usr/hdf/2.0.0.0-579/kafka/bin/  
./kafka-topics.sh --create --zookeeper  
localhost:2181 --replication-factor 1 --partitions  
2 --topic tweet
```



Twitter to HDF





Check result on Kafka

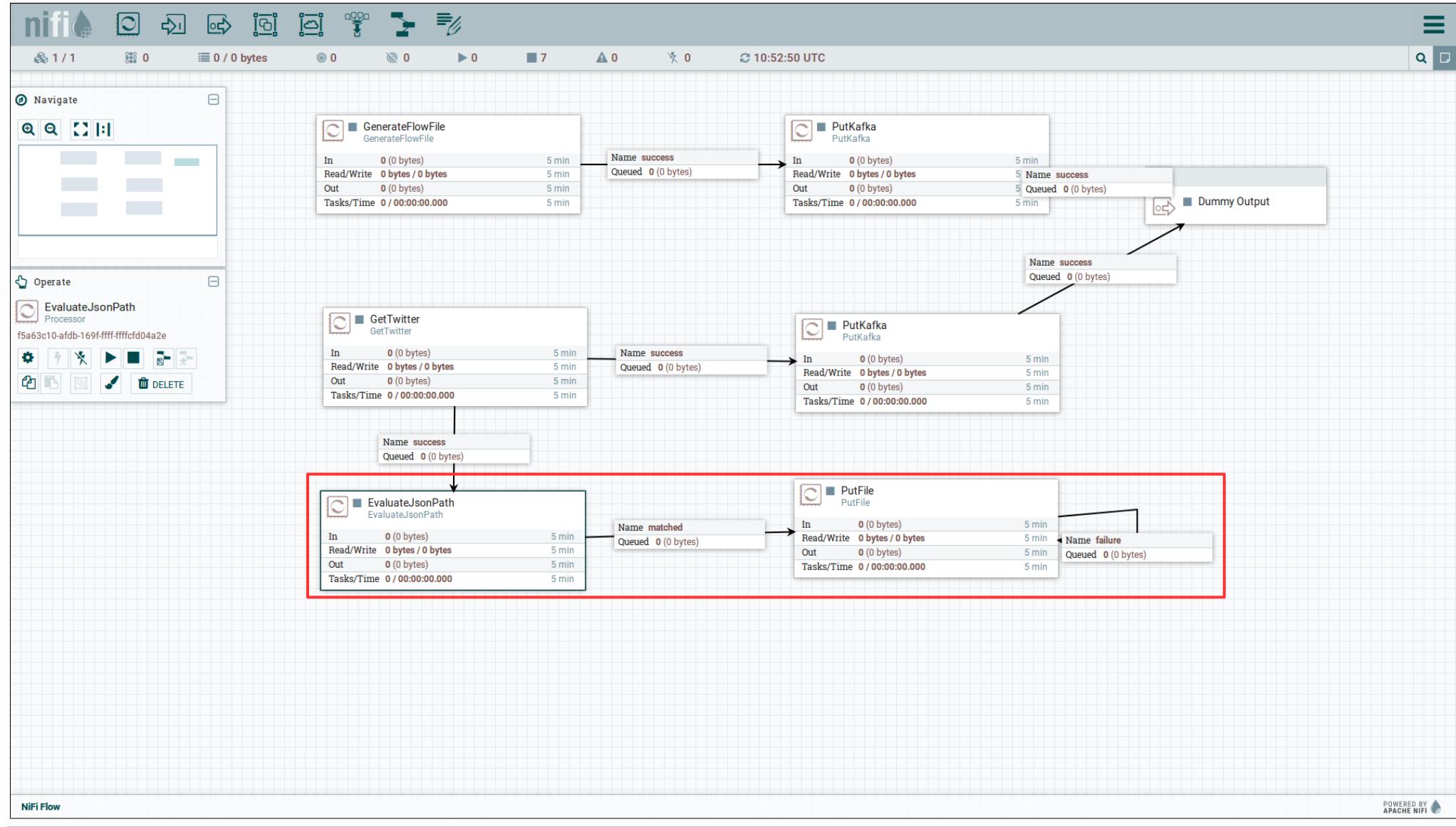
```
./kafka-console-consumer.sh --zookeeper  
localhost:2181 --topic tweet --from-beginning
```



Pre-Process

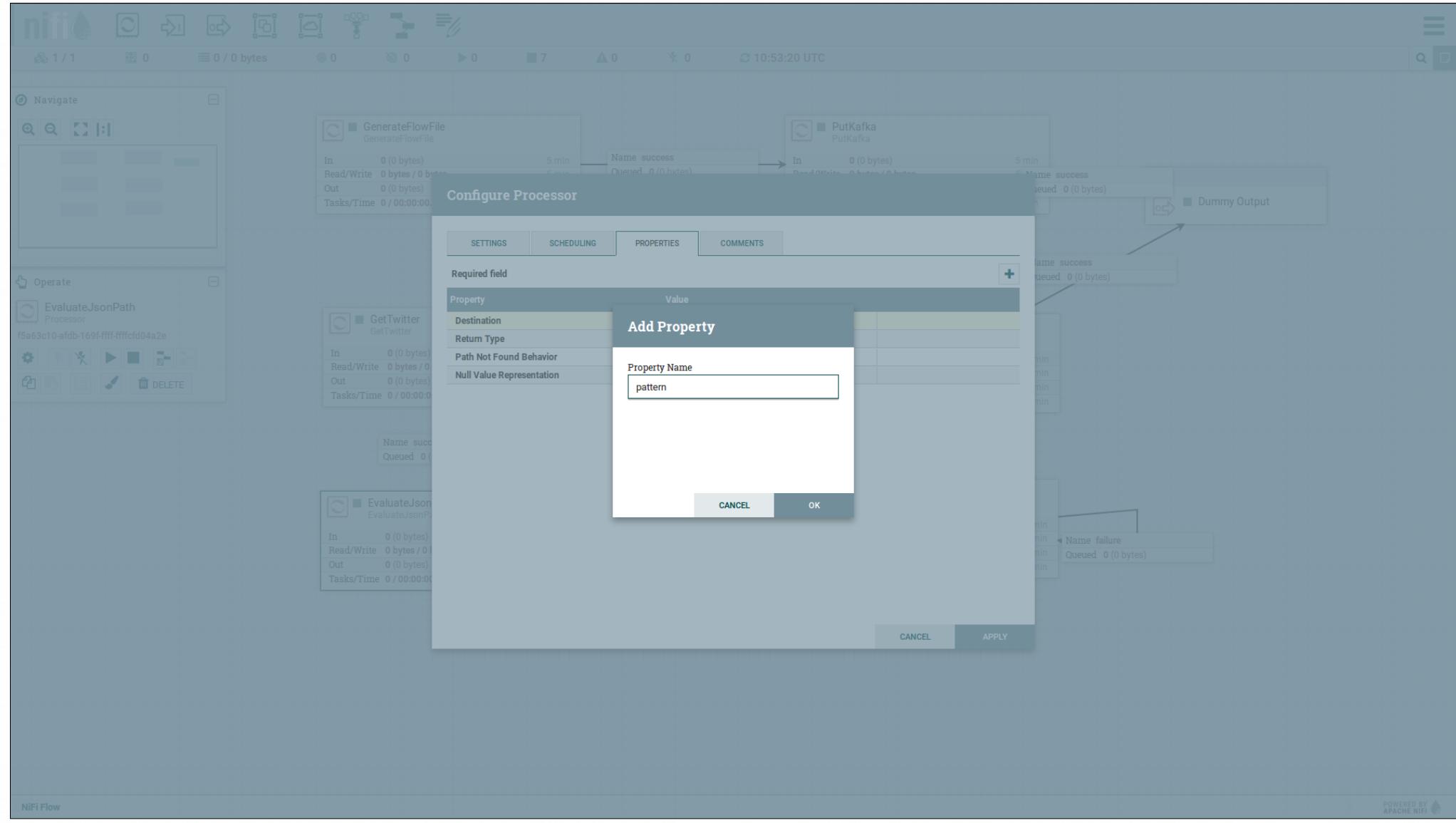


Content extraction



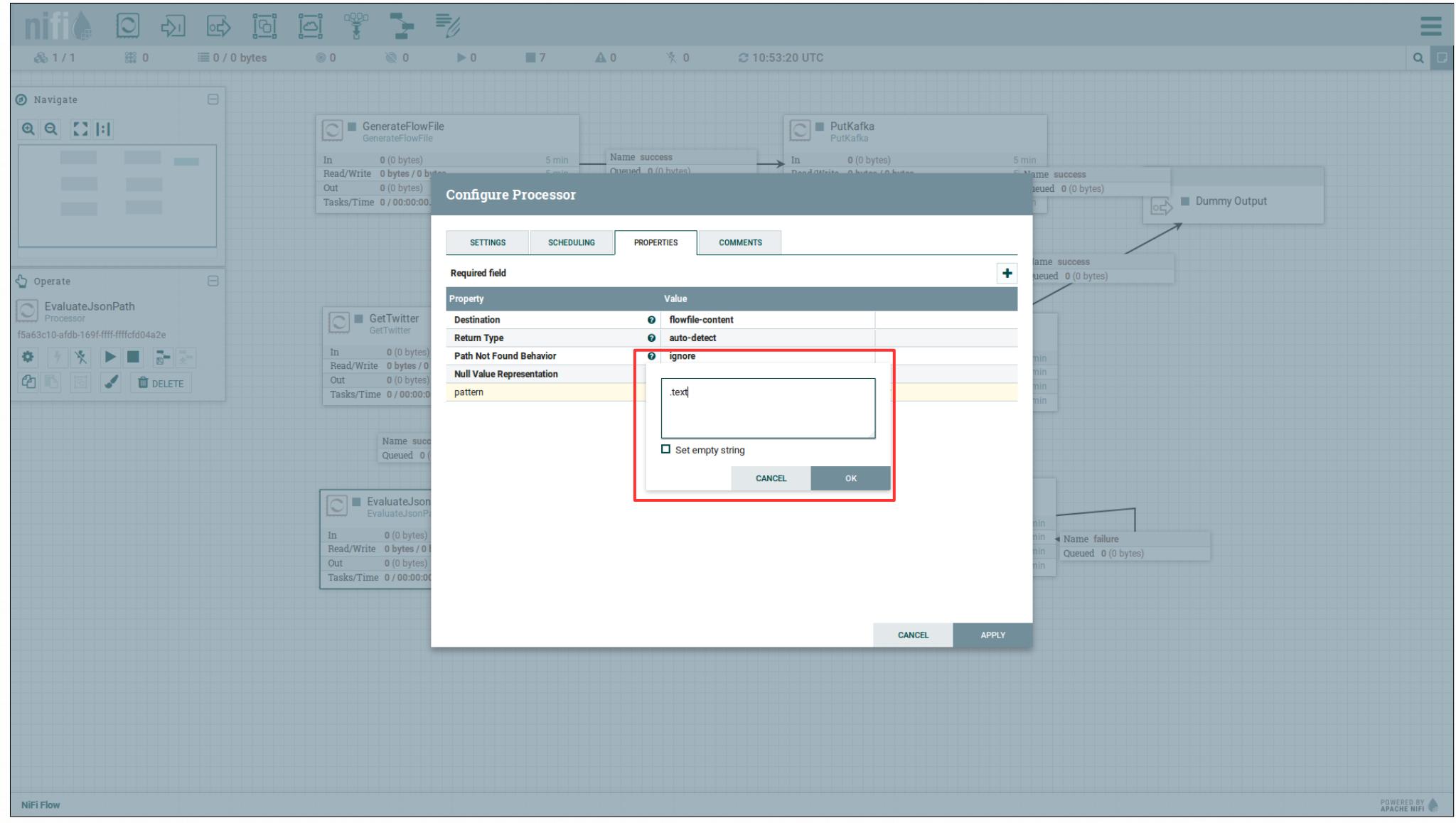


Content extraction



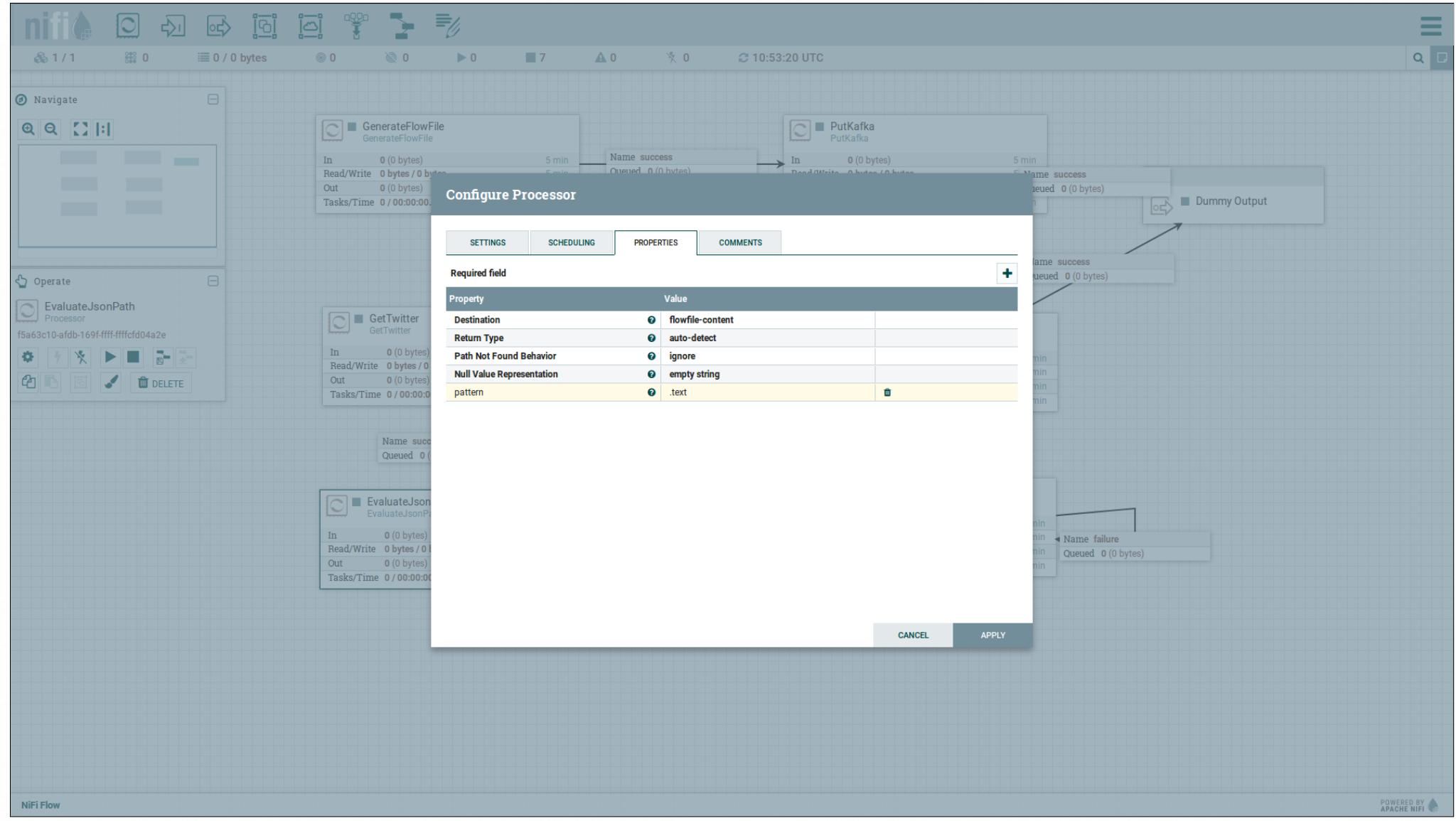


Content extraction





Content extraction





Content extraction

The screenshot shows the Apache NiFi user interface with a flow diagram and a detailed configuration dialog for an EvaluateJsonPath processor.

Flow Diagram: The main workspace shows a flow starting with a "GenerateFlowFile" processor, followed by a "Name success" relationship leading to a "PutKafka" processor. A "GetTwitter" processor is connected to an "EvaluateJsonPath" processor, which has three outgoing relationships: "failure" (red box), "matched" (green box), and "unmatched" (blue box). The "failure" relationship leads to a "Name failure" processor. The "matched" and "unmatched" relationships lead to a "Dummy Output" processor.

Processor Configuration Dialog: A modal window titled "Configure Processor" is open for the "EvaluateJsonPath" processor.

- SETTINGS Tab:** Shows the processor's name ("EvaluateJsonPath"), which is enabled, and its ID ("f5a63c10-afdb-169f-ffff-ffffcf04a2e").
- SCHEDULING Tab:** Shows "Penalty Duration" set to "30 sec" and "Yield Duration" set to "1 sec".
- PROPERTIES Tab:** Shows the "Bulletin Level" set to "WARN".
- COMMENTS Tab:** Contains a section titled "Auto Terminate Relationships" with three options:
