Apache NIFi Deep Dive 300

Timothy Spann

Developer Advocate / StreamNative

Agenda

Tuesday 17:10 UTC
Apache NIFi Deep Dive 300

For Data Engineers who have flows already in production, I will dive deep into best practices, advanced use cases, performance optimizations, tips, tricks, edge cases, and interesting examples. This is a master class for those looking to learn quickly things I have picked up after years in the field with Apache NiFi in production. This will be interactive and I encourage questions and discussions.You will take away examples and tips in slides, github, and articles.

This talk will cover:

- Load Balancing
- Parameters and Parameter Contexts
- Stateless vs Stateful NiFi
- Reporting Tasks
- NiFi CLI
- NiFi REST Interface
- DevOps
- Advanced Record Processing
- Schemas
- RetryFlowFile
- Lookup Services
- Record Path
- Expression Language
- Advanced Error Handling Techniques

My Other Talks & Apache Pulsar Talks

- Tuesday 14:10 UTC - Exclusive Producer: Using Apache Pulsar to build distributed applications by Matteo Merli
- Wednesday 15:50 UTC - Replicated Subscriptions: taking Apache Pulsar Geo-Replication to next level by Matteo Merli
- Tuesday 17:10 UTC - Apache NIFi Deep Dive 300 by Tim Spann
- Tuesday 18:00 UTC - Apache Deep Learning 302 by Tim Spann
- Wednesday 15:00 UTC - Smart Transit: Real-Time Transit Information with FLaNK by David Kjerrumgaard & Tim Spann
- Wednesday 17:10 UTC - Cracking the Nut, Solving Edge AI... by David Kjerrumgaard & Tim Spann
- Thursday 14:10 UTC - Apache NiFi 101: Introduction and Best Practices - Tim Spann

Tim SPANN

https://github.com/tspannhw

https://www.datainmotion.dev/

Announcing

Flink SQL on StreamNative Cloud

Timothy Spann

Developer Advocate, StreamNative

Tim Spann is a Developer Advocate for StreamNative. He works with StreamNative Cloud, Apache Pulsar, Apache Flink, Flink SQL, Apache NiFi, MiniFi, Apache MXNet, TensorFlow, Apache Spark, big data, the IoT, machine learning, and deep learning. Tim has over a decade of experience with the IoT, big data, distributed computing, streaming technologies, and Java programming.

FLaNK and FLiP Stacks

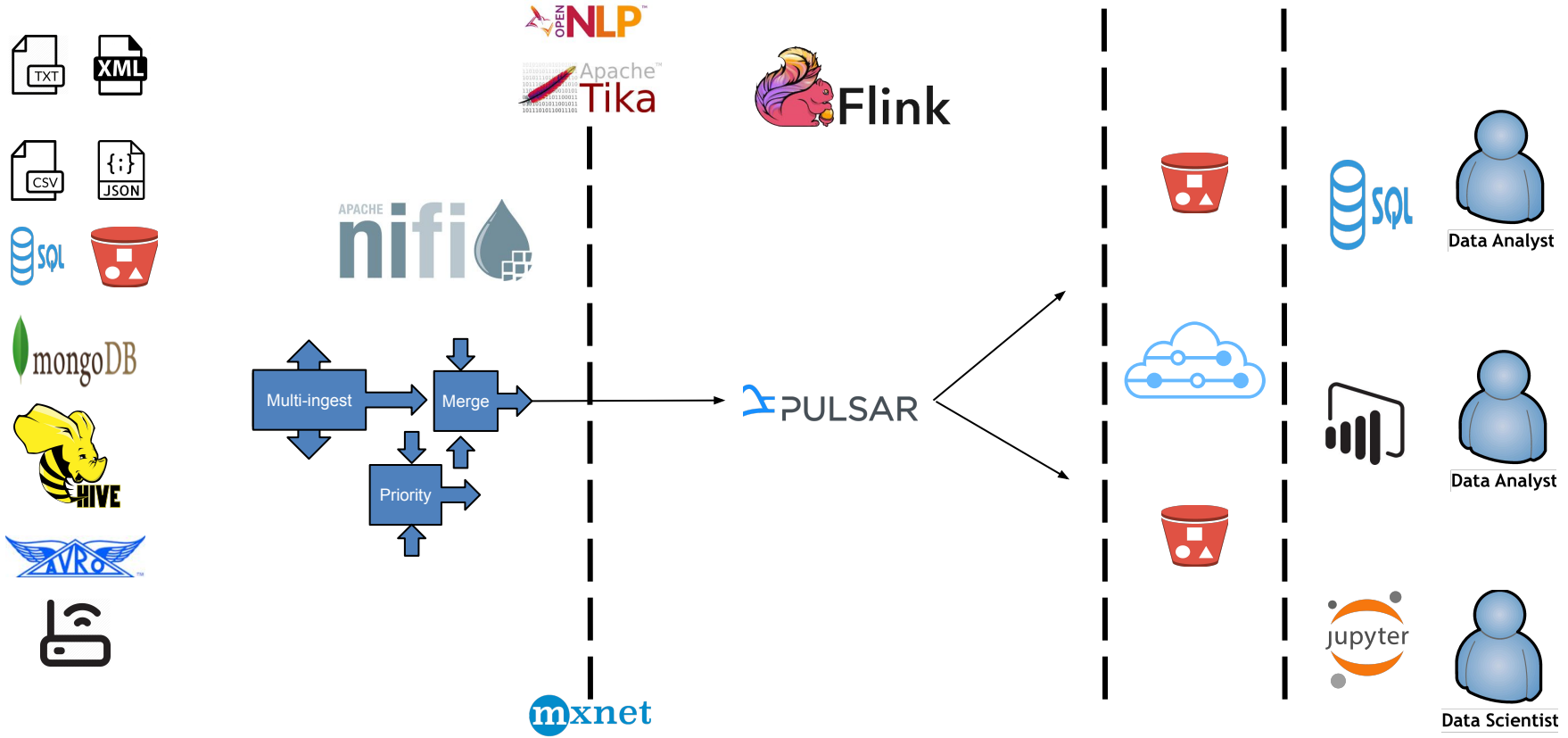
- Apache **Fl**ink
- Apache **N**iFi
- Apache **K**afka

- Apache **Fl**ink
- Apache **P**ulsar
- StreamNative's Flink Connector for Pulsar
- Apache +++

Apache projects are the way for all streaming use cases.

StreamNative Solution

All Data - Anytime - Every Cloud - Multi-Protocol



NiFi Flows

Pick Data Source(s)

Apache Pulsar, Databases, Files, REST Endpoints, Cloud Resources, NoSQL, JMS, MQTT, TCP/IP, etc…

Validate and Convert

Validate types, nulls, convert types and check against schemas.

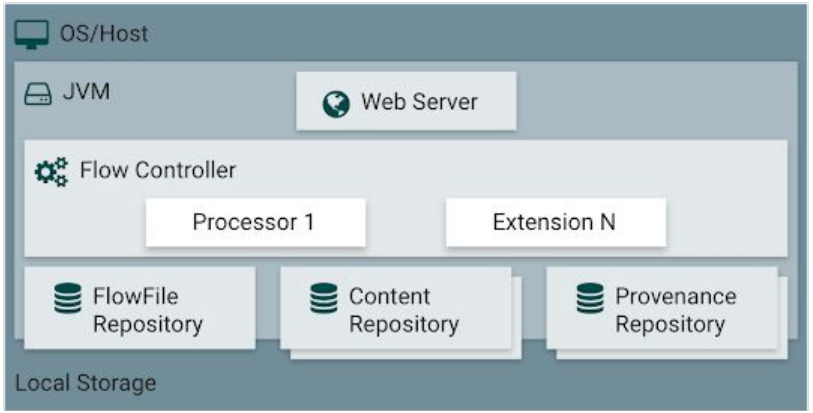
Aggregate or Eliminate

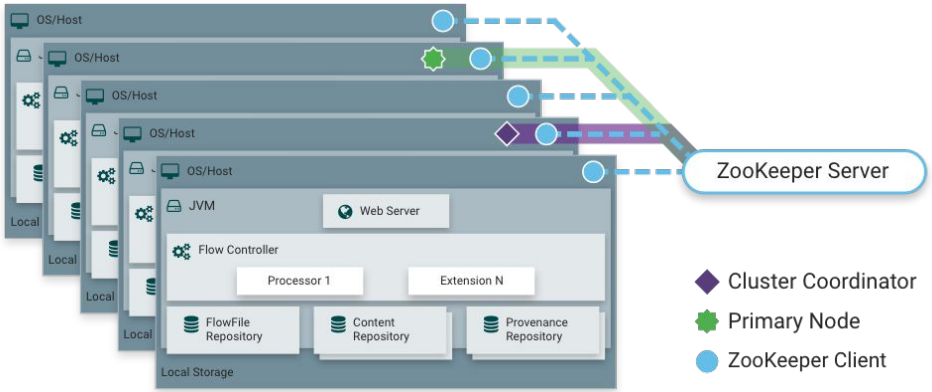
build up key data with multiple sources or removing unneeded via Queries on records

Send to Data Sink(s)

Apache Pulsar, Databases, Files, REST Endpoints, Cloud Resources, NoSQL, JMS, MQTT, TCP/IP, etc…

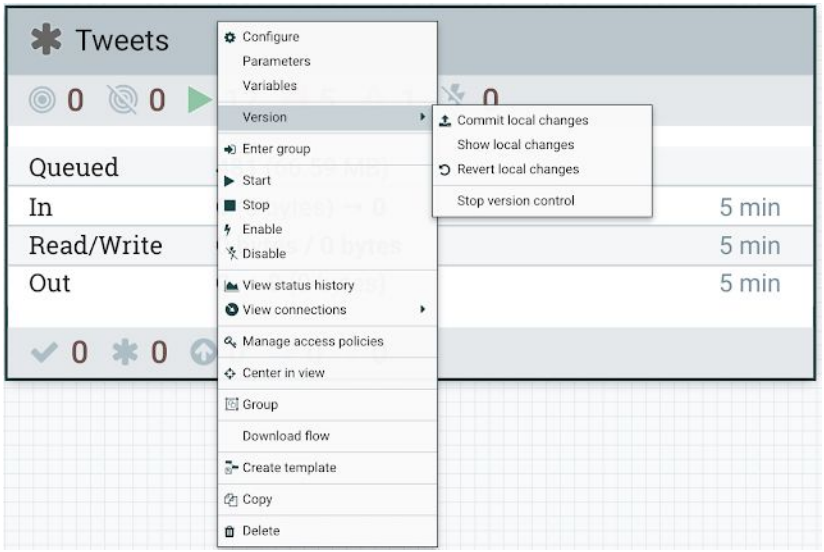
Architecture

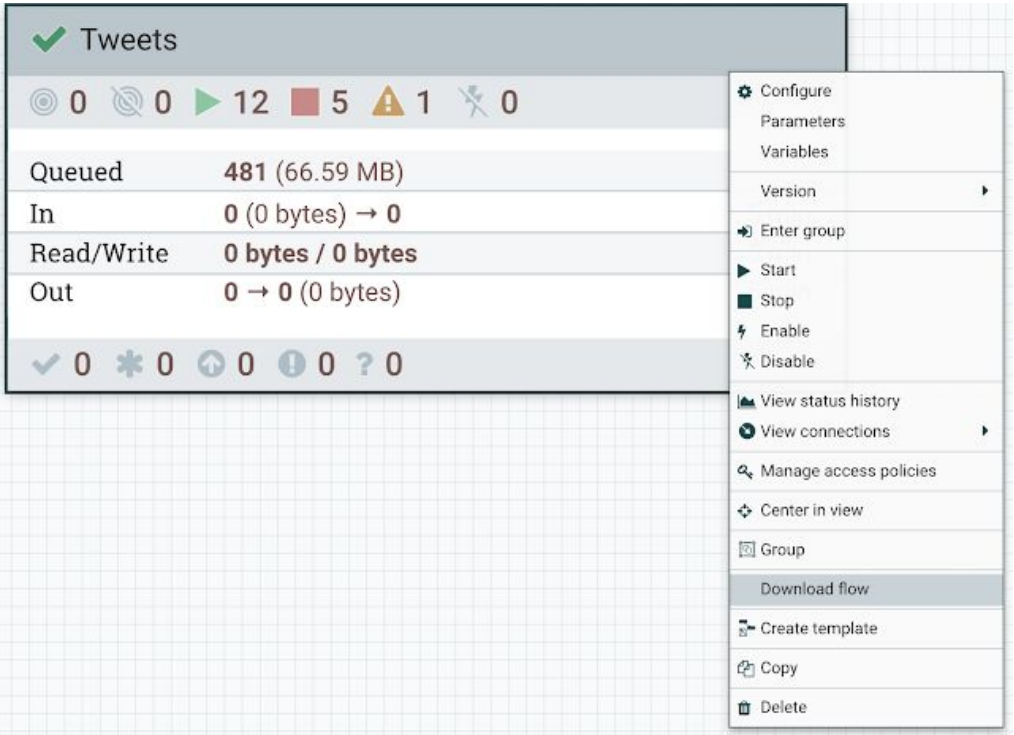




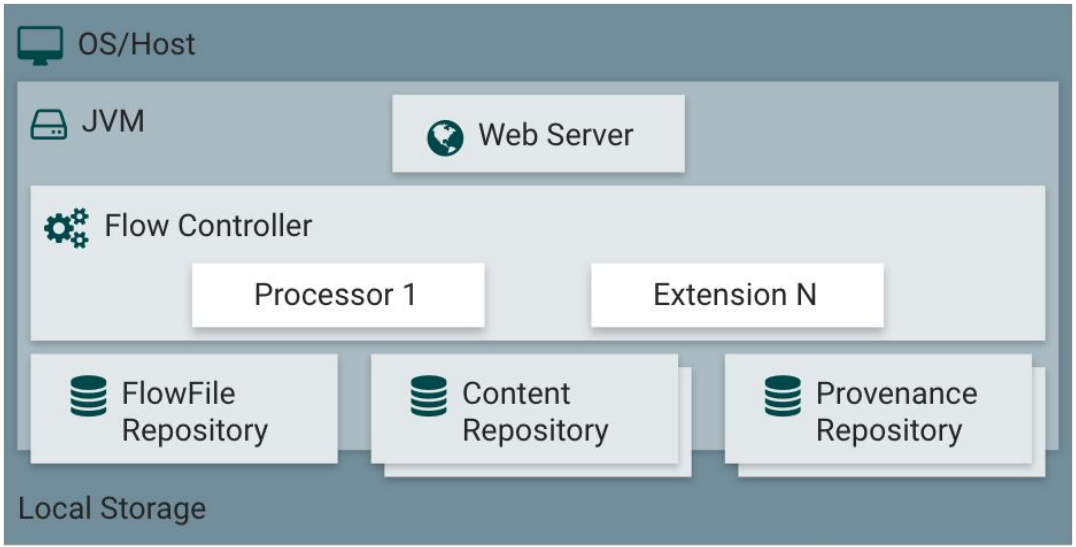
https://nifi.apache.org/docs/nifi-docs/html/overview.html

Version Control (Github and Beyond)





Repositories



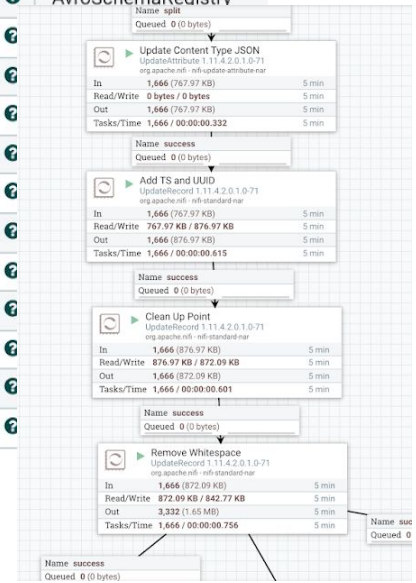
https://nifi.apache.org/docs/nifi-docs/html/nifi-in-depth.html#repositories

Record Processors

- XML, CSV, JSON, AVRO and more
- Schemas or Inferred Schemas
- Easily convert between them
- Support SQL with Apache Calcite

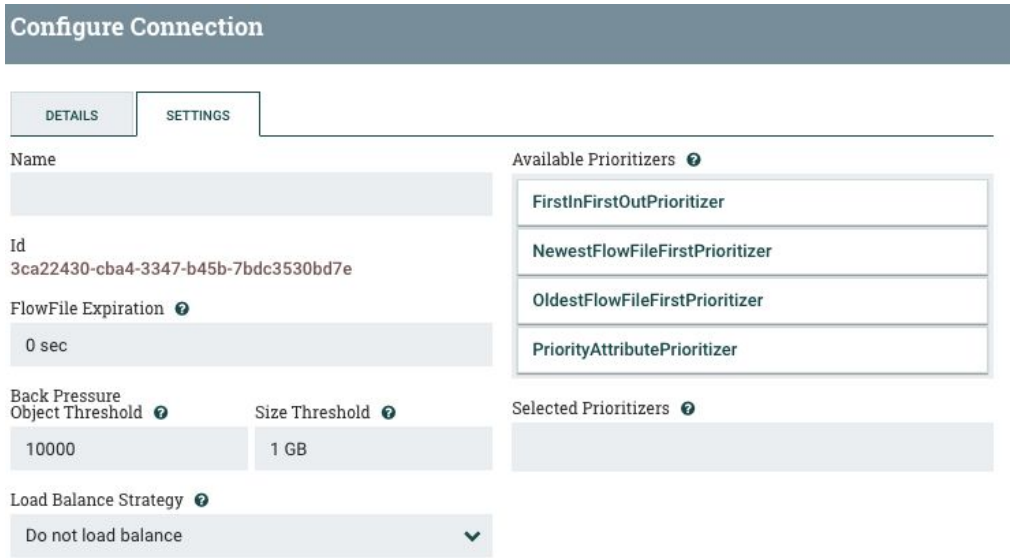
Property	Value
Record Reader	**XMLReader**
Record Writer	**JsonRecordSetWriter**
Include Zero Record FlowFiles	**false**
Cache Schema	**true**
query1	SELECT * FROM FLOWFILE

Property	Value
Schema Access Strategy	**Infer Schema**
Schema Registry	AvroSchemaRegistry
Schema Name	
Schema Version	
Schema Branch	
Schema Text	
Schema Inference Cache	
Expect Records as Array	
Attribute Prefix	
Field Name for Content	
Date Format	
Time Format	
Timestamp Format	



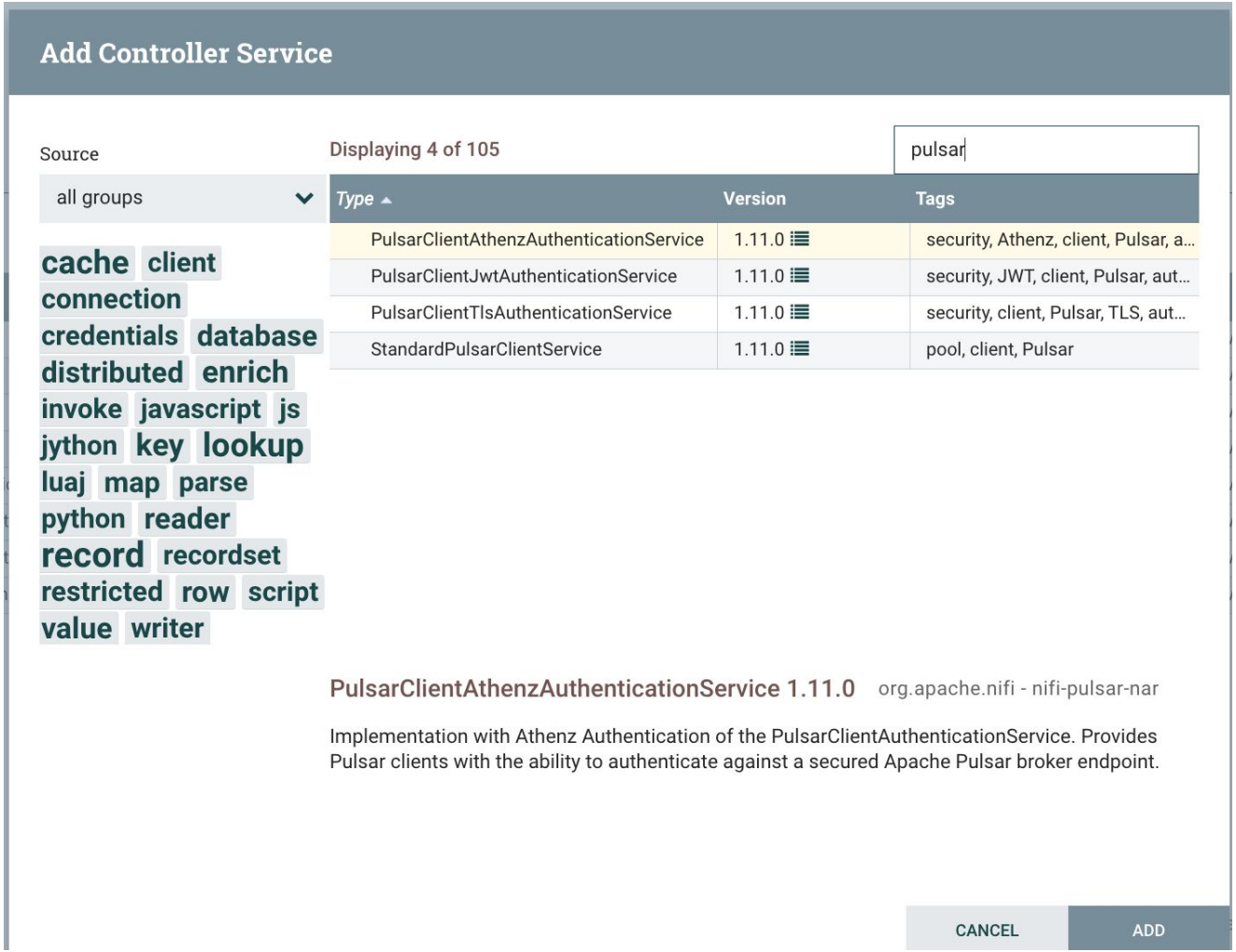
https://www.datainmotion.dev/2019/03/advanced-xml-processing-with-apache.html

Backpressure & Prioritizers

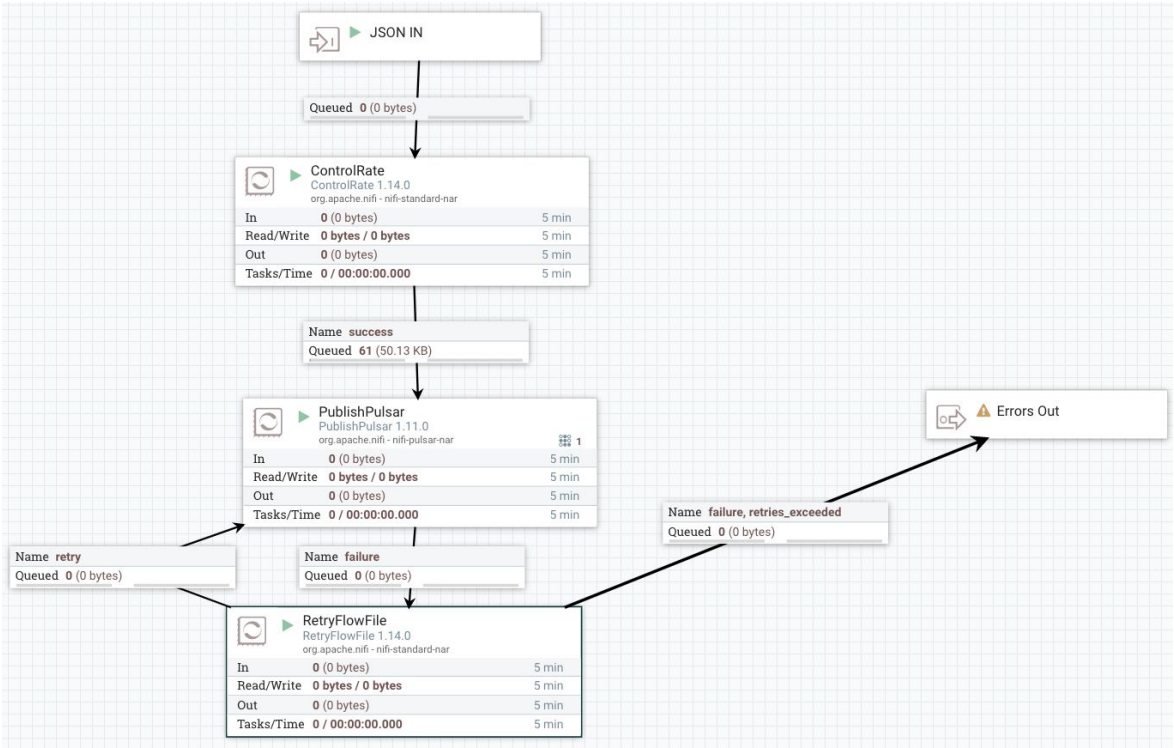


https://www.datainmotion.dev/2019/11/exploring-apache-nifi-110-parameters.html

System Diagnostics

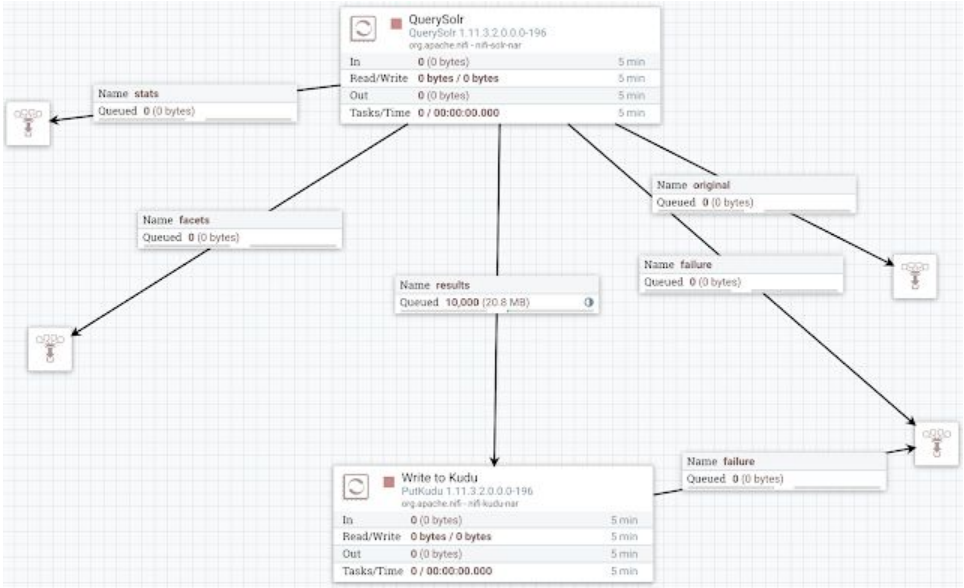

Add Controller Service
Source
all groups
cache client connection credentials database distributed enrich invoke javascript js jython key lookup luaj map parse python reader record recordset restricted row script value writer
Displaying 4 of 105
pulsar
Type
Version
Tags
PulsarClientAthenzAuthenticationService 1.11.0 security, Athenz, client, Pulsar, a...
PulsarClientJwtAuthenticationService 1.11.0 security, JWT, client, Pulsar, aut...
PulsarClientTlsAuthenticationService 1.11.0 security, client, Pulsar, TLS, aut...
StandardPulsarClientService 1.11.0 pool, client, Pulsar
PulsarClientAthenzAuthenticationService 1.11.0
org.apache.nifi - nifi-pulsar-nar
Implementation with Athenz Authentication of the PulsarClientAuthenticationService. Provides Pulsar clients with the ability to authenticate against a secured Apache Pulsar broker endpoint.
CANCEL
ADD

Scheduling


JSON IN
Queued 0 (0 bytes)
ControlRate
ControlRate 1.14.0
org.apache.nifi - nifi-standard-nar
In 0 (0 bytes) 5 min
Read/Write 0 bytes / 0 bytes 5 min
Out 0 (0 bytes) 5 min
Tasks/Time 0 / 00:00:00.000 5 min
Name success
Queued 61 (50.13 KB)
PublishPulsar
PublishPulsar 1.11.0
org.apache.nifi - nifi-pulsar-nar
1
In 0 (0 bytes) 5 min
Read/Write 0 bytes / 0 bytes 5 min
Out 0 (0 bytes) 5 min
Tasks/Time 0 / 00:00:00.000 5 min
Errors Out
Name failure, retries_exceeded
Queued 0 (0 bytes)
Name retry
Queued 0 (0 bytes)
Name failure
Queued 0 (0 bytes)
RetryFlowFile
RetryFlowFile 1.14.0
org.apache.nifi - nifi-standard-nar
In 0 (0 bytes) 5 min
Read/Write 0 bytes / 0 bytes 5 min
Out 0 (0 bytes) 5 min
Tasks/Time 0 / 00:00:00.000 5 min

SOLR Connectors

- XML, CSV, JSON, AVRO and more
- Schemas or Inferred Schemas
- Use Records or Raw Text
- Support SQL with Apache Calcite



Configure Processor

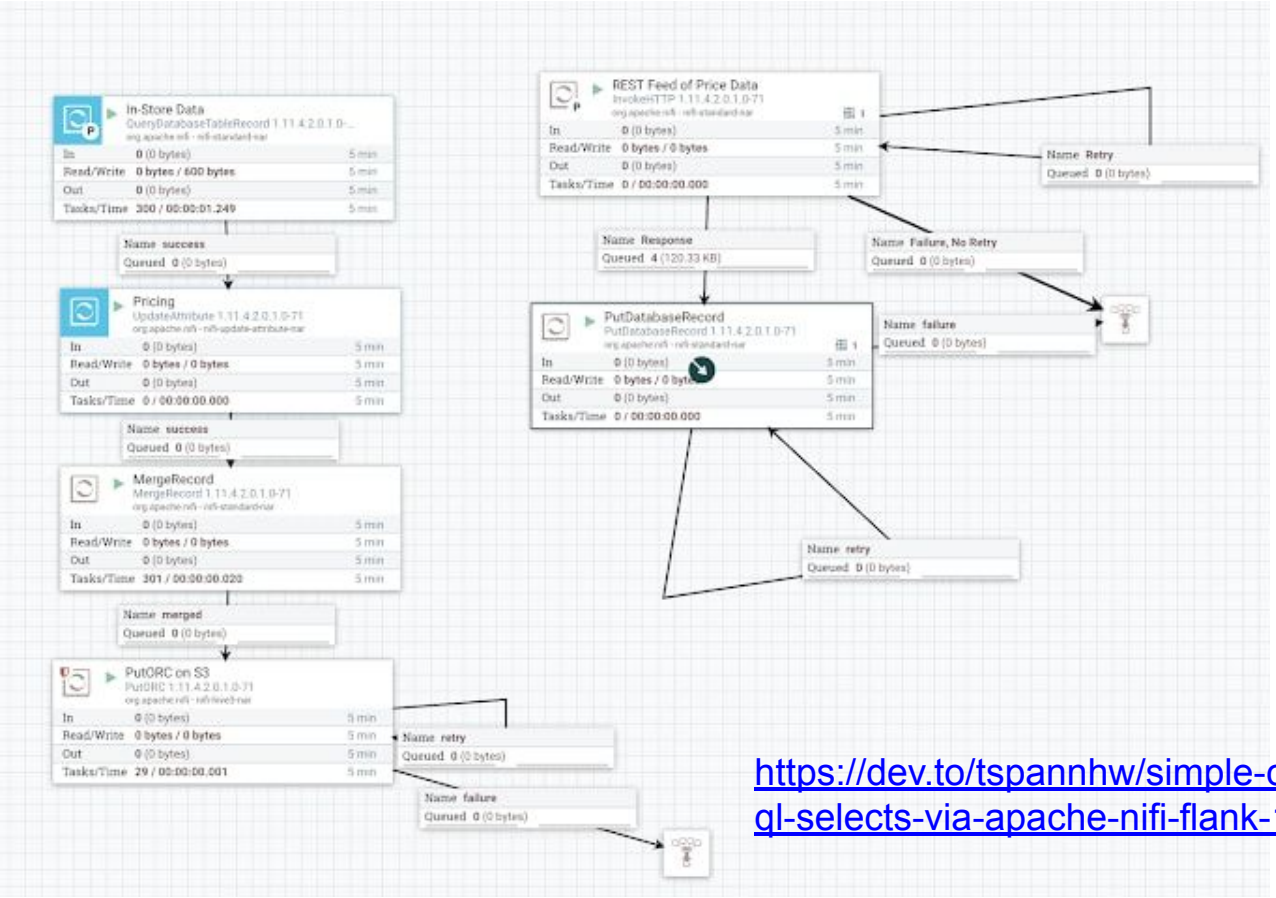
Stopped

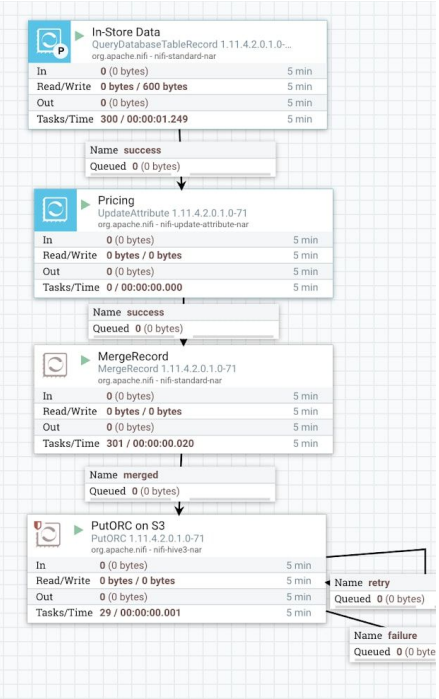
SETTINGS | SCHEDULING | PROPERTIES | COMMENTS

Required field

Property	Value
Solr Type	Cloud
Solr Location	.com:2181/solr
Collection	sensors
Solr Update Path	/update
Record Reader	JsonTreeReader
Fields To Index	sensor_id,sensor_ts
Commit Within	5000
Kerberos Credentials Service	No value set
Kerberos Principal	No value set
Kerberos Password	No value set
Basic Auth Username	No value set
Basic Auth Password	No value set

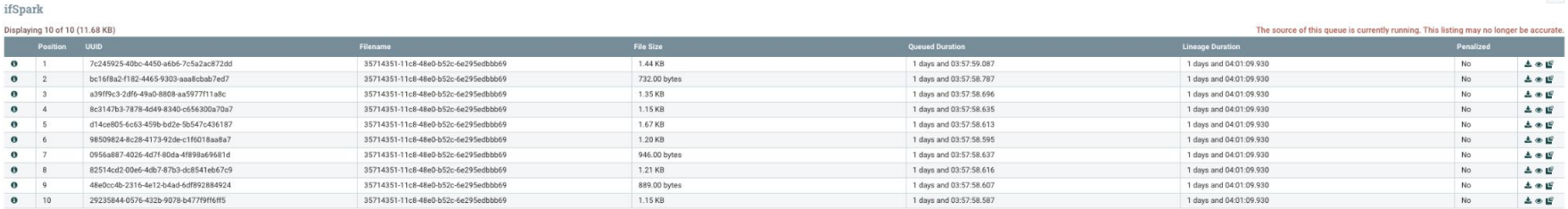
Simple CDC





https://dev.to/tspannhw/simple-change-data-capture-cdc-with-sql-selects-via-apache-nifi-flank-19m4

Bulletin Board



ifSpark

Displaying 10 of 10 (11.68 KB)

The source of this queue is currently running. This listing may no longer be accurate.

	Position	UUID	Filename	File Size	Queued Duration	Lineage Duration	Penalized	
	1	7c245925-40bc-4450-a6b6-7c5a2ac872dd	35714351-11c8-48e0-b52c-6e295edbbb69	1.44 KB	1 days and 03:57:59.087	1 days and 04:01:09.930	No	
	2	bc16f8a2-f182-4465-9303-aaa8cbab7ed7	35714351-11c8-48e0-b52c-6e295edbbb69	732.00 bytes	1 days and 03:57:58.787	1 days and 04:01:09.930	No	
	3	a39ff9c3-2df6-49a0-8808-aa5977f11a8c	35714351-11c8-48e0-b52c-6e295edbbb69	1.35 KB	1 days and 03:57:58.696	1 days and 04:01:09.930	No	
	4	8c3147b3-7878-4d49-8340-c656300a70a7	35714351-11c8-48e0-b52c-6e295edbbb69	1.15 KB	1 days and 03:57:58.635	1 days and 04:01:09.930	No	
	5	d14ce805-6c63-459b-bd2e-5b547c436187	35714351-11c8-48e0-b52c-6e295edbbb69	1.67 KB	1 days and 03:57:58.613	1 days and 04:01:09.930	No	
	6	98509824-8c28-4173-92de-c1f6018aa8a7	35714351-11c8-48e0-b52c-6e295edbbb69	1.20 KB	1 days and 03:57:58.595	1 days and 04:01:09.930	No	
	7	0956a887-4026-4d7f-80da-4f898a69681d	35714351-11c8-48e0-b52c-6e295edbbb69	946.00 bytes	1 days and 03:57:58.637	1 days and 04:01:09.930	No	
	8	82514cd2-00e6-4db7-87b3-dc8541eb67c9	35714351-11c8-48e0-b52c-6e295edbbb69	1.21 KB	1 days and 03:57:58.616	1 days and 04:01:09.930	No	
	9	48e0cc4b-2316-4e12-b4ad-6df892884924	35714351-11c8-48e0-b52c-6e295edbbb69	889.00 bytes	1 days and 03:57:58.607	1 days and 04:01:09.930	No	
	10	29235844-0576-432b-9078-b477f9ff6ff5	35714351-11c8-48e0-b52c-6e295edbbb69	1.15 KB	1 days and 03:57:58.587	1 days and 04:01:09.930	No	

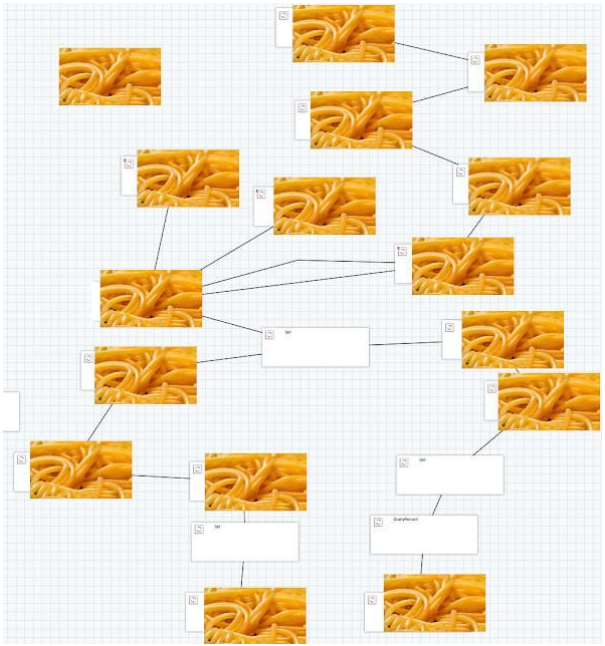
https://www.datainmotion.dev/2019/04/simple-apache-nifi-operations-dashboard.html

DevOps: Automate All The Things

```
nifi-toolkit/bin/cli.sh nifi pg-enable-services -u http://server:8080
--processGroupId root
nifi pg-status -u http://server:8080 -verbose -pgid a8e909d9f-adf-asdsdf
curl 'http://server:8080/nifi-api/flow/parameter-contexts'
nifi pg-list -u http://server:8080
```

https://www.datainmotion.dev/2021/01/automating-starting-services-in-apache.html

No More Spaghetti Flows - DO NOT

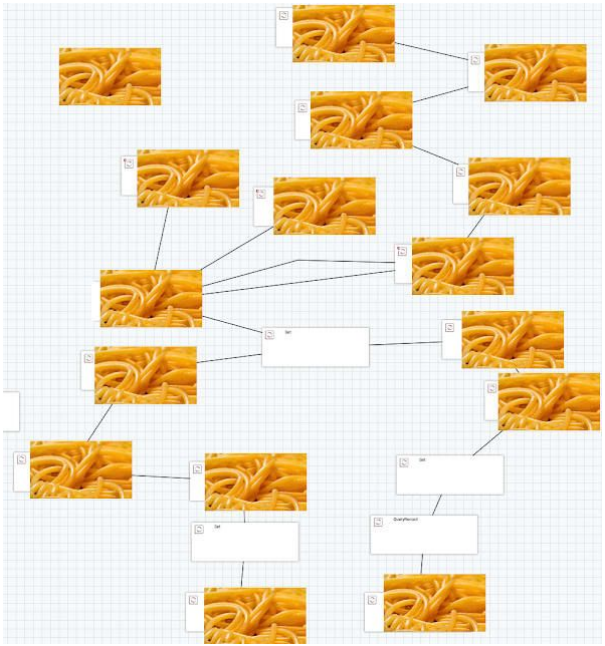


https://dev.to/tspannhw/no-more-spaghetti-flows-2emd

Do Not

- Do not Put 1,000 Flows on one workspace.
- If your flow has hundreds of steps, this is a Flow Smell. Investigate why.
- Do not Use ExecuteProcess, ExecuteScripts or a lot of Groovy scripts as a default, look for existing processors
- Do not Use Random Custom Processors you find that have no documentation or are unknown.
- Do not forget to upgrade, if you are running anything before Apache NiFi 1.14, upgrade now!
- Do not run on default 512M RAM.
- Do not run one node and think you have a highly available cluster.
- Do not split a file with millions of records to individual records in one shot without checking available space/memory and back pressure.
- Use Split processors only as an absolute last resort. Many processors are designed to work on FlowFiles that contain many records or many lines of text. Keeping the FlowFiles together instead of splitting them apart can often yield performance that is improved by 1-2 orders of magnitude.

No More Spaghetti Flows - DO



https://dev.to/tspannhw/no-more-spaghetti-flows-2emd

Do

- Reduce, Reuse, Recycle. Use Parameters to reuse common modules.
- Put flows, reusable chunks (write to Slack, Database, Kafka) into separate Process Groups.
- Write custom processors if you need new or specialized features
- Use Record Processors everywhere
- Read the Docs!
- Use the NiFi Registry for version control.
- Use NiFi CLI and DevOps for Migrations.
- Walk through your flow and make sure you understand every step and it's easy to read and follow. Is every processor used? Are there dead ends?
- Do run Zookeeper on different nodes from Apache NiFi.
- Use routing based on content and attributes to allow one flow to handle multiple nearly identical flows is better than deploying the same flow many times with tweaks to parameters in same cluster.
- Use the correct driver for your database. There's usually a couple different JDBC drivers.

Split JSON and EvaluateJSONPath

Property	Value
JsonPath Expression	$.*
Null Value Representation	empty string

Property	Value
Destination	flowfile-attribute
Return Type	json
Path Not Found Behavior	ignore
Null Value Representation	empty string
url	$.url

https://dzone.com/articles/lets-build-a-simple-ingest-to-cloud-data-warehouse

Query Record - SQL

Property	Value
Record Reader	**XMLReader**
Record Writer	**CFM Workshop Jso**
Include Zero Record FlowFiles	**true**
Cache Schema	**true**
all	SELECT * FROM FL
cold	SELECT * FROM FL

Schema Write Strategy	**Do Not Write Schema**
Schema Cache	No value set
Schema Access Strategy	**Inherit Record Schema**
Schema Registry	No value set
Schema Name	${schema.name}
Schema Version	No value set
Schema Branch	No value set
Schema Text	${avro.schema}
Date Format	No value set
Time Format	No value set
Timestamp Format	No value set
Pretty Print JSON	**false**
Suppress Null Values	**Never Suppress**
Output Grouping	**Array**

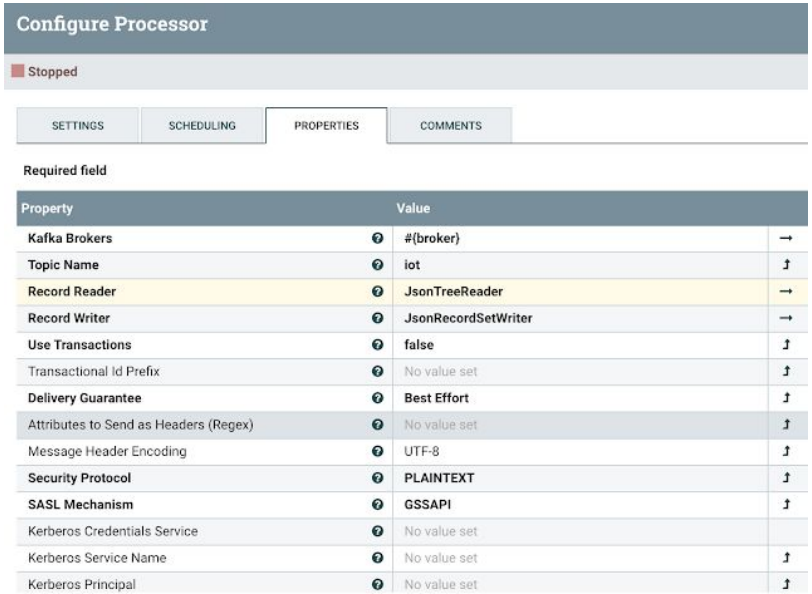
https://dzone.com/articles/lets-build-a-simple-ingest-to-cloud-data-warehouse

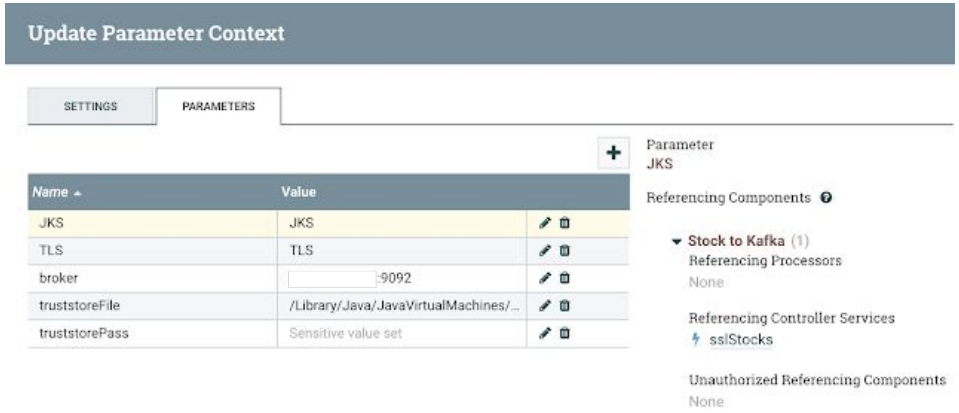
MergeRecord

Property	Value
Record Reader	**JsonTreeReader**
Record Writer	**CFM Workshop JsonRecordSetWriter**
Merge Strategy	**Bin-Packing Algorithm**
Correlation Attribute Name	No value set
Attribute Strategy	**Keep Only Common Attributes**
Minimum Number of Records	**250**
Maximum Number of Records	1000000
Minimum Bin Size	**1000 B**
Maximum Bin Size	No value set
Max Bin Age	1 minutes
Maximum Number of Bins	**10**

https://dzone.com/articles/lets-build-a-simple-ingest-to-cloud-data-warehouse

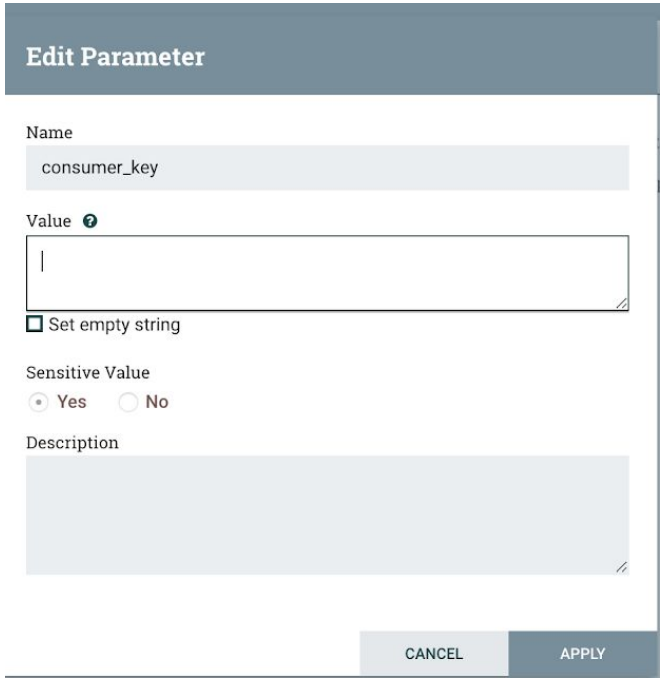
Parameters

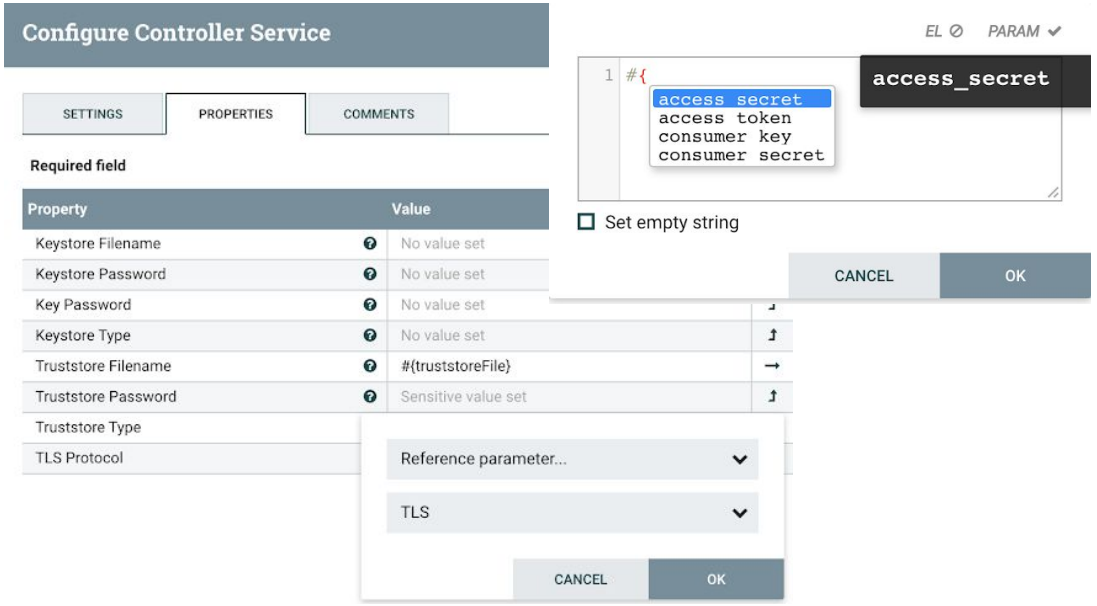




https://dzone.com/articles/exploring-apache-nifi-110-parameters-and-stateless

Parameters



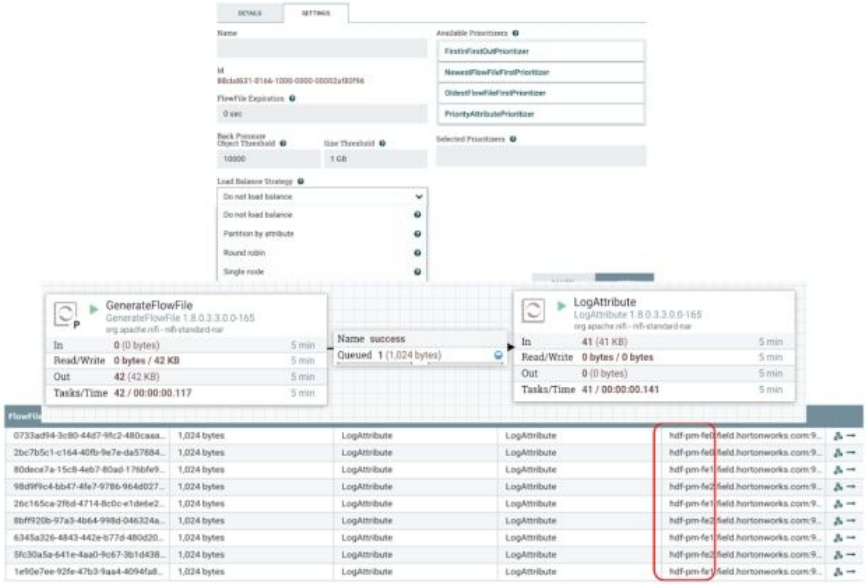


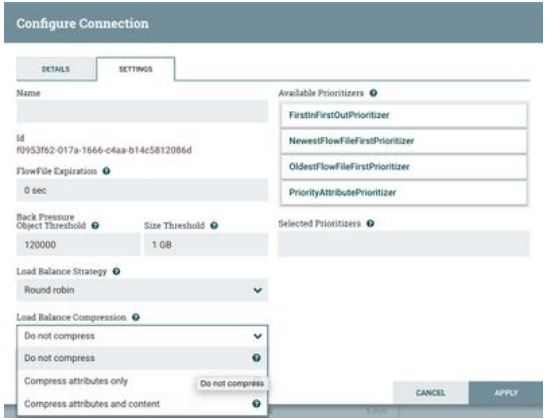
https://dzone.com/articles/exploring-apache-nifi-110-parameters-and-stateless

Load Balancing

NiFi Load Balancing

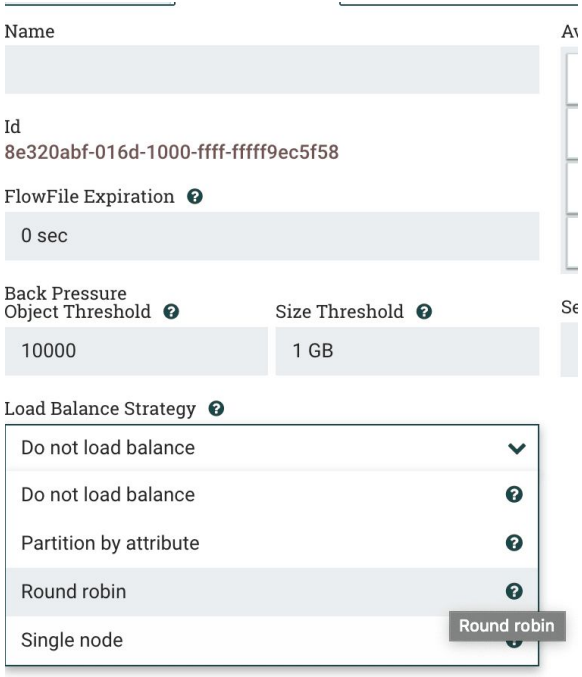
- Improve NiFi cluster throughput
- Defined at connection level
- Configurable balancing strategies
- Critical for scale up and scale down paradigm in Kubernetes
- Alleviates S2S balancing "hack" customers use



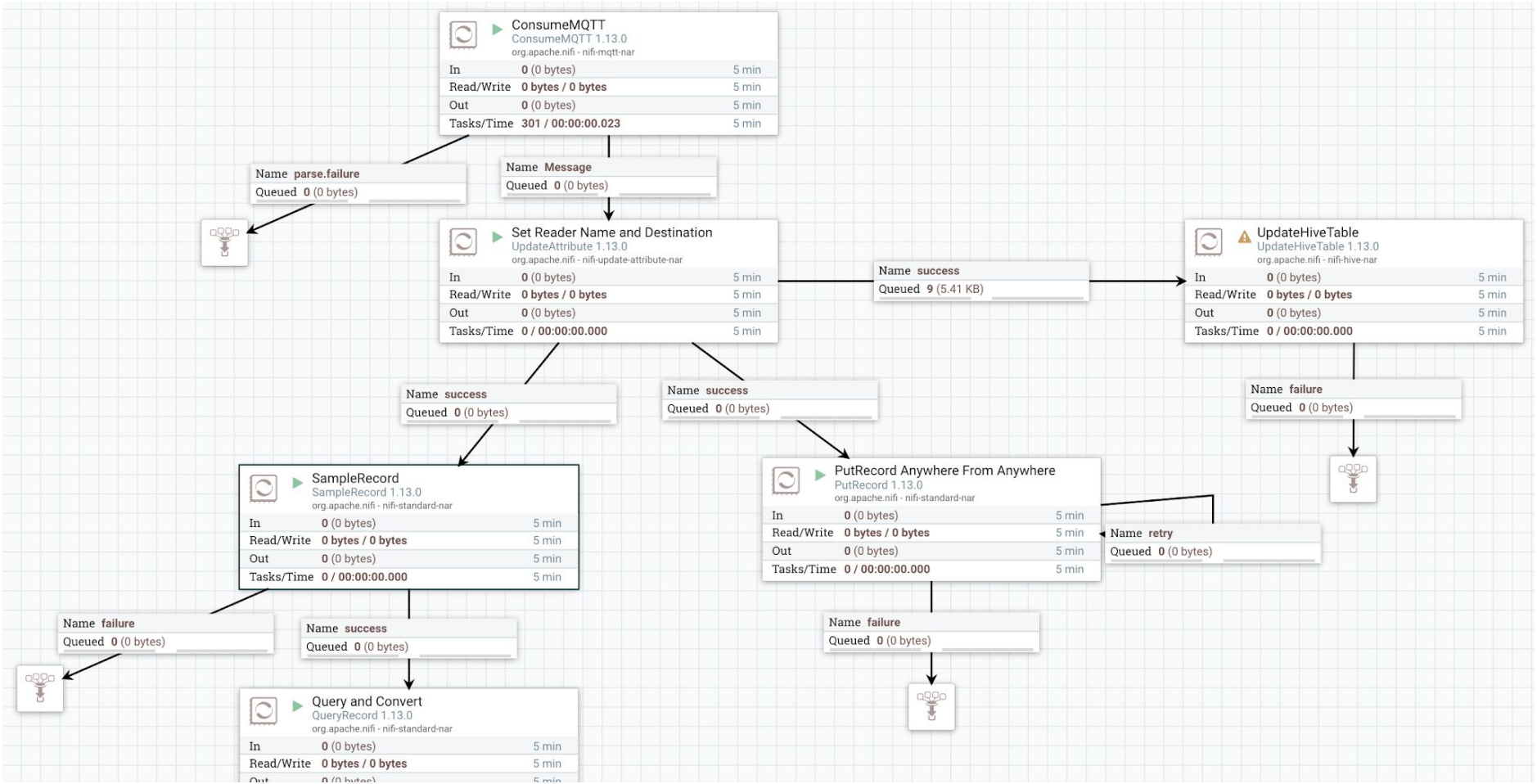


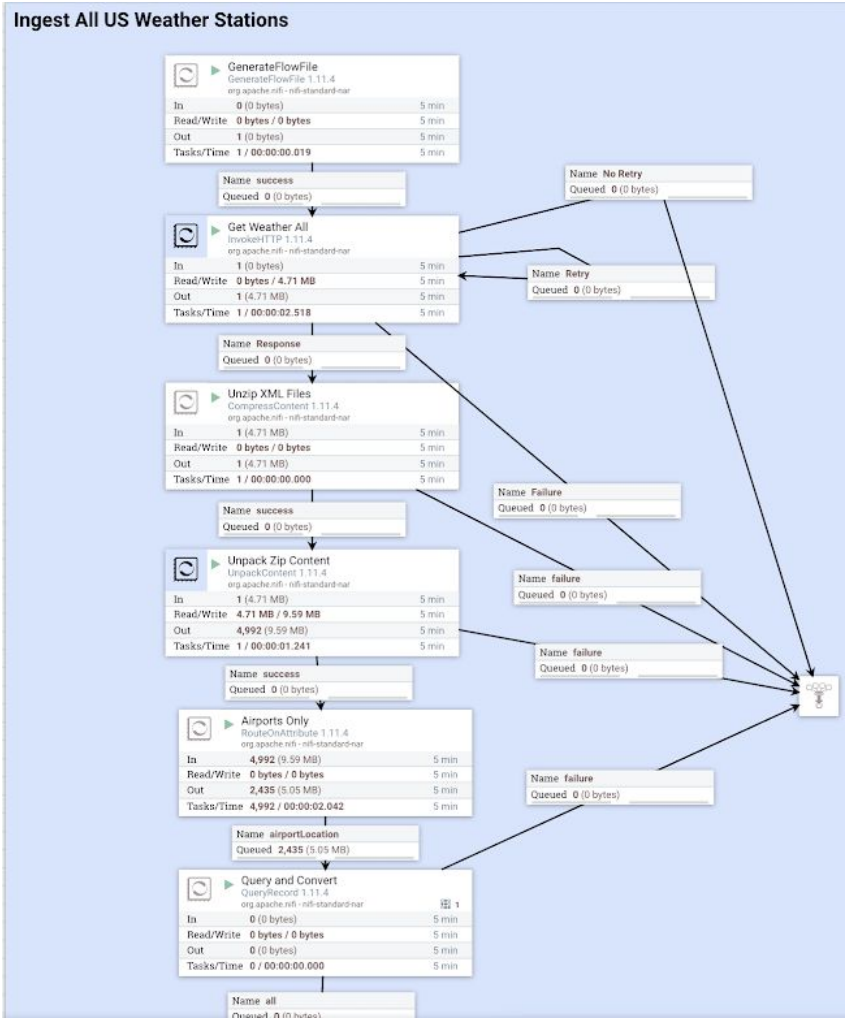
https://dev.to/tspannhw/apache-nifi-load-balancing-via-load-balanced-connections-593m

Load Balancing



https://dev.to/tspannhw/apache-nifi-load-balancing-via-load-balanced-connections-593m


ConsumeMQTT
ConsumeMQTT 1.13.0
org.apache.nifi - nifi-mqtt-nar
In 0 (0 bytes) 5 min
Read/Write 0 bytes / 0 bytes 5 min
Out 0 (0 bytes) 5 min
Tasks/Time 301 / 00:00:00.023 5 min
Name parse.failure
Queued 0 (0 bytes)
Name Message
Queued 0 (0 bytes)
Set Reader Name and Destination
UpdateAttribute 1.13.0
org.apache.nifi - nifi-update-attribute-nar
In 0 (0 bytes) 5 min
Read/Write 0 bytes / 0 bytes 5 min
Out 0 (0 bytes) 5 min
Tasks/Time 0 / 00:00:00.000 5 min
Name success
Queued 9 (5.41 KB)
UpdateHiveTable
UpdateHiveTable 1.13.0
org.apache.nifi - nifi-hive-nar
In 0 (0 bytes) 5 min
Read/Write 0 bytes / 0 bytes 5 min
Out 0 (0 bytes) 5 min
Tasks/Time 0 / 00:00:00.000 5 min
Name success
Queued 0 (0 bytes)
Name success
Queued 0 (0 bytes)
Name failure
Queued 0 (0 bytes)
SampleRecord
SampleRecord 1.13.0
org.apache.nifi - nifi-standard-nar
In 0 (0 bytes) 5 min
Read/Write 0 bytes / 0 bytes 5 min
Out 0 (0 bytes) 5 min
Tasks/Time 0 / 00:00:00.000 5 min
PutRecord Anywhere From Anywhere
PutRecord 1.13.0
org.apache.nifi - nifi-standard-nar
In 0 (0 bytes) 5 min
Read/Write 0 bytes / 0 bytes 5 min
Out 0 (0 bytes) 5 min
Tasks/Time 0 / 00:00:00.000 5 min
Name retry
Queued 0 (0 bytes)
Name failure
Queued 0 (0 bytes)
Name success
Queued 0 (0 bytes)
Name failure
Queued 0 (0 bytes)
Query and Convert
QueryRecord 1.13.0
org.apache.nifi - nifi-standard-nar
In 0 (0 bytes) 5 min
Read/Write 0 bytes / 0 bytes 5 min



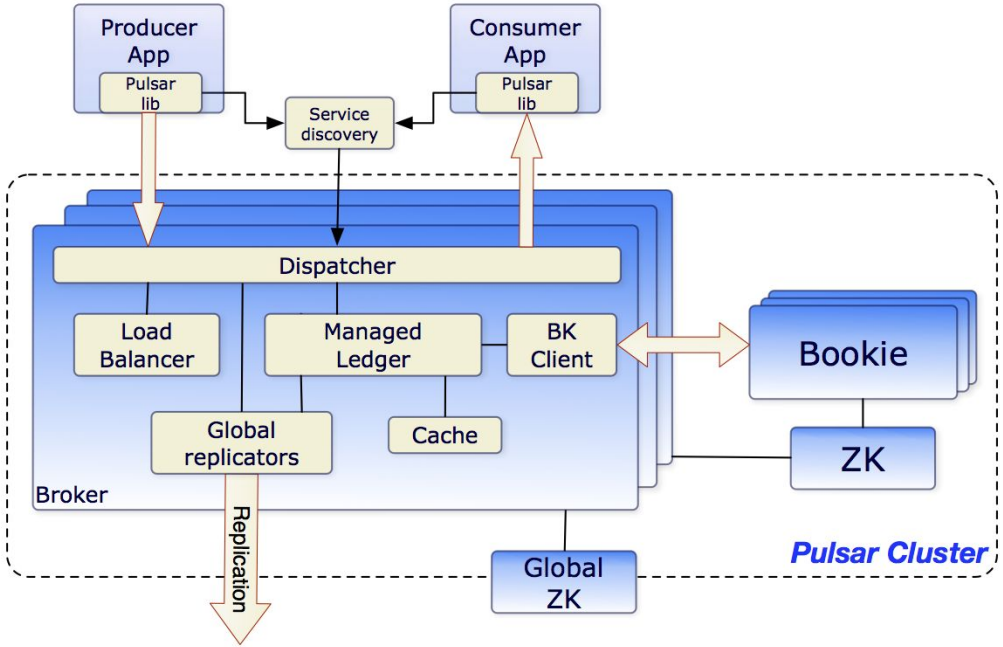
Property	Value
Record Reader	XMLReader
Record Writer	Standard Inherit JsonRecordSetWriter
Include Zero Record FlowFiles	false
Cache Schema	true
all	SELECT * FROM FLOWFILE WHERE NOT location LIKE '%Unknown%'

Property	Value
Routing Strategy	Route to Property name
airportLocation	${filename:startsWith('K')}

https://www.datainmotion.dev/2019/03/advanced-xml-processing-with-apache.html

Apache Pulsar Overview

- Pub-Sub
- Geo-Replication
- Pulsar Functions
- Horizontal Scalability
- Multi-tenancy
- Tiered Persistent Storage
- Pulsar Connectors
- REST API
- CLI
- Many clients available
- Four Different Subscription Types
- Multi-Protocol Support
  - MQTT
  - AMQP
  - JMS
  - Kafka
  - ...



What are the Benefits of Pulsar with NiFi?

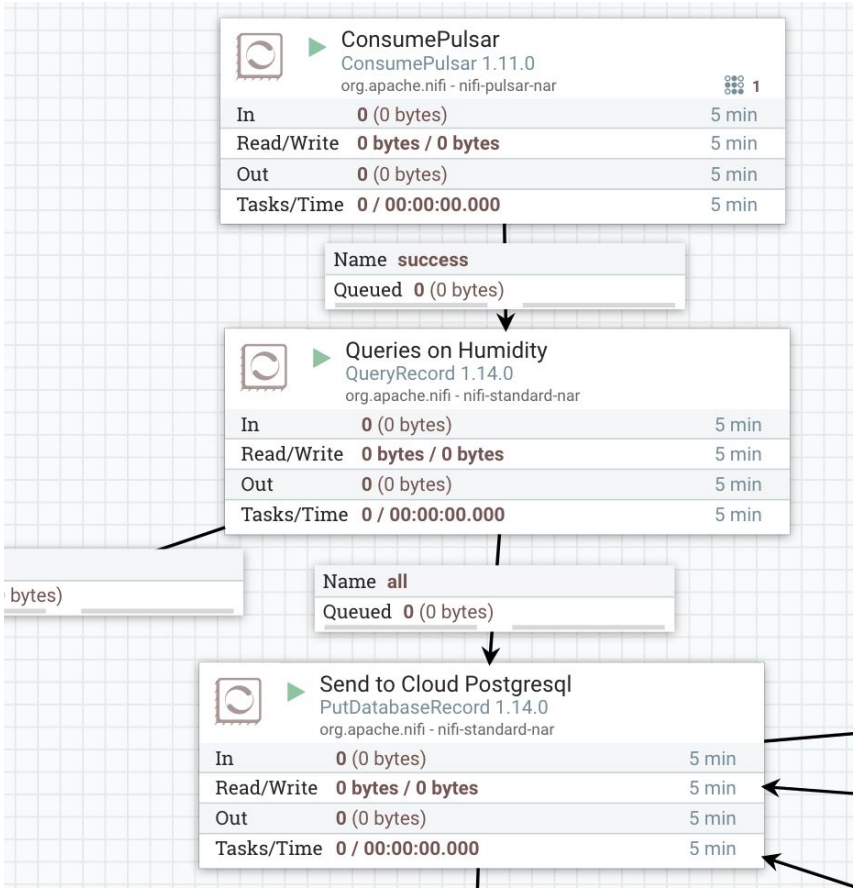
Multi-Tenancy

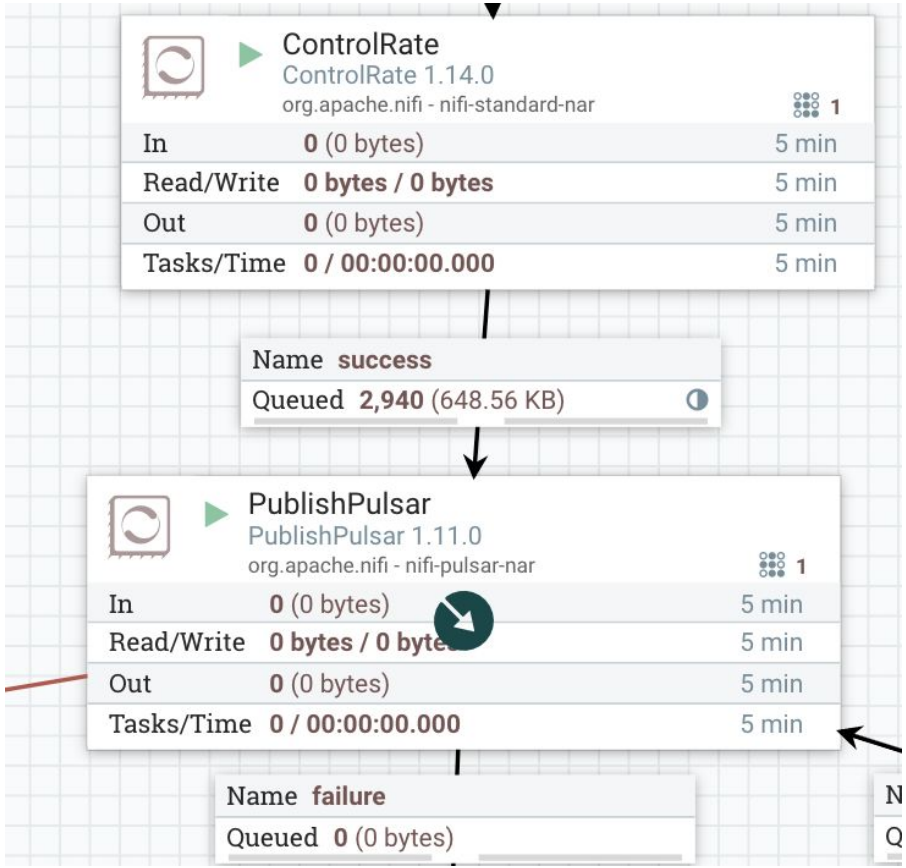
Scalability

Geo-Replication

Unified Messaging Model

Data Durability





https://github.com/david-streamlio/pulsar-nifi-bundle

Deeper Content

- https://www.datainmotion.dev/2020/06/no-more-spaghetti-flows.html
- https://github.com/tspannhw/EverythingApacheNiFi
- https://www.datainmotion.dev/2019/03/apache-nifi-101.html
- https://nifi.apache.org/docs/nifi-docs/html/nifi-in-depth.html
- https://pierrevillard.com/best-of-nifi/
- https://blogs.apache.org/nifi/
- https://www.nifi.rocks/documents/nifi-expression-language-cheat-sheet.pdf
- https://dev.to/tspannhw/new-features-of-apache-nifi-1-13-0-45ln
- https://dev.to/tspannhw/tracking-satellites-with-apache-nifi-44o7
- https://www.datainmotion.dev/2020/07/sizing-your-apache-nifi-cluster-for.html
- https://www.datainmotion.dev/2021/01/flank-using-apache-kudu-as-cache-for.html
- https://www.datainmotion.dev/2020/12/basic-understanding-of-cloudera-flow.html

https://datainmotion.dev/

timothyspann

@PaasDev