



**Stream
Native**

+



influxdata[®]

Using FLiP with InfluxDB for Edge AI IoT at Scale

Tim Spann | Developer Advocate

Tim Spann, Developer Advocate at StreamNative



Tim Spann
Developer Advocate



- FLiP(N) Stack = Flink, Pulsar and NiFi Stack
- Streaming Systems & Data Architecture Expert
- Experience:
 - 15+ years of experience with streaming technologies including Pulsar, Flink, Spark, NiFi, Kafka, Big Data, Cloud, MXNet, IoT and more.
 - Today, he helps to grow the Pulsar community sharing rich technical knowledge and experience at both global conferences and through individual conversations.

CLOUDERA



Pivotal

BARNES
& NOBLE



Hewlett Packard
Enterprise



Agenda

- Learn how to build an end-to-end streaming edge app
- How to pull messages from Pulsar topics and persists the messages to InfluxDB
- Building a data stream for IoT with NiFi and InfluxDB
- Using Apache Flink + Apache Pulsar
- Using Apache Spark + Apache Pulsar





FLiP Stack Weekly

This week in Apache Flink, Apache Pulsar, Apache NiFi, Apache Spark and open source friends.

<https://bit.ly/32dAJft>

FLiP(N)(S) Stack

- Apache Flink
 - Apache Pulsar
 - Pulsar Functions
 - StreamNative's Flink Connector for Pulsar
 - Apache NiFi
 - Apache Spark
 - Python, Java, Golang





Apache Pulsar is a Cloud-Native
Messaging and Event-Streaming Platform.

Pulsar Benefits



Unified Messaging Model

Simplify your data infrastructure and enable new use cases with queuing and streaming capabilities in one platform.



Multi-tenancy

Enable multiple user groups to share the same cluster, either via access control, or in entirely different namespaces.



Scalability

Decoupled data computing and storage enable horizontal scaling to handle data scale and management complexity.



Geo-replication

Support for multi-datacenter replication with both asynchronous and synchronous replication for built-in disaster recovery.

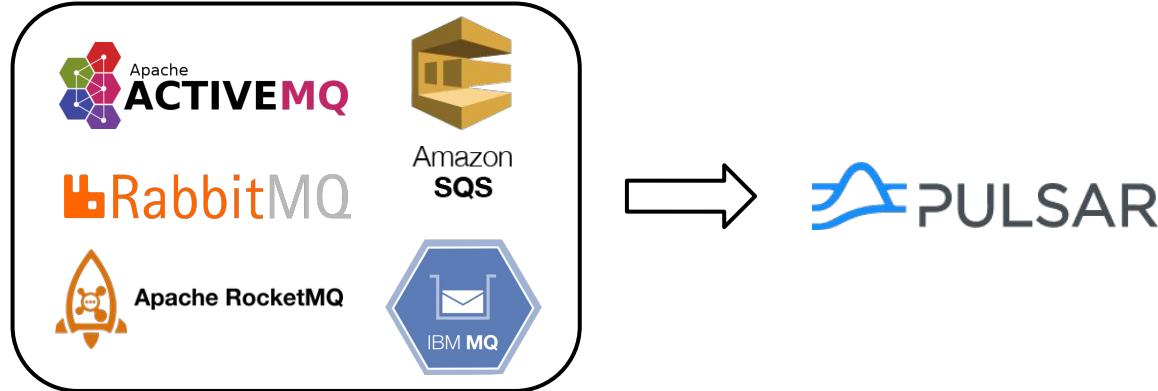


Tiered storage

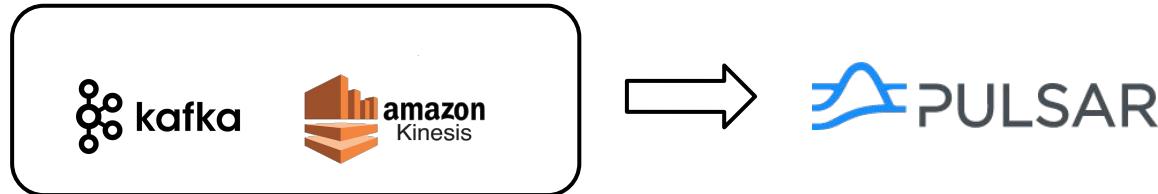
Enable historical data to be offloaded to cloud-native storage and store event streams for indefinite periods of time.

Top Pulsar Use Cases

#1 Message Queuing



#2 Data Streaming



- ✓ Not built for the cloud
- ✓ Single tenant systems
- ✓ Monolithic architecture couples compute with storage
- ✓ Lack of geo replication support

Pulsar: Unified Messaging + Data Streaming

Messaging

Ideal for work queues that do not require tasks to be performed in a particular order—for example, sending one email message to many recipients.

RabbitMQ and [Amazon SQS](#) are examples of popular queue-based message systems.

Pulsar: Unified Messaging + Data Streaming

Messaging

Ideal for work queues that do not require tasks to be performed in a particular order—for example, sending one email message to many recipients.

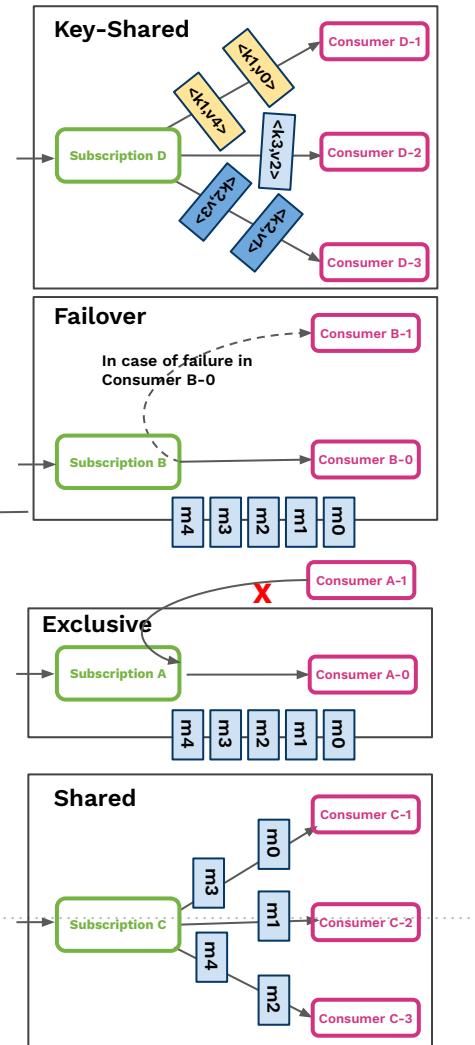
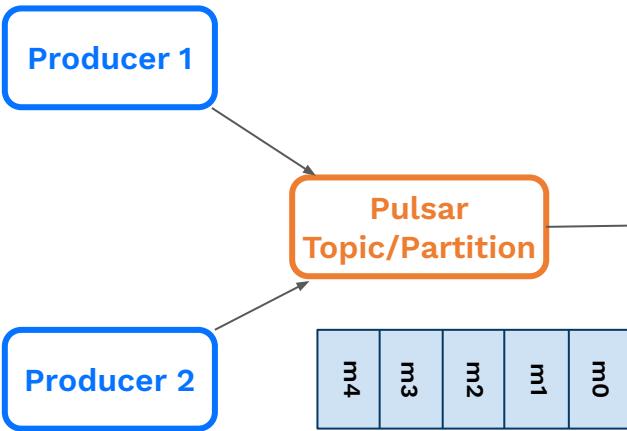
RabbitMQ and Amazon SQS are examples of popular queue-based message systems.

.. and Streaming

Works best in situations where the order of messages is important—for example, data ingestion.

[Kafka](#) and [Amazon Kinesis](#) are examples of messaging systems that use streaming semantics for consuming messages.

Unified Messaging Model



Streaming

Messaging

Use Cases

Multi-Tenant Data
Infrastructure

AdTech

Fraud Detection

Connected Car

IoT Analytics

Microservices Development

Pulsar Global Adoption



splunk®

Flipkart



verizon
media

YAHOO!
JAPAN

Tencent 腾讯

OVHcloud

NUTANIX™

COMCAST

narvar

INSTRUCTURE



BestPAY



ITERABLE

overstock™

中国电信
CHINA TELECOM

tuya.com

toast

clever cloud

IoTiUM

</ndustrial.rio>

mercado
libre

OKCOIN

360
www.360.cn

proxyclick

BrandsEye

Huya

THE HUT GROUP®

Giggs

Kingsoft Cloud
kscloud

Max Kelsen

环球易购
Globegrow E-Commerce

VIP KID

EMQ

5G+

中国移动
China Mobile

Pulsar Adoption Use Cases



Adopted Pulsar to **replace Kafka in their DSP** (Data Streaming Platform).

- 1.5-2x lower in capex cost
- 5-50x improvement in latency
- 2-3x lower in opex due



Adopted Pulsar to **power their billing platform**, Midas, which processing hundreds of billions of financial transactions daily.

Adoption then expanded to Tencent's Federated Learning Platform and Tencent Gaming.



Applied Materials is one of the biggest semiconductor hardware and software supplier in the industry.

They adopted Pulsar to enable them to **build a message bus to tie all of their data together**. They previously used Tibco.



Founded by the original developers of Apache Pulsar.

Passionate and dedicated team.

StreamNative helps teams to **capture**, **manage**, and **leverage data** using Pulsar's unified messaging and streaming platform.

streamnative.io

Founded By The Creators Of Apache Pulsar



Sijie Guo

Founder and CEO

ASF Member
Pulsar/BookKeeper PMC



Matteo Merli

CTO

ASF Member
Pulsar/BookKeeper PMC



Jia Zhai

Co-Founder

Pulsar/BookKeeper PMC

- ✓ Original creators of Apache Pulsar & BookKeeper
- ✓ Operated the largest Pulsar/BookKeeper cluster
- ✓ Data veterans with extensive industry experience

StreamNative Cloud

Powered by Apache Pulsar, StreamNative provides a cloud-native, real-time messaging and streaming platform to support multi-cloud and hybrid cloud strategies.



Cloud-Native



kubernetes

Built for Containers

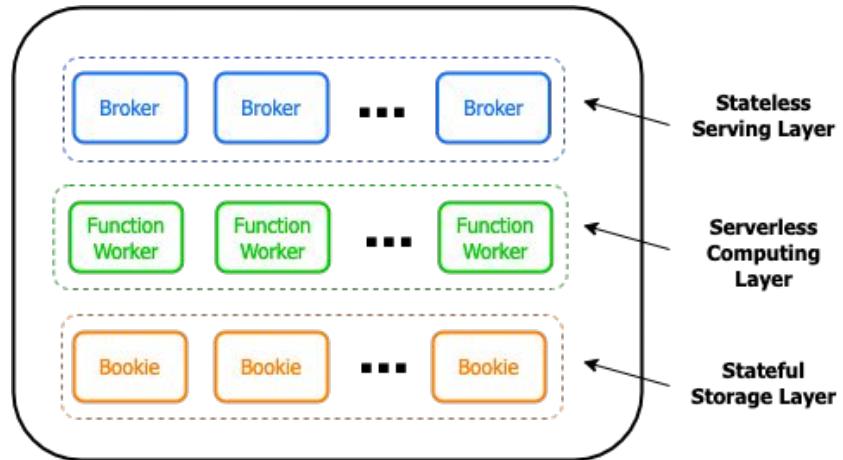


Flink

Flink SQL

Apache Pulsar

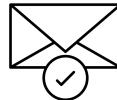
- Serverless computing framework.
- Unbounded storage, multi-tiered architecture, and tiered-storage.
- Streaming & Pub/Sub messaging semantics.
- Multi-protocol support.



Why Apache Pulsar?



Unified
Messaging
Platform



Guaranteed
Message
Delivery



Resiliency

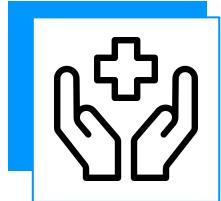


Infinite
Scalability



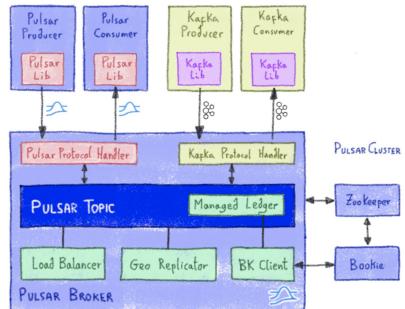
StreamNative + *influxdata*

streamnative.io

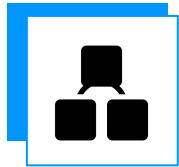


Connectivity

hub.streamnative.io



- **Functions** - Lightweight Stream Processing (Java, Python, Go)
- **Connectors** - Sources & Sinks (InfluxDB, Kafka, S3, Kinesis, Lambda, ...)
- **Protocol Handlers** - AoP (AMQP), KoP (Kafka), MoP (MQTT)
- **Processing Engines** - Flink, Spark, Presto/Trino via Pulsar SQL
- **Data Offloaders** - Tiered Storage - (S3)



Use Cases

- Unified Messaging Platform
- AdTech
- Fraud Detection
- Connected Car
- IoT Analytics
- Microservices Development

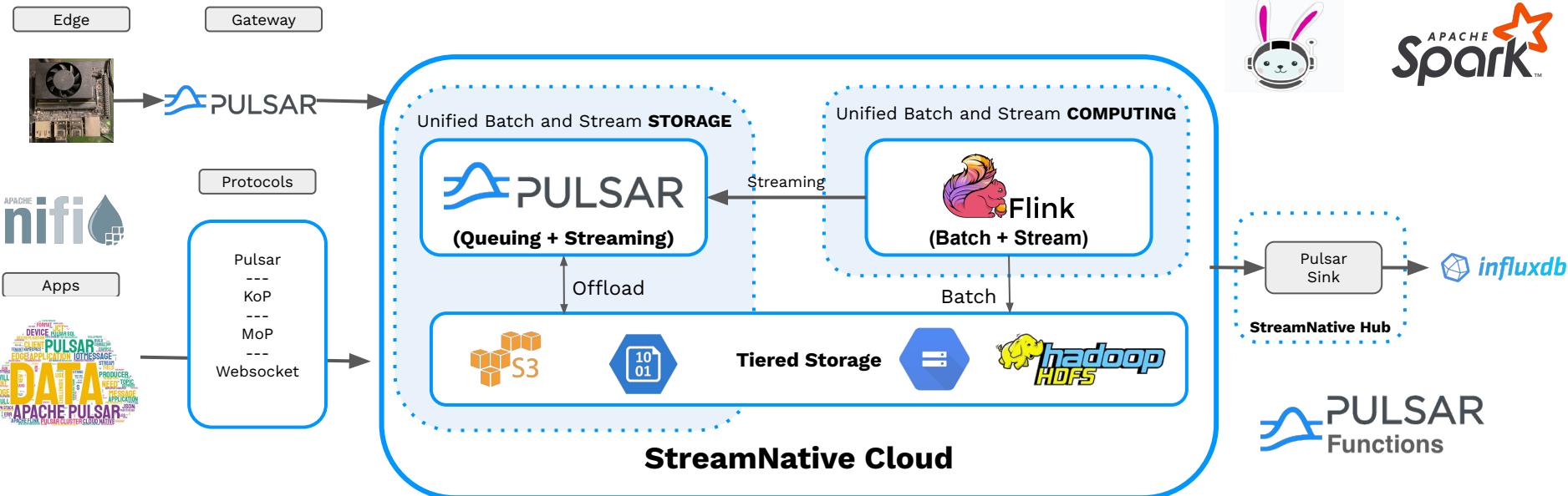
Streaming Edge Apps



<-> Sensors <->



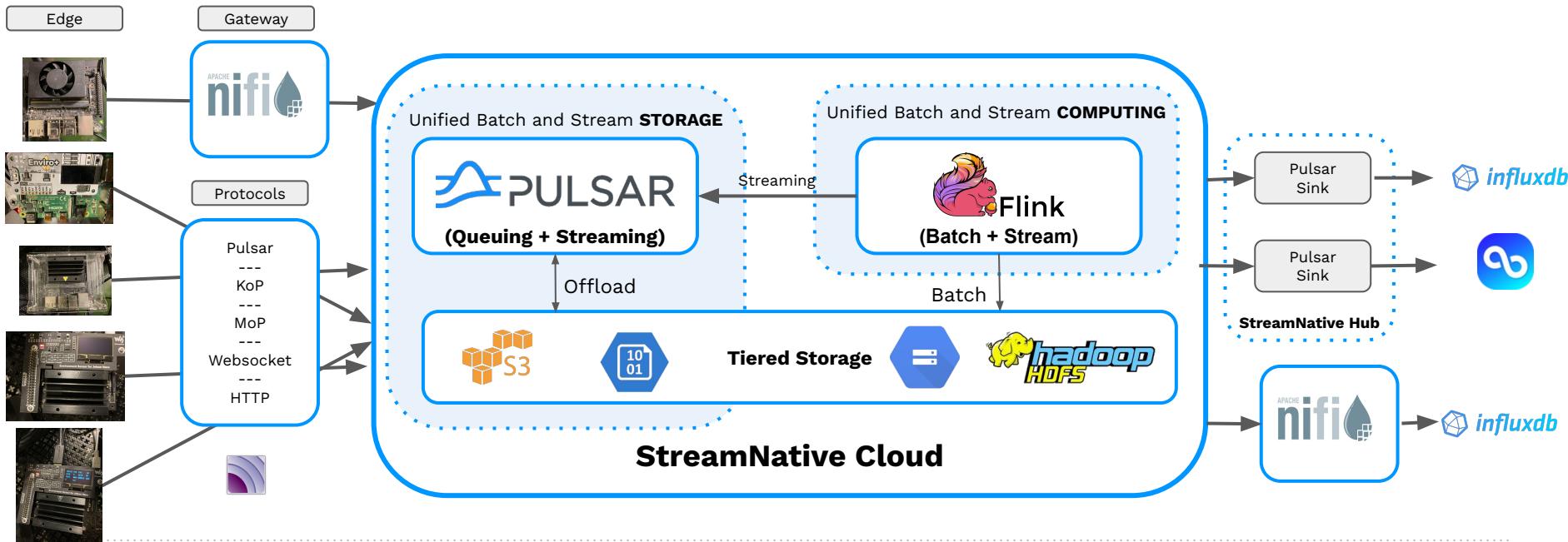
<-> Sensors <->



StreamNative + influxdata®

End-to-End Streaming Edge App

Apache Flink - Apache Pulsar - Apache NiFi <-> Devices <-> influxdb



Why



StreamNative + *influxdata*[®]



Analytics

Getting Started

1



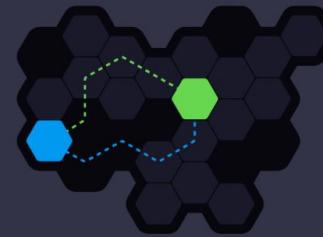
LOAD YOUR DATA

2



BUILD A DASHBOARD

3



SET UP ALERTING



StreamNative + *influxdata*[®]



influxdb

Easy Visualization

Visualize the Result

Limited to most recent 100 results per series

Simple Table CONFIGURE RUN ...

table	_measurement	_field	_value	_time	host	topic
_RESULT	GROUP STRING	GROUP STRING	NO GROUP DOUBLE	NO GROUP DATETIME:RFC3339	GROUP STRING	GROUP STRING
2	sensors	value	46.94	2022-01-14T20:13:52.263Z	pulsar1	sensors
2	sensors	value	46.94	2022-01-14T20:13:55.278Z	pulsar1	sensors
2	sensors	value	46.94	2022-01-14T20:13:58.290Z	pulsar1	sensors
2	sensors	value	46.94	2022-01-14T20:14:01.304Z	pulsar1	sensors
2	sensors	value	46.94	2022-01-14T20:14:04.318Z	pulsar1	sensors
2	sensors	value	46.94	2022-01-14T20:14:07.334Z	pulsar1	sensors
2	sensors	value	46.94	2022-01-14T20:14:10.350Z	pulsar1	sensors
2	sensors	value	46.94	2022-01-14T20:14:13.362Z	pulsar1	sensors

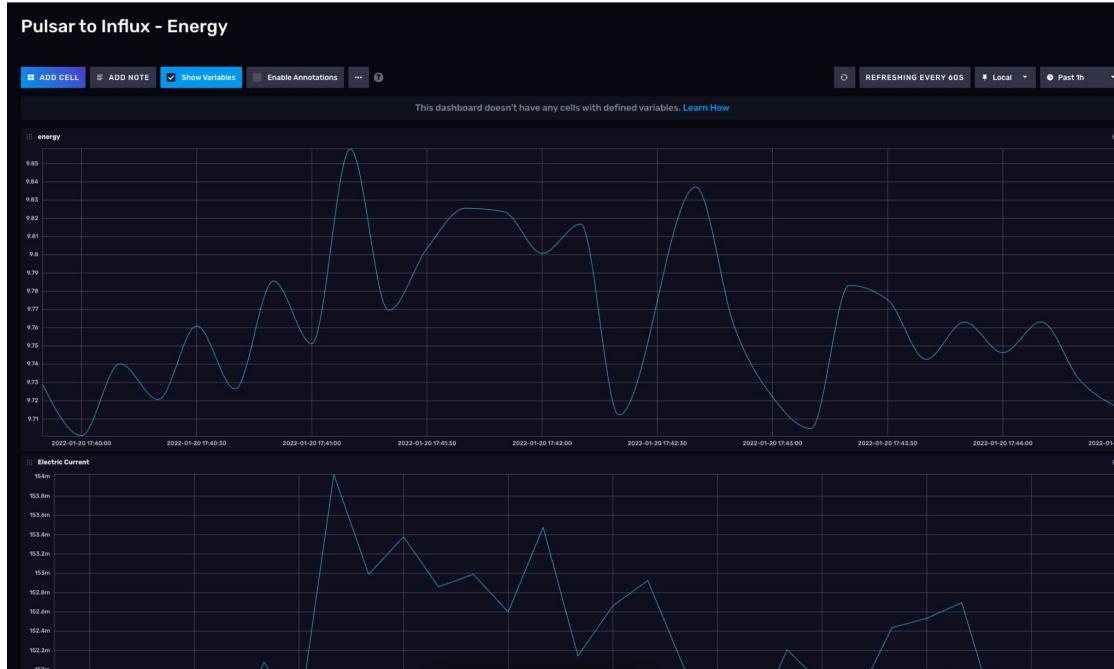


StreamNative + *influxdata*[®]



influxdb

Analytics



StreamNative + *influxdata*[®]



Analytics

▶ RUN SAVE NOTEBOOK

Presentation UTC Past 15m

Build a Query

FROM

Search for a bucket
pulsar
tim.spann's Bucket
SYSTEM
_monitoring
tasks
SAMPLE
Air Sensor Data

FILTER

_measurement
Search _measurement tag
current

FILTER

_field
Search _field tag values
power

FILTER

host
Search host tag values
pulsar1

FILTER

os
Search os tag values
linux

FILTER

topic
Search topic tag values
persistent://public/default/energy-influx

No tag keys found in the current time range

Validate the Data

Limited to most recent 100 results per series

RUN

result	_measurement	_field	_value	_time	host	os	topic
0	current	power	9.71197	2022-01-20T22:27:02.693Z	pulsar1	linux	persistent://public/default/energy-influx
0	current	power	9.721268	2022-01-20T22:27:02.760Z	pulsar1	linux	persistent://public/default/energy-influx
0	current	power	9.743162	2022-01-20T22:27:02.796Z	pulsar1	linux	persistent://public/default/energy-influx
0	current	power	9.759187	2022-01-20T22:27:02.826Z	pulsar1	linux	persistent://public/default/energy-influx
0	current	power	9.795259	2022-01-20T22:27:02.857Z	pulsar1	linux	persistent://public/default/energy-influx
0	current	power	9.692648	2022-01-20T22:27:02.887Z	pulsar1	linux	persistent://public/default/energy-influx
0	current	power	9.759846	2022-01-20T22:27:02.913Z	pulsar1	linux	persistent://public/default/energy-influx
0	current	power	9.801128	2022-01-20T22:27:02.936Z	pulsar1	linux	persistent://public/default/energy-influx



StreamNative + *influxdata*[®]

Apache Pulsar InfluxDB Sink

<https://pulsar.apache.org/docs/en/io-influxdb-sink/>



Apache Pulsar InfluxDB Sink

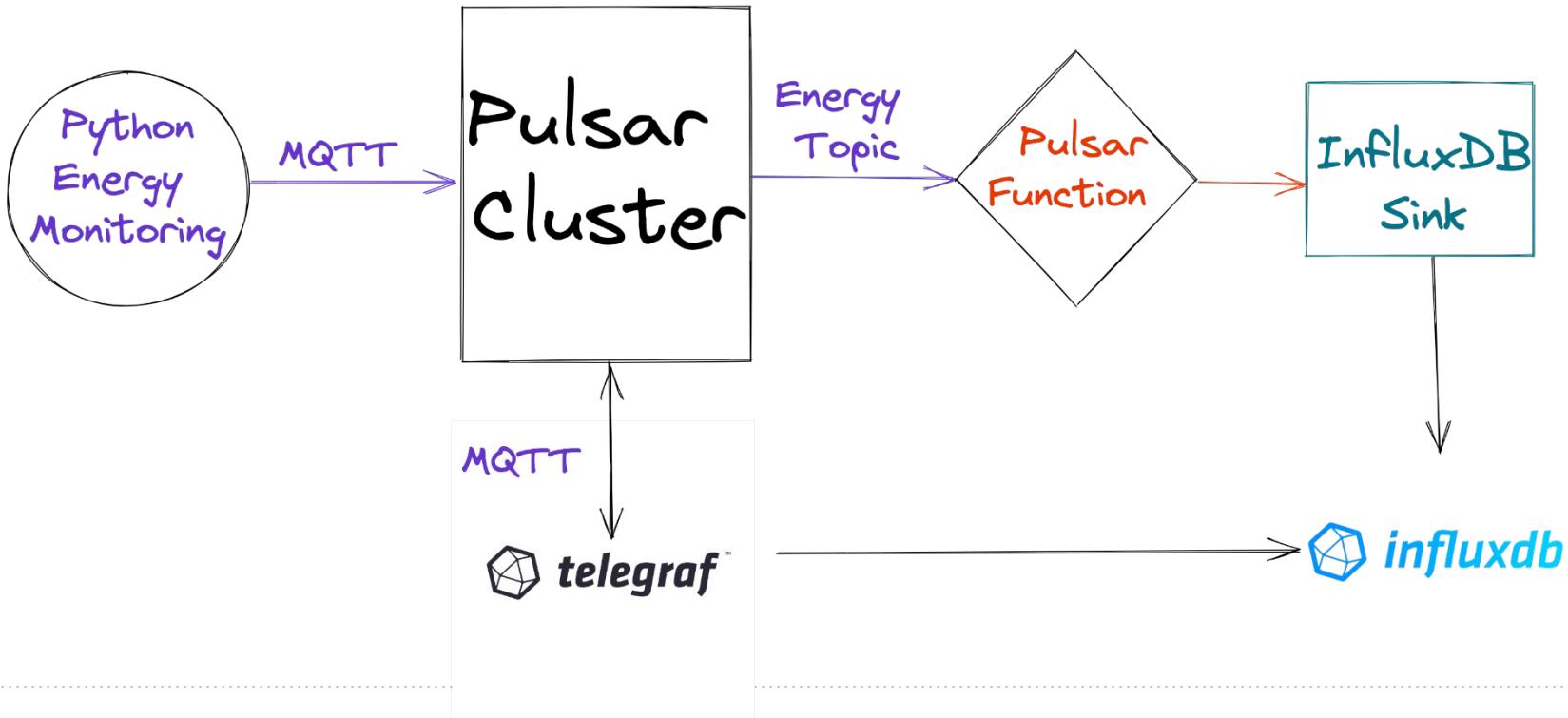
```
pulsar-admin sinks create --archive  
./connectors/pulsar-io-influxdb-2.9.1.nar  
--name inflx-jtsn  
--sink-config-file conf/influxcloud.yml  
--inputs jtsninflix  
  
pulsar-admin sinks get --name inflx-jtsn  
pulsar-admin sinks status --name inflx-jtsn
```



StreamNative +  *influxdata*[®]

streamnative.io

Apache Pulsar InfluxDB Sink



InfluxDB Telegraf MQTT

<https://www.influxdata.com/integration/mqtt-monitoring/>

MQTT on Pulsar (MoP)



<https://github.com/streamnative/mop>



InfluxDB Telegraf MQTT

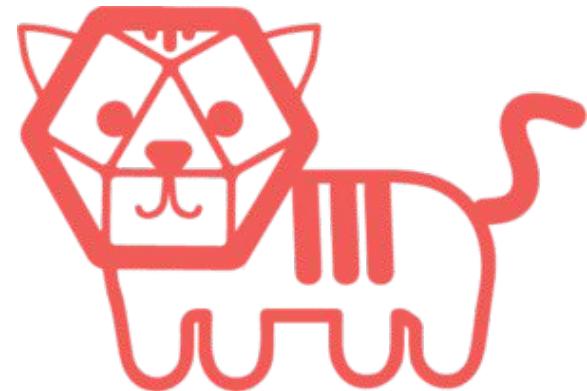


Load Data

SOURCES BUCKETS TELEGRAF API TOKENS

Filter telegraf configurations... Sort by Name (A → Z) INFLUXDB OUTPUT PLUGIN

mqttconsumer
No description
Bucket: tim.spann's Bucket Setup Instructions
+ Add a label

A screenshot of the Telegraf configuration interface showing a single configuration named "mqttconsumer". The configuration is described as an "INFLUXDB OUTPUT PLUGIN". It has no description and is associated with the bucket "tim.spann's Bucket". There is a link to "Setup Instructions" and a button to "Add a label".

InfluxDB Telegraf MQTT

```
● ● ●

# Read metrics from MQTT topic(s)
[[inputs.mqtt_consumer]]
    ## Broker URLs for the MQTT server or cluster. To connect to multiple
    ## clusters or standalone servers, use a separate plugin instance.
    ##   example: servers = ["tcp://localhost:1883"]
    ##           servers = ["ssl://localhost:1883"]
    ##           servers = ["ws://localhost:1883"]
servers = ["tcp://192.168.1.230:1883"]

    ## Topics that will be subscribed to.
topics = [
    "telegrafcpu",
    "telegrafmem",
    "sensors",
]
```



InfluxDB Telegraf MQTT



```
export INFLUX_TOKEN=INFLUXCLOUDTOKENGGENERATED  
telegraf --config https://us-east-1-1.aws.cloud2.influxdata.com/api/v2/telegrafs/code
```



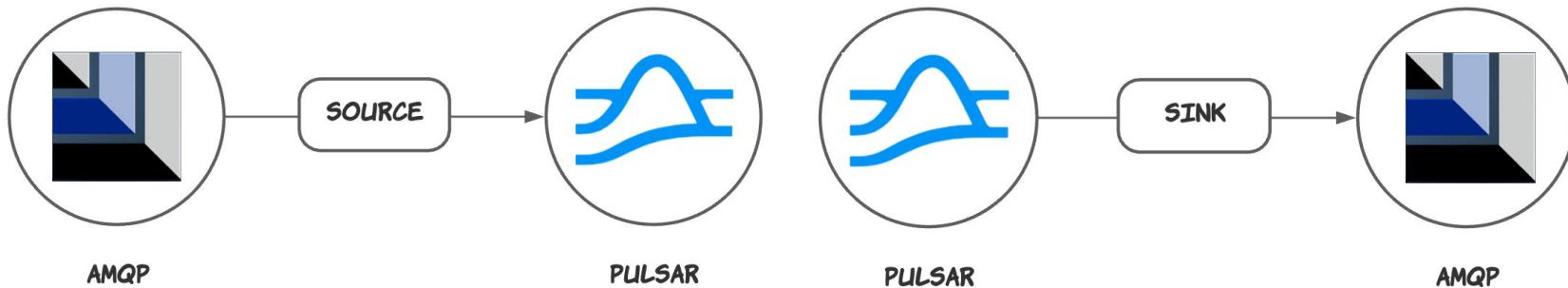
InfluxDB Telegraf RabbitMQ

AMQP on Pulsar (AoP)

<https://github.com/streamnative/aop>

<https://hub.streamnative.io/connectors/amqp-1-0-sink/v2.7.1.1>

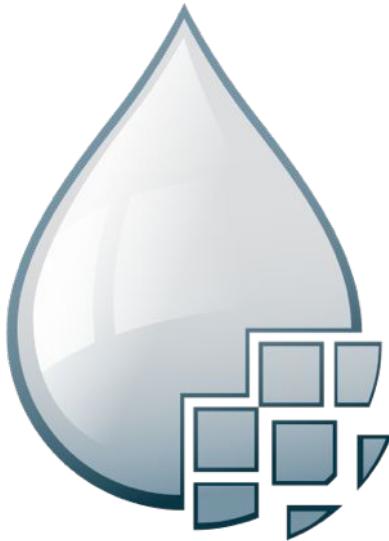
<https://hub.streamnative.io/connectors/amqp-1-0-source/v2.7.1.1>



StreamNative + *influxdata*[®]

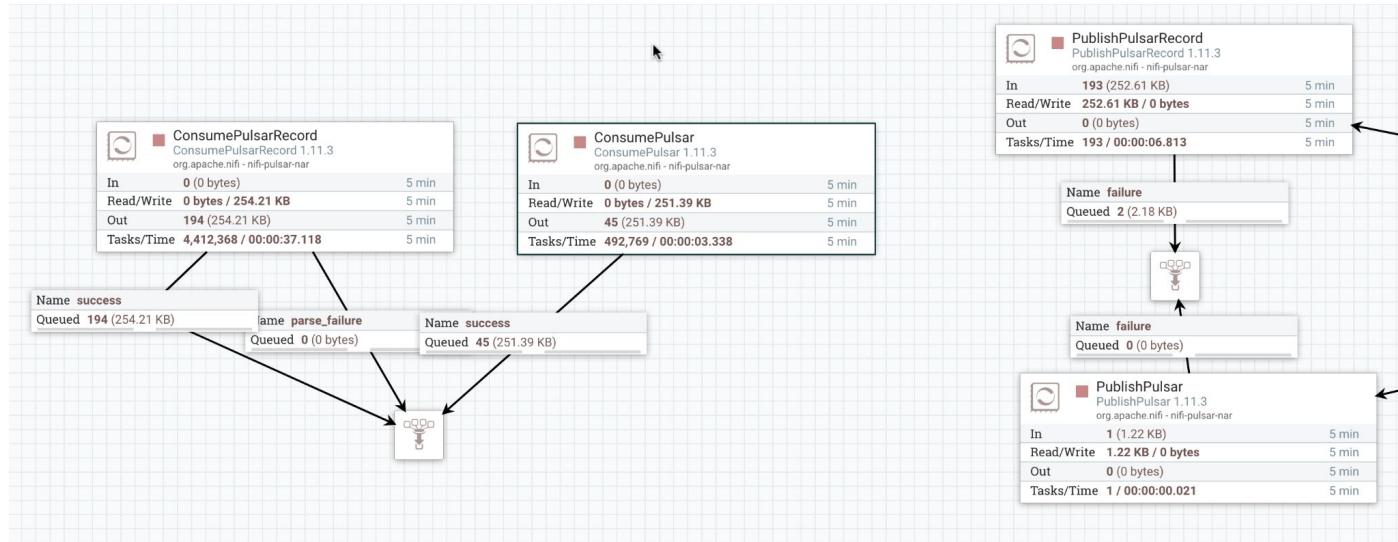
streamnative.io

Why Apache NiFi?



- Guaranteed delivery
- Data buffering
 - Backpressure
 - Pressure release
- Prioritized queuing
- Flow specific QoS
 - Latency vs. throughput
 - Loss tolerance
- Data provenance
- Supports push and pull models
- Hundreds of processors
- Visual command and control
- Over a 300 sources
- Flow templates
- Pluggable/multi-role security
- Designed for extension
- Clustering
- Version Control

Apache NiFi Pulsar Connector



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

Apache NiFi Pulsar Connector

Displaying 12 of 339

Type	Version ▾	Tags
ConsumePulsar	1.11.0	PubSub, Consume, Ingest, Get, I...
ConsumePulsarRecord	1.11.0	PubSub, Consume, Ingest, Get, ...
PublishPulsar	1.11.0	PubSub, Message, Pulsar, Apac...
PublishPulsarRecord	1.11.0	PubSub, 1.0, Message, csv, json...
ConsumePulsar	1.11.3	PubSub, Consume, Ingest, Get, I...
ConsumePulsarRecord	1.11.3	PubSub, Consume, Ingest, Get, ...
PublishPulsar	1.11.3	PubSub, Message, Pulsar, Apac...
PublishPulsarRecord	1.11.3	PubSub, 1.0, Message, csv, json...
ConsumePulsar	1.14.0	PubSub, Consume, Ingest, Get, I...
ConsumePulsarRecord	1.14.0	PubSub, Consume, Ingest, Get, ...
PublishPulsar	1.14.0	PubSub, Message, Pulsar, Apac...
PublishPulsarRecord	1.14.0	PubSub, 1.0, Message, csv, json...

ConsumePulsar 1.11.3 org.apache.nifi - nifi-pulsar-nar

Consumes messages from Apache Pulsar. The complementary NiFi processor for sending messages is PublishPulsar.

Apache NiFi Pulsar Connector

Controller Service Details

SETTINGS PROPERTIES COMMENTS

Required field

Property	Value
Pulsar Service URL	pulsar+ssl://gke.sndev.snio.cloud:6651
Pulsar Client Authentication Service	PulsarClientOauthAuthenticationService14sn →
Maximum concurrent lookup-requests	5000
Maximum connects per Pulsar broker	1
I/O Threads	1
Keep Alive interval	30 sec
Listener Threads	1
Maximum lookup requests	50000
Maximum rejected requests per connection	50
Operation Timeout	30 sec
Stats interval	60 sec
Allow TLS Insecure Connection	false
Enable TLS Hostname Verification	false
Use TCP no-delay flag	false



StreamNative + *influxdata*[®]

streamnative.io

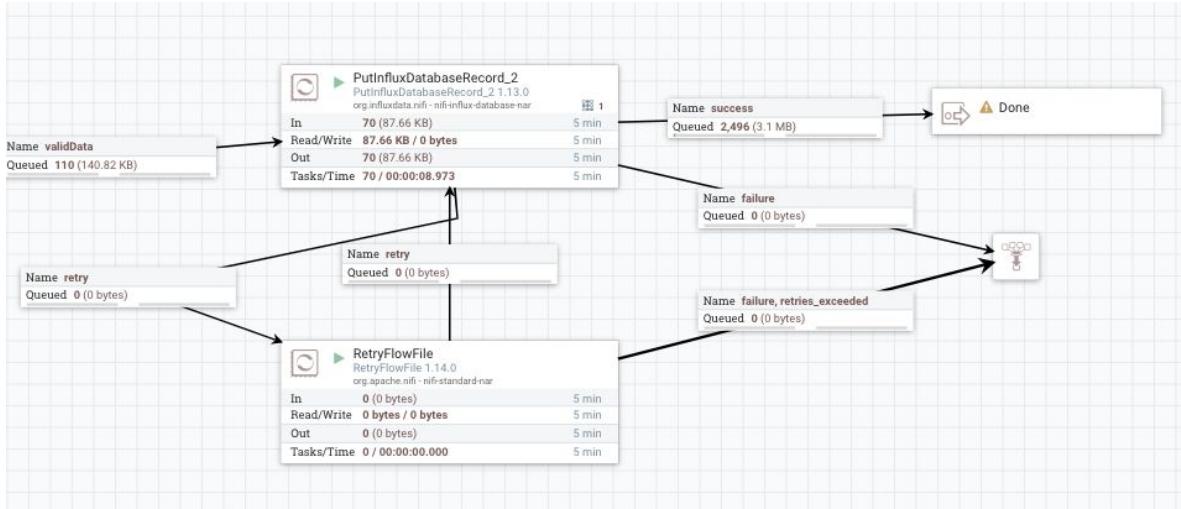
Apache NiFi Pulsar Connector

Controller Service Details

SETTINGS	PROPERTIES	COMMENTS															
Required field																	
<table><thead><tr><th>Property</th><th>Value</th><th></th></tr></thead><tbody><tr><td>Audience</td><td>urn:sn:pulsar:sndev:gke</td><td></td></tr><tr><td>Issuer URL</td><td>https://auth.streamnative.cloud</td><td></td></tr><tr><td>Private key file</td><td>file:///Users/tspann/Documents/documents/services/apache-pulsar-2.8.0/sndev-tspann.json</td><td></td></tr><tr><td>Trusted Certificate Filename</td><td>?</td><td>No value set</td></tr></tbody></table>			Property	Value		Audience	urn:sn:pulsar:sndev:gke		Issuer URL	https://auth.streamnative.cloud		Private key file	file:///Users/tspann/Documents/documents/services/apache-pulsar-2.8.0/sndev-tspann.json		Trusted Certificate Filename	?	No value set
Property	Value																
Audience	urn:sn:pulsar:sndev:gke																
Issuer URL	https://auth.streamnative.cloud																
Private key file	file:///Users/tspann/Documents/documents/services/apache-pulsar-2.8.0/sndev-tspann.json																
Trusted Certificate Filename	?	No value set															



InfluxDB via Apache NiFi

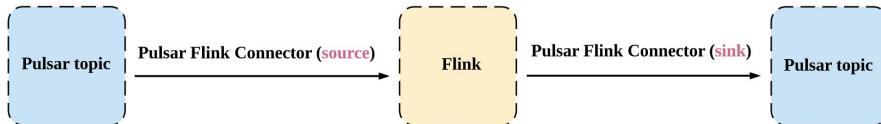


<https://github.com/influxdata/nifi-influxdb-bundle>

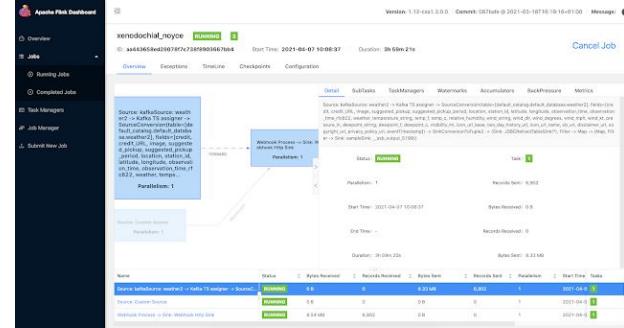
<https://www.influxdata.com/blog/building-a-data-stream-for-iot-with-nifi-and-influxdb/>



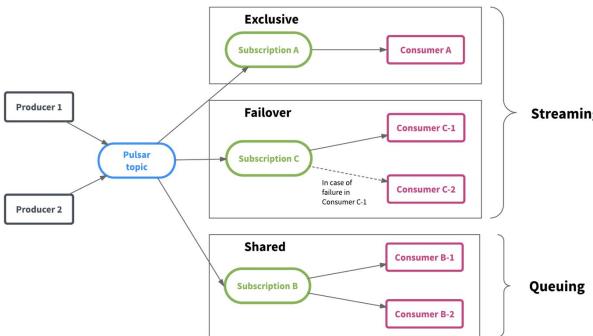
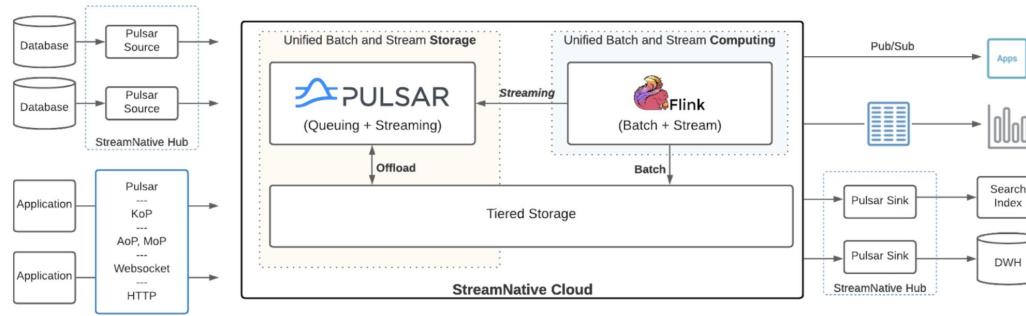
Why Apache Flink?



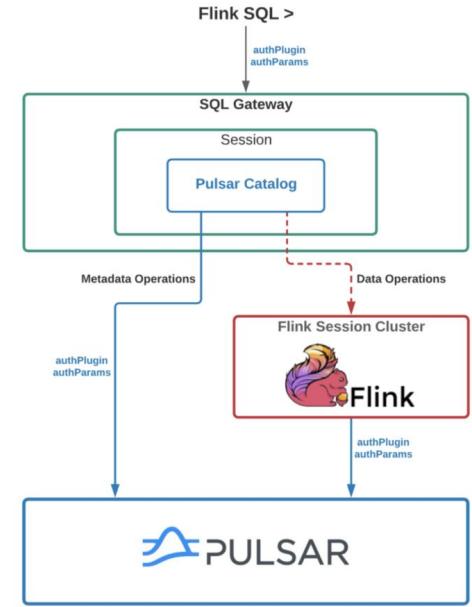
- Unified computing engine
- Batch processing is a special case of stream processing
- Stateful processing
- Massive Scalability
- Flink SQL for queries, inserts against Pulsar Topics
- Streaming Analytics
- Continuous SQL
- Continuous ETL
- Complex Event Processing
- Standard SQL Powered by Apache Calcite