 - failure**: Checked. Description: FlowFiles are routed to this relationship when the JsonPath cannot be evaluated against the content of the FlowFile; for instance, if the FlowFile is not valid JSON.
 - matched**: Unchecked. Description: FlowFiles are routed to this relationship when the JsonPath is successfully evaluated and the FlowFile is modified as a result.
 - unmatched**: Checked. Description: FlowFiles are routed to this relationship when the JsonPath does not match the content of the FlowFile and the Destination is set to flowfile-content.

Buttons at the bottom: "CANCEL" and "APPLY".



Content extraction

The screenshot shows the Apache NiFi user interface with a flow defined for content extraction. The flow consists of the following components and connections:

- GenerateFlowFile** → **PutKafka** → **Dummy Output**
- GetTwitter** → **EvaluateJsonPath** → **PutKafka** → **Dummy Output**
- PutKafka** has three outgoing paths:
 - To **Dummy Output**
 - To a **Name success** placeholder
 - To a **Name failure** placeholder

A modal dialog titled "Configure Processor" is open over the **PutKafka** component, specifically for the "Properties" tab. A red box highlights the "Directory" property field, which contains the value `/home/nifi/tweet_text/`. Other properties listed include **Conflict Resolution Strategy**, **Create Missing Directories**, **Maximum File Count**, **Last Modified Time**, **Permissions**, **Owner**, and **Group**. The "OK" button is visible at the bottom right of the dialog.



Content extraction

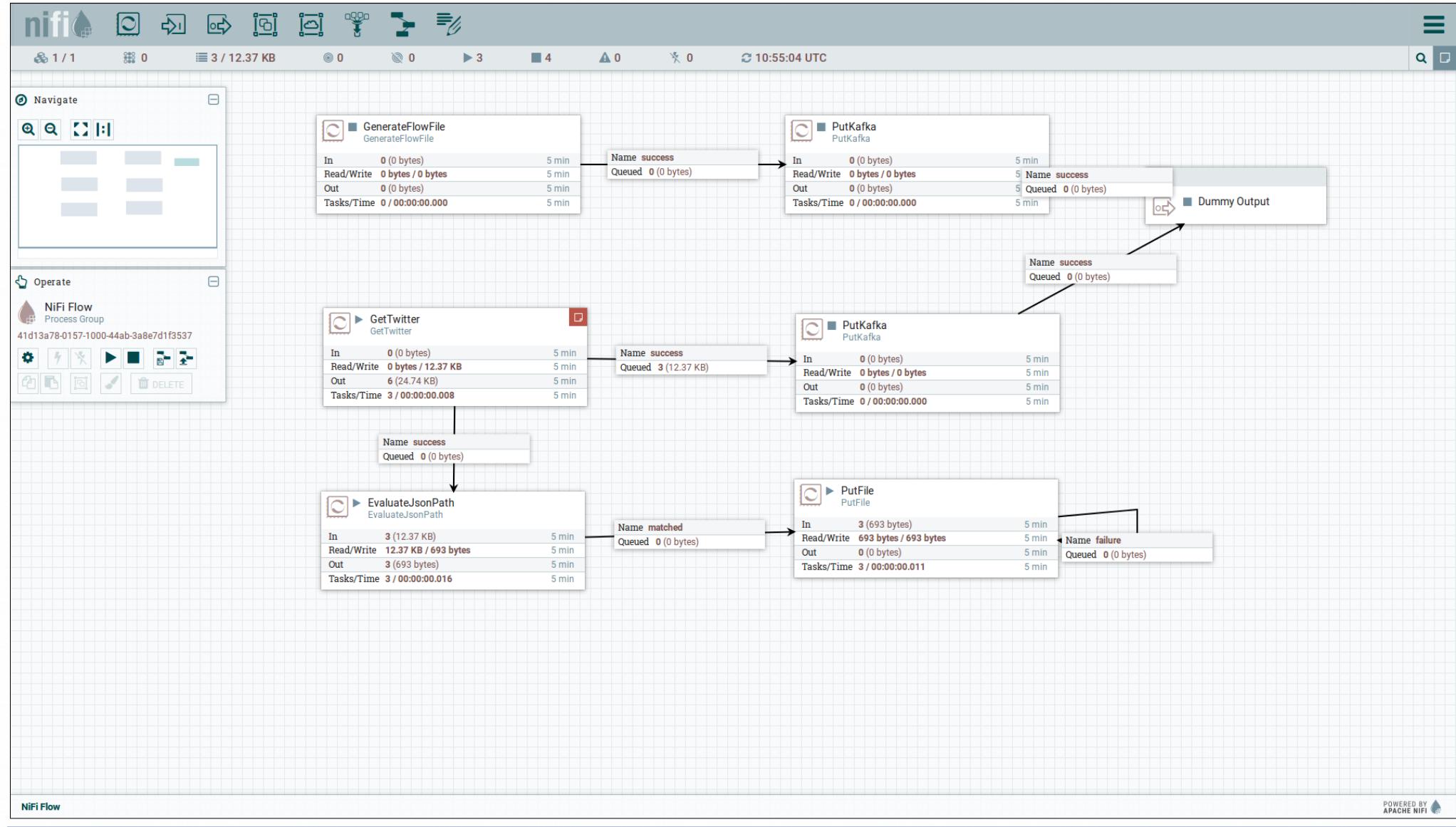
The screenshot shows the Apache NiFi user interface with a flow defined for content extraction. The flow consists of the following components and connections:

- GenerateFlowFile** → **PutKafka**: A relationship named "success" connects these two components.
- PutKafka** → **Dummy Output**: A relationship named "success" connects these two components.
- GetTwitter** → **EvaluateJsonPath**: A relationship named "success" connects these two components.
- EvaluateJsonPath** → **PutFile**: A relationship named "success" connects these two components.
- PutFile** → **Dummy Output**: A relationship named "failure" connects these two components.

A modal dialog titled "Configure Processor" is open over the flow, specifically for the **PutFile** processor. The "SETTINGS" tab is selected. In the "Comments" section, the "Auto Terminate Relationships" checkbox is checked. The "failure" and "success" options are highlighted with a red border. The "failure" option is described as "Files that could not be written to the output directory for some reason are transferred to this relationship". The "success" option is described as "Files that have been successfully written to the output directory are transferred to this relationship".



Start Content extraction





Check result in terminal

The screenshot shows a terminal window with a dark background and light-colored text. The title bar reads "nifi@HDF:~/tweet_text". The menu bar includes "File Edit View Search Terminal Help". The terminal prompt is "[centos@HDF ~]\$". The user runs several commands:
1. "sudo su - nifi" to switch to the nifi user.
2. "ls" to list files in the current directory.
3. "cd tweet_text/" to change directory to "tweet_text".
4. "ls" again to list files in "tweet_text".
The output shows four JSON files: "34825229606727.json", "34831236481883.json", "34828233992880.json", and "34834239411248.json". The terminal prompt "[nifi@HDF tweet_text]\$" is followed by a small square icon.



ExecuteProcess

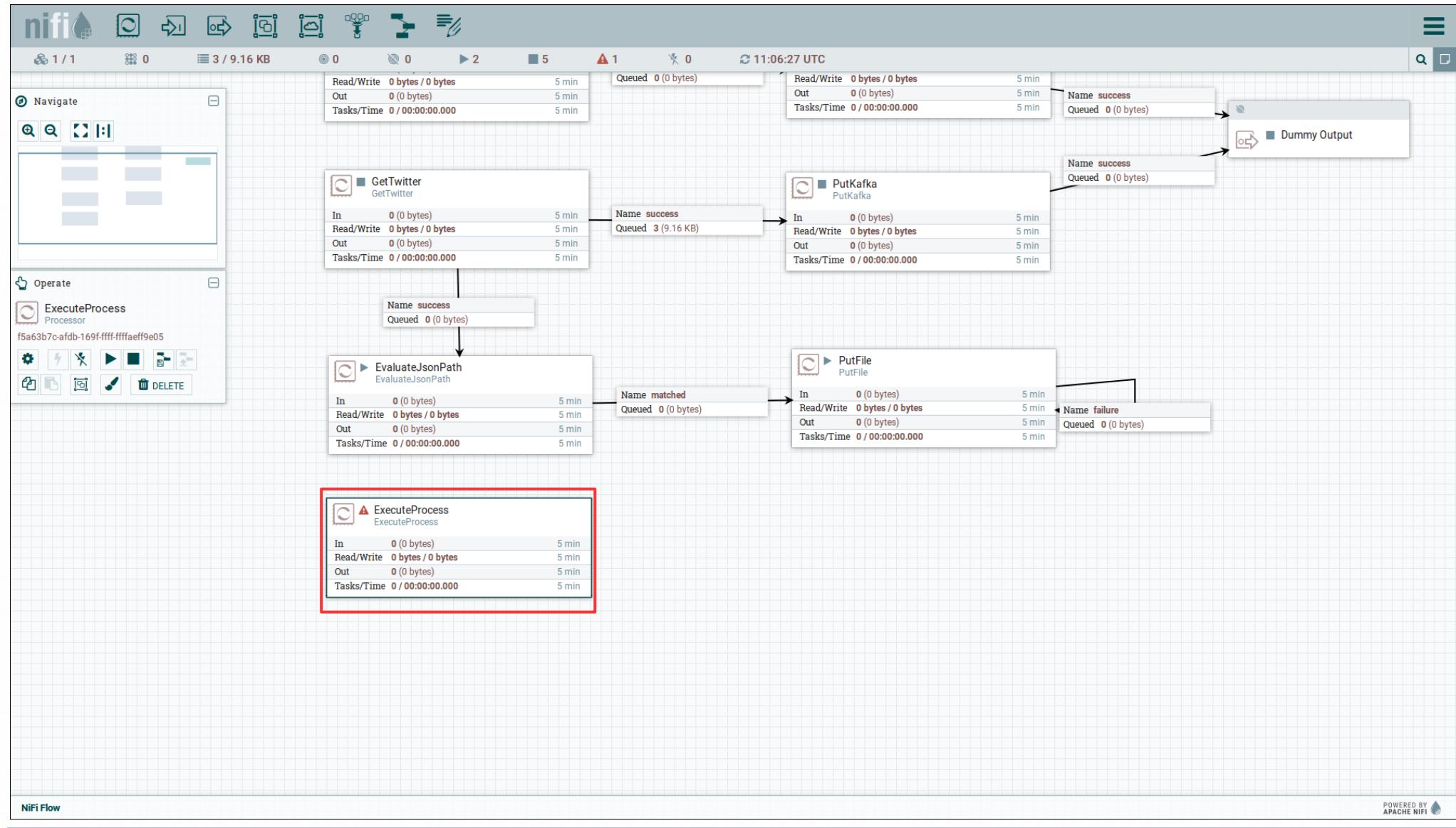


Create process

```
vi random_number.py
import random
print '%.Of' % random.uniform(1, 10) + ' ' +
str(random.uniform(100, 1000))
```



Create process





Create process

The screenshot shows the Apache NiFi user interface. On the left, the 'NiFi Flow' panel displays a process flow with several components: GetTwitter, EvaluateJsonPath, PutKafka, and ExecuteProcess. The 'PutKafka' component has a status card indicating 0 bytes written, 0 tasks, and 5 min processing time. The 'ExecuteProcess' component has a status card indicating 0 bytes written, 0 tasks, and 5 min processing time. The 'EvaluateJsonPath' component has a status card indicating 0 bytes written, 0 tasks, and 5 min processing time. The 'GetTwitter' component has a status card indicating 0 bytes written, 0 tasks, and 5 min processing time.

In the center, a 'Configure Processor' dialog box is open for the 'ExecuteProcess' processor. The dialog has tabs for SETTINGS, SCHEDULING, PROPERTIES, and COMMENTS. The PROPERTIES tab is selected, showing a 'Required field' section. Under the 'Property' column, the 'Command' field is highlighted and contains the value 'python'. A red box highlights this input field. Below it, there is a checkbox labeled 'Set empty string' which is unchecked. At the bottom of the dialog are 'CANCEL' and 'OK' buttons, with 'OK' being the active button.

On the right side of the screen, the main NiFi interface shows the flow: GetTwitter → EvaluateJsonPath → ExecuteProcess → PutKafka. The 'ExecuteProcess' step is currently active, indicated by a yellow border. The 'PutKafka' step is also highlighted with a yellow border. The flow starts with 'GetTwitter' (green icon), followed by 'EvaluateJsonPath' (blue icon), then 'ExecuteProcess' (yellow icon), and finally 'PutKafka' (green icon). Arrows indicate the data flow from one component to the next. The 'PutKafka' component has a status card indicating 0 bytes written, 0 tasks, and 5 min processing time. The 'ExecuteProcess' component has a status card indicating 0 bytes written, 0 tasks, and 5 min processing time. The 'EvaluateJsonPath' component has a status card indicating 0 bytes written, 0 tasks, and 5 min processing time. The 'GetTwitter' component has a status card indicating 0 bytes written, 0 tasks, and 5 min processing time.



Create process

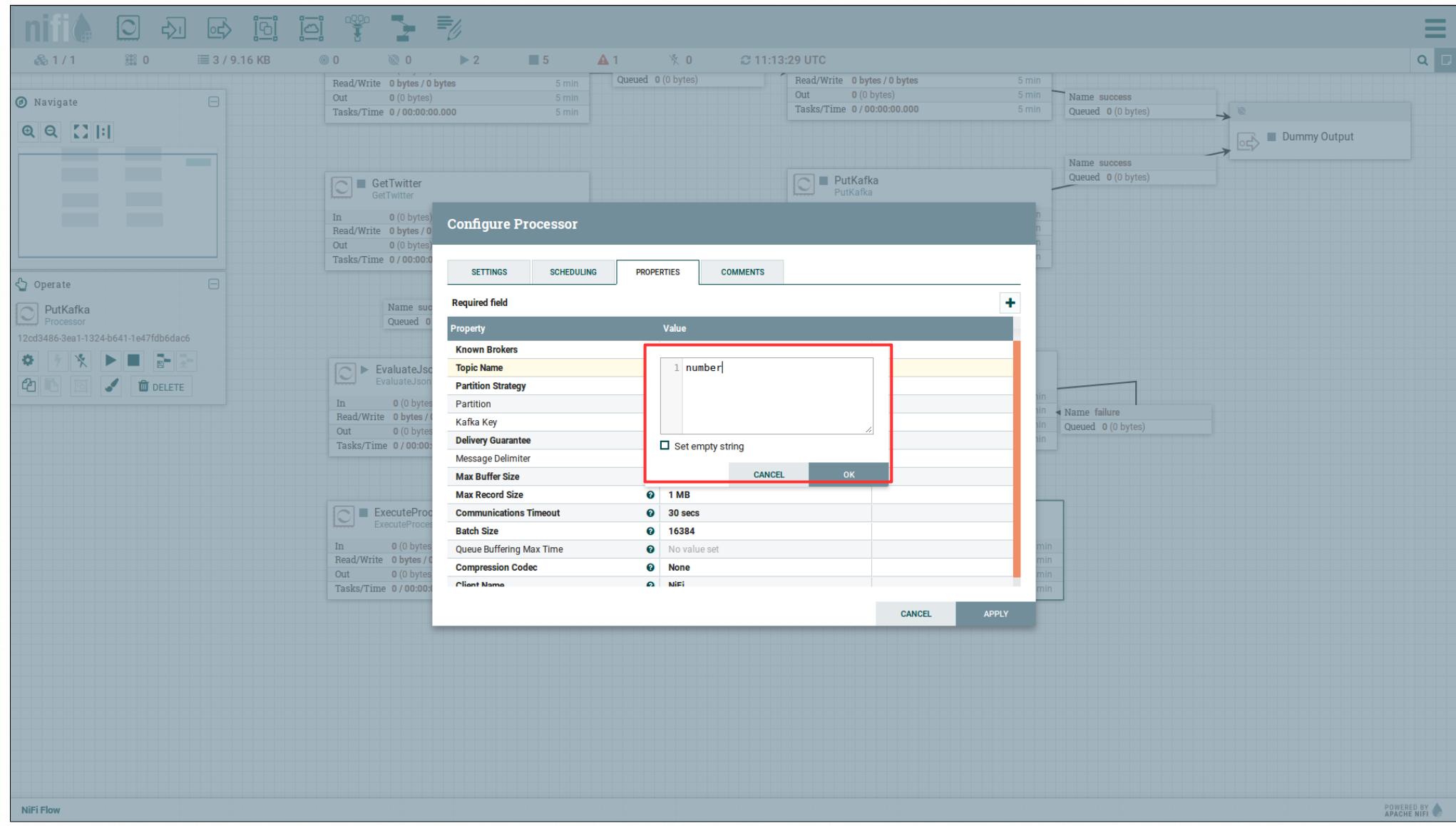
The screenshot shows the Apache NiFi user interface. A central modal window titled "Configure Processor" is open, specifically for an "ExecuteProcessor". The modal has tabs for SETTINGS, SCHEDULING, PROPERTIES, and COMMENTS, with PROPERTIES selected. Under the PROPERTIES tab, there is a "Required field" section. A table lists properties and their values:

Property	Value
Command	/home/nifi/random_number.py
Command Arguments	
Batch Duration	
Redirect Error Stream	
Argument Delimiter	

The value for "Command" is highlighted with a red box. At the bottom of the modal are "CANCEL" and "OK" buttons, with "OK" being the active button.

In the background, the main NiFi interface shows a flow with several processors: GetTwitter, EvaluateJsonPath, ExecuteProcessor, and a Dummy Output. The "ExecuteProcessor" node has its "Command" property set to the same value as in the modal: "/home/nifi/random_number.py".

Create process

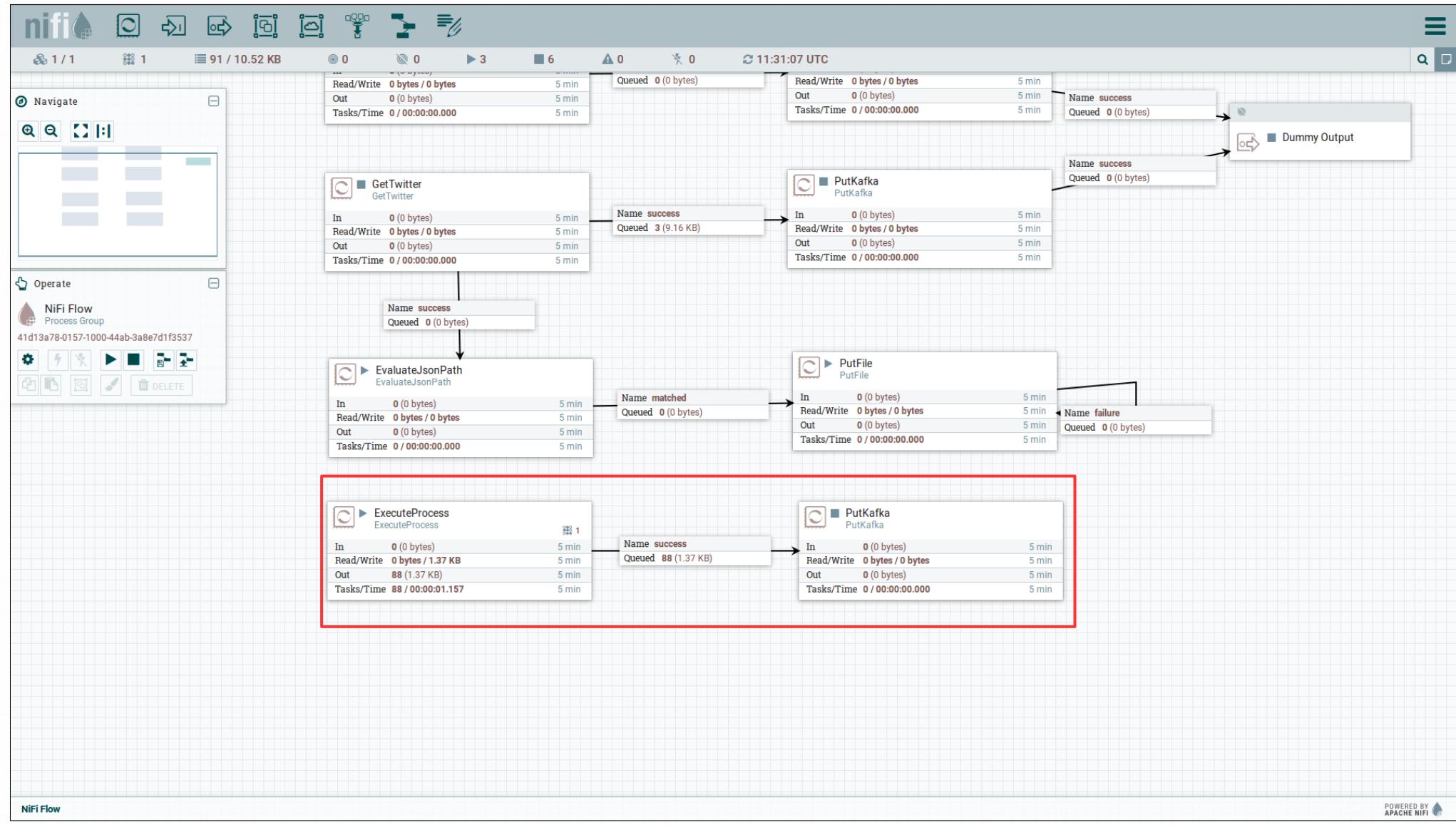




Create Kafka topic

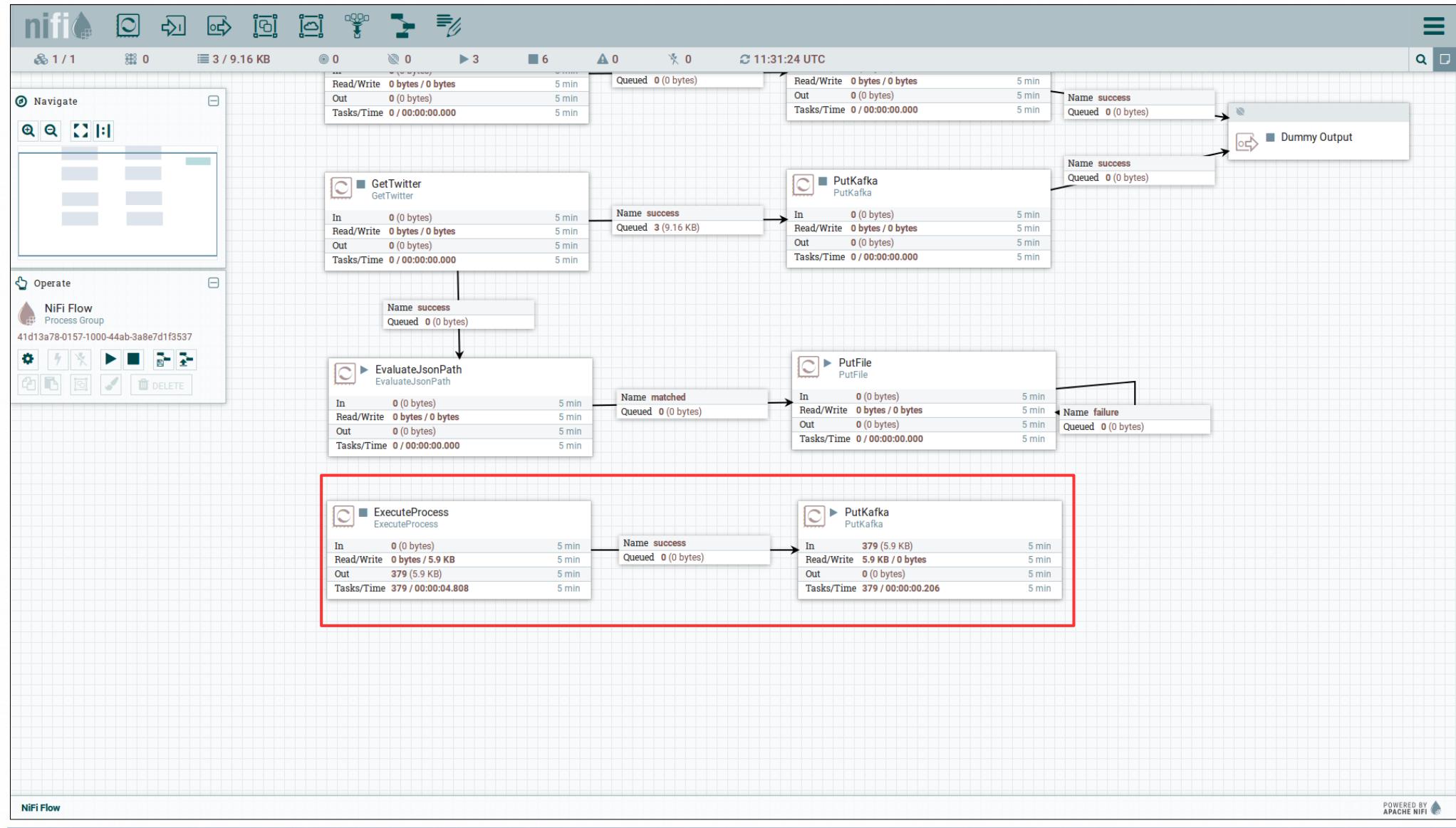
```
cd /usr/hdf/2.0.0.0-579/kafka/bin/  
./kafka-topics.sh --create --zookeeper  
localhost:2181 --replication-factor 1 --partitions  
2 --topic number
```

Start ExecuteProcess





Start ExecuteProcess





Check result on Kafka

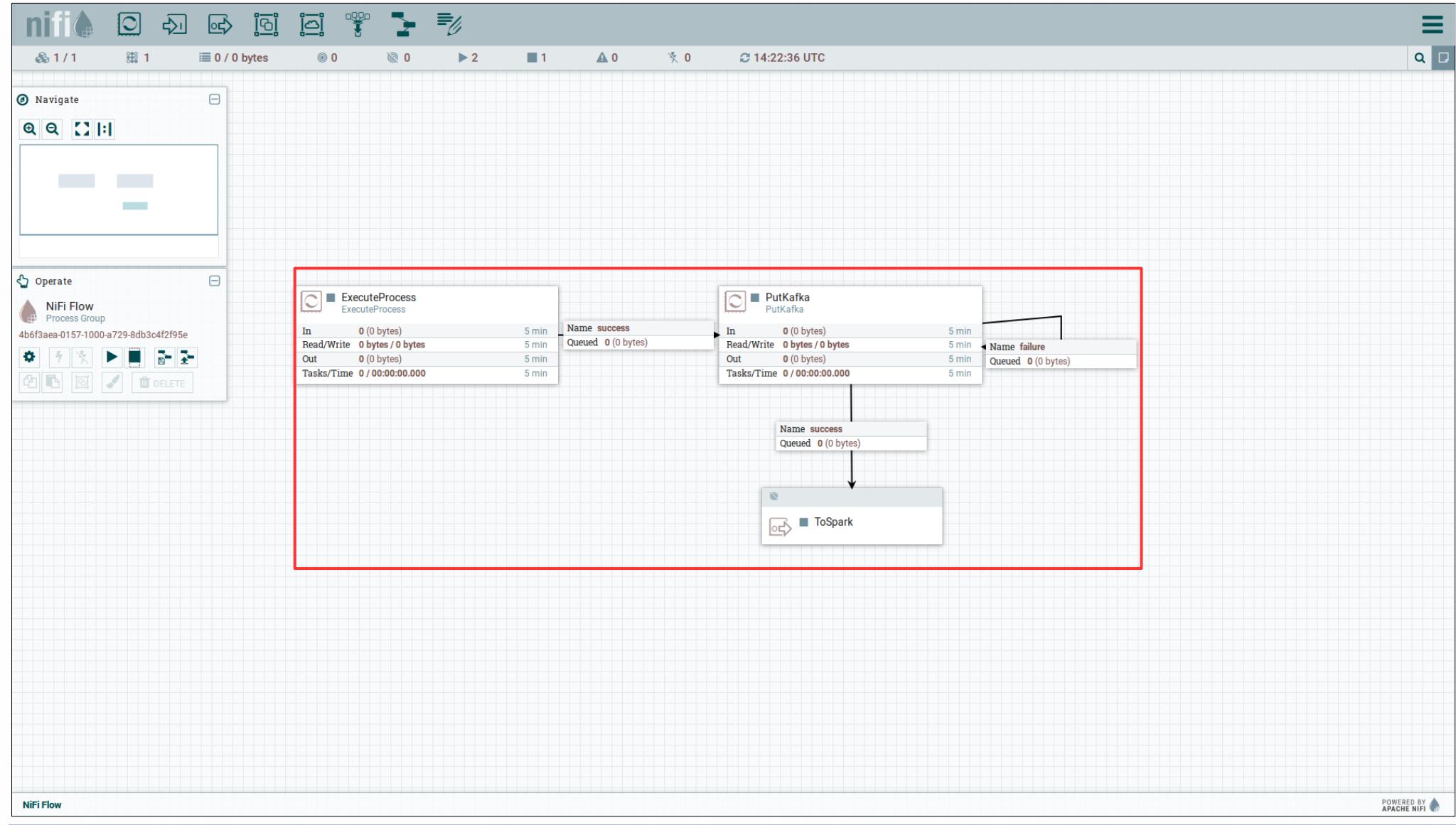
```
./kafka-console-consumer.sh --zookeeper  
localhost:2181 --topic number --from-beginning
```



To Spark Streaming

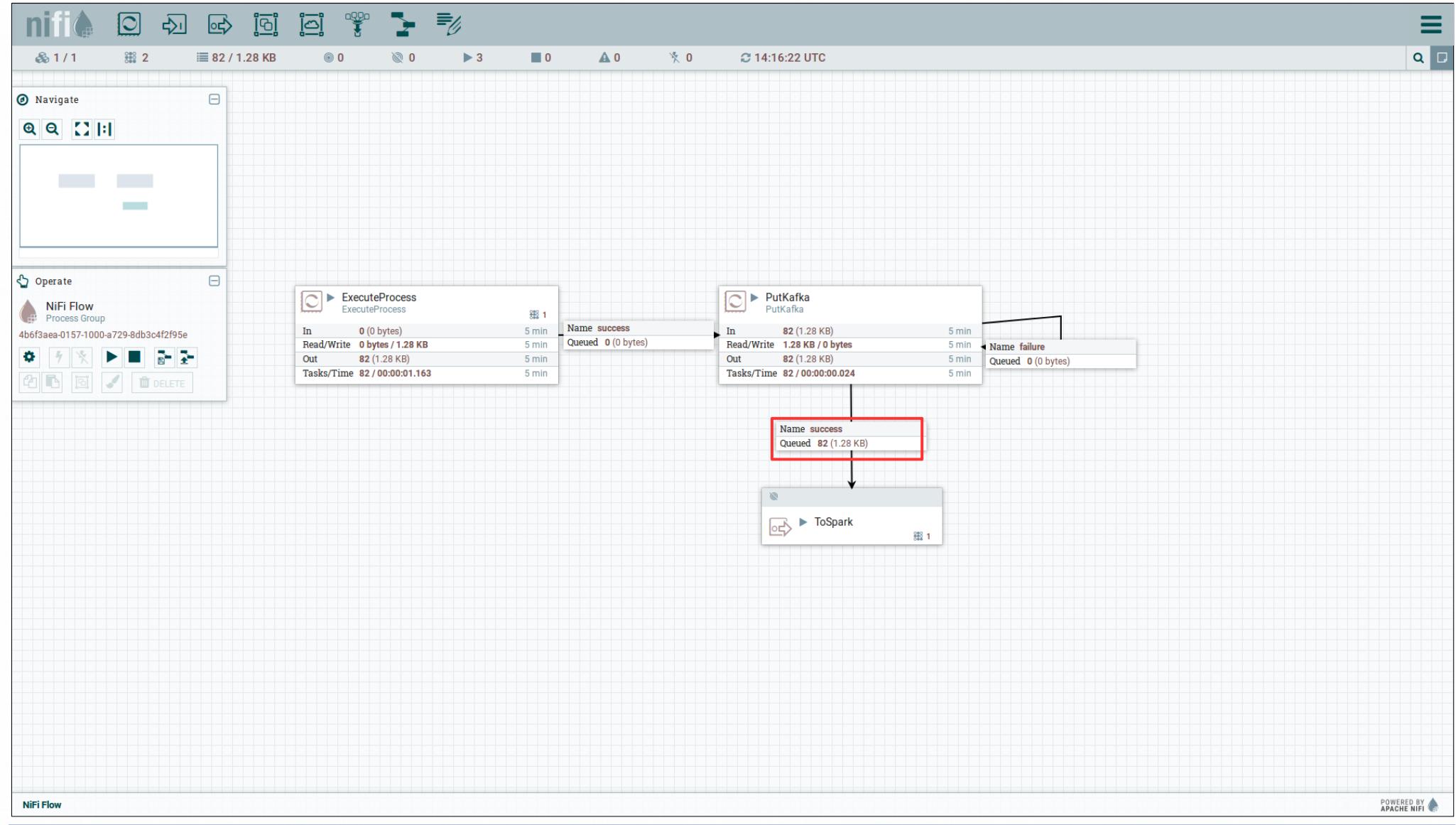


NiFi





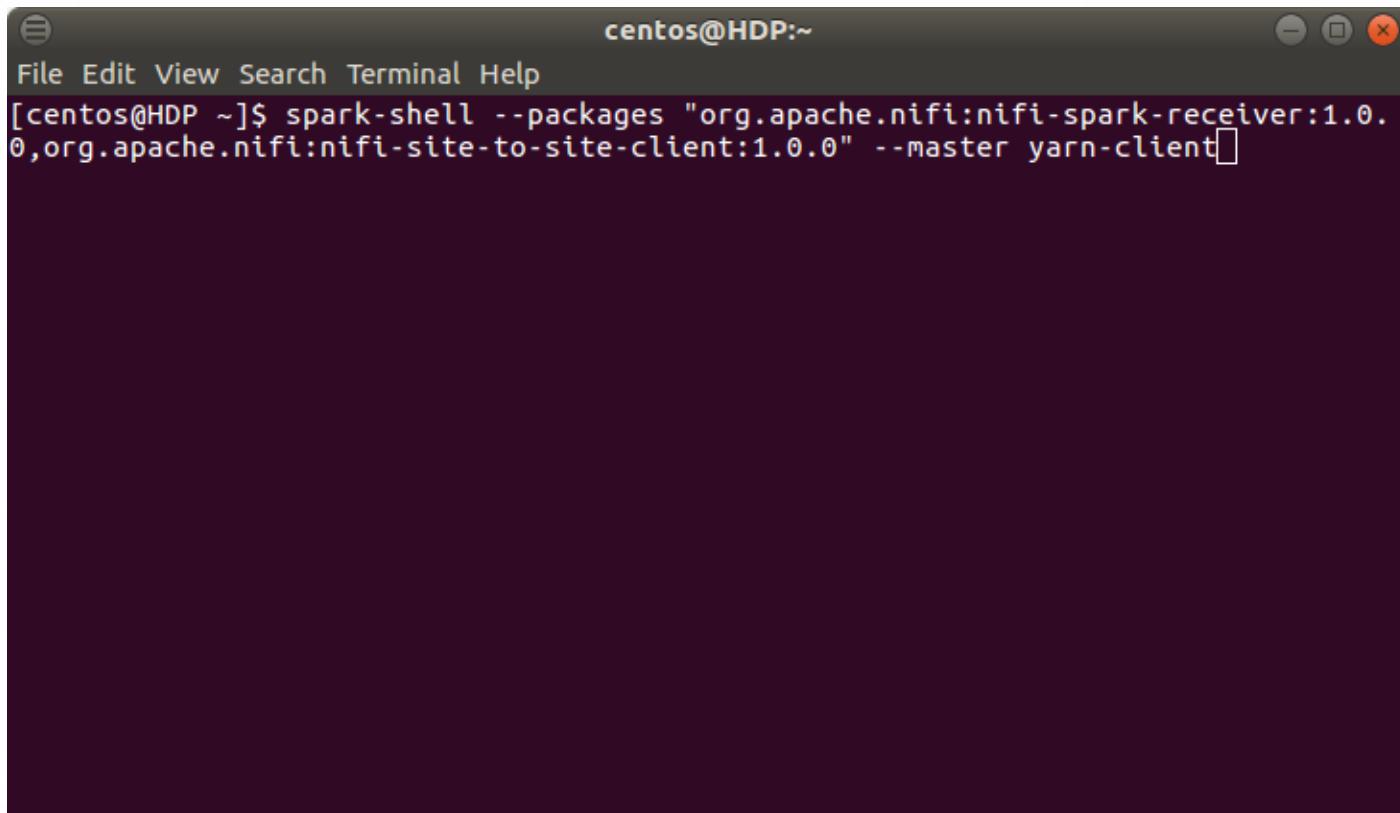
NiFi





Spark-Shell

```
spark-shell --packages "org.apache.nifi:nifi-spark-receiver:1.0.0,org.apache.nifi:nifi-site-to-site-client:1.0.0" --master yarn-client
```



A screenshot of a terminal window titled "centos@HDP:~". The window has a dark background and light-colored text. At the top, there's a menu bar with "File", "Edit", "View", "Search", "Terminal", and "Help". Below the menu, the prompt "[centos@HDP ~]\$" is followed by the command "spark-shell --packages \"org.apache.nifi:nifi-spark-receiver:1.0.0,org.apache.nifi:nifi-site-to-site-client:1.0.0\" --master yarn-client". The command is partially typed, with the cursor visible at the end of the package list.



Spark-Shell

:paste

```
centos@HDP:~  
File Edit View Search Terminal Help  
16/09/21 14:30:33 INFO ClientWrapper: Inspected Hadoop version: 2.7.3.2.5.0.0-12  
45  
16/09/21 14:30:33 INFO ClientWrapper: Loaded org.apache.hadoop.hive.shims.Hadoop  
23Shims for Hadoop version 2.7.3.2.5.0.0-1245  
16/09/21 14:30:33 INFO metastore: Trying to connect to metastore with URI thrift  
://hdp.maas:9083  
16/09/21 14:30:33 INFO metastore: Connected to metastore.  
16/09/21 14:30:33 INFO SessionState: Created local directory: /tmp/75f36064-c2ae  
-4541-b864-558e1ec2c5f6_resources  
16/09/21 14:30:33 INFO SessionState: Created HDFS directory: /tmp/hive/centos/75  
f36064-c2ae-4541-b864-558e1ec2c5f6  
16/09/21 14:30:33 INFO SessionState: Created local directory: /tmp/centos/75f360  
64-c2ae-4541-b864-558e1ec2c5f6  
16/09/21 14:30:33 INFO SessionState: Created HDFS directory: /tmp/hive/centos/75  
f36064-c2ae-4541-b864-558e1ec2c5f6/_tmp_space.db  
16/09/21 14:30:33 INFO SparkILoop: Created sql context (with Hive support)..  
SQL context available as sqlContext.  
  
scala> :paste  
// Entering paste mode (ctrl-D to finish)  
[]
```



Spark-Shell

```
import java.nio.charset.StandardCharsets  
import scala.collection.mutable.ListBuffer  
import org.apache.spark.SparkConf  
import org.apache.spark.SparkContext  
import org.apache.spark.rdd.RDD  
import org.apache.spark.sql.SQLContext  
import org.apache.spark.storage.StorageLevel
```



Spark-Shell

```
import org.apache.spark.streaming.{Seconds,  
StreamingContext, Time}  
  
import org.apache.nifi.remote.client.SiteToSiteClient  
  
import  
org.apache.nifi.remote.client.SiteToSiteClientConfig  
  
import org.apache.nifi.spark.NiFiReceiver  
  
import org.apache.nifi.spark.NiFiDataPacket
```



Spark-Shell

```
case class Record_sum(time: Time, id: String,  
sum: Double)
```

```
case class Record_count(time: Time, id: String,  
count: Int)
```

```
case class Record(time: Time, id: String, avg:  
Double)
```



Spark-Shell

```
val clientConfig = new SiteToSiteClient.Builder()  
    .url("http://nifi.maas:9090/nifi")  
    .portName("ToSpark")  
    .buildConfig();
```



Spark-Shell

```
val sparkConf = new  
SparkConf().setAppName("NiFi_Spark  
Streaming example")
```

```
val ssc = new StreamingContext(sc,  
Seconds(5))
```

```
val stream = ssc.receiverStream(new  
NiFiReceiver(clientConfig,  
StorageLevel.MEMORY_ONLY))
```



Spark-Shell

```
val lines = stream.map( (packet:  
NiFiDataPacket) => new  
String(packet.getContent(),  
StandardCharsets.UTF_8) )
```

```
val sum = lines.map(line => (line(0).toString,  
line(1).toDouble))  
    .reduceByKey((x, y) => x + y )
```



Spark-Shell

```
val count = lines.map(line => (line(0).toString,  
1))  
    .reduceByKey((x, y) => x + y )
```



Spark-Shell

```
val avg = lines.map(line => (line(0).toString,  
                                (line(1).toDouble, 1)))  
              .reduceByKey((x, y) => (x._1 + y._1,  
                                         x._2 + y._2))  
              .mapValues{ case (sum, count) =>  
                           (1.0 * sum) / count }
```



Spark-Shell

sum.print()

count.print()

avg.print()

ssc.start()



NiFi

```
centos@HDP:~  
File Edit View Search Terminal Help  
  
val sum = lines.map(line => (line(0).toString, line(1).toDouble))  
    .reduceByKey((x, y) => x + y )  
  
val count = lines.map(line => (line(0).toString, 1))  
    .reduceByKey((x, y) => x + y )  
  
val avg = lines.map(line => (line(0).toString, (line(1).toDouble, 1)))  
    .reduceByKey((x, y) => (x._1 + y._1, x._2 + y._2))  
    .mapValues{ case (sum, count) => (1.0 * sum) / count }  
  
sum.print()  
count.print()  
avg.print()  
ssc.start()  
  
// Exiting paste mode, now interpreting.  
□
```



NiFi

```
centos@HDP:~  
File Edit View Search Terminal Help  
16/09/21 14:26:35 INFO DAGScheduler: Job 6 finished: print at <console>:65, took  
0.012255 s  
-----  
Time: 1474467995000 ms  
  
 (4,32.0)  
 (8,32.0)  
 (6,32.0)  
 (2,32.0)  
 (7,32.0)  
 (5,32.0)  
 (9,32.0)  
 (3,32.0)  
 (1,40.12698412698413)  
  
16/09/21 14:26:35 INFO JobScheduler: Finished job streaming job 1474467995000 ms  
.2 from job set of time 1474467995000 ms  
16/09/21 14:26:35 INFO JobScheduler: Total delay: 0.422 s for time 1474467995000  
ms (execution: 0.356 s)  
16/09/21 14:26:35 INFO ReceivedBlockTracker: Deleting batches ArrayBuffer()  
16/09/21 14:26:35 INFO InputInfoTracker: remove old batch metadata:  
□
```



NiFi

Logged in as: d

All Applications

hadoop

Cluster Metrics	Scheduler Metrics	Show 20 entries	Search:														
Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	Vcores Used	Vcores Total	Vcores Reserved	Active Nodes	Decommissioned Nodes	Lost Nodes	Unhealthy Nodes	Rebooted Nodes		
5	0	1	4	1	1 GB	2 GB	0 B	1	1	0	1	0	0	0	0		
Scheduler Type		Scheduling Resource Type				Minimum Allocation				Maximum Allocation							
Capacity Scheduler		[MEMORY]				<memory:512, vCores:1>				<memory:2048, vCores:1>							
ID	User	Name	Application Type	Queue	Application Priority	StartTime	FinishTime	State	FinalStatus	Running Containers	Allocated CPU Vcores	Allocated Memory MB	% of Queue	% of Cluster	Progress	Tracking UI	Blacklist Nodes
application_1474450620212_0004	centos	Spark shell	SPARK	default	0	Wed Sep 21 22:29:40 +0800 2016	N/A	RUNNING	UNDEFINED	1	1	1024	50.0	50.0	<input type="button"/>	ApplicationMaster	0
application_1474450620212_0003	centos	Spark shell	SPARK	default	0	Wed Sep 21 20:04:13 +0800 2016	Wed Sep 21 20:07:16 +0800 2016	FINISHED	SUCCEEDED	N/A	N/A	N/A	0.0	0.0	<input type="button"/>	History	N/A
application_1474450620212_0002	zeppelin	Zeppelin	SPARK	default	0	Wed Sep 21 17:47:12 +0800 2016	Wed Sep 21 17:48:09 +0800 2016	FINISHED	SUCCEEDED	N/A	N/A	N/A	0.0	0.0	<input type="button"/>	History	N/A
application_1474450620212_0001	zeppelin	Zeppelin	SPARK	default	0	Wed Sep 21 17:42:56 +0800 2016	Wed Sep 21 17:46:49 +0800 2016	FINISHED	SUCCEEDED	N/A	N/A	N/A	0.0	0.0	<input type="button"/>	History	N/A
application_1474270722186_0001	zeppelin	Zeppelin	SPARK	default	0	Mon Sep 19 15:55:38 +0800 2016	Wed Sep 21 17:40:22 +0800 2016	FAILED	FAILED	N/A	N/A	N/A	0.0	0.0	<input type="button"/>	History	N/A

Showing 1 to 5 of 5 entries

First Previous 1 Next Last



Homework



Homework

1. Create three topics.
2. Match Twitter's text, user.name, created_at.
(use unmatched)
3. Output text to **Spark Streaming** and write a wordcount.
4. Bonus: Output the result to hive.



Homework

What you have to give me:

1. ScreenShots.
2. Source Code.
3. Documents. (PDF)
4. GitHub or Bitbucket.



Reference

- <http://hortonworks.com/>
- <https://community.hortonworks.com/articles/12708/nifi-feeding-data-to-spark-streaming.html>
- https://blogs.apache.org/nifi/entry/stream_processing_nifi_and_spark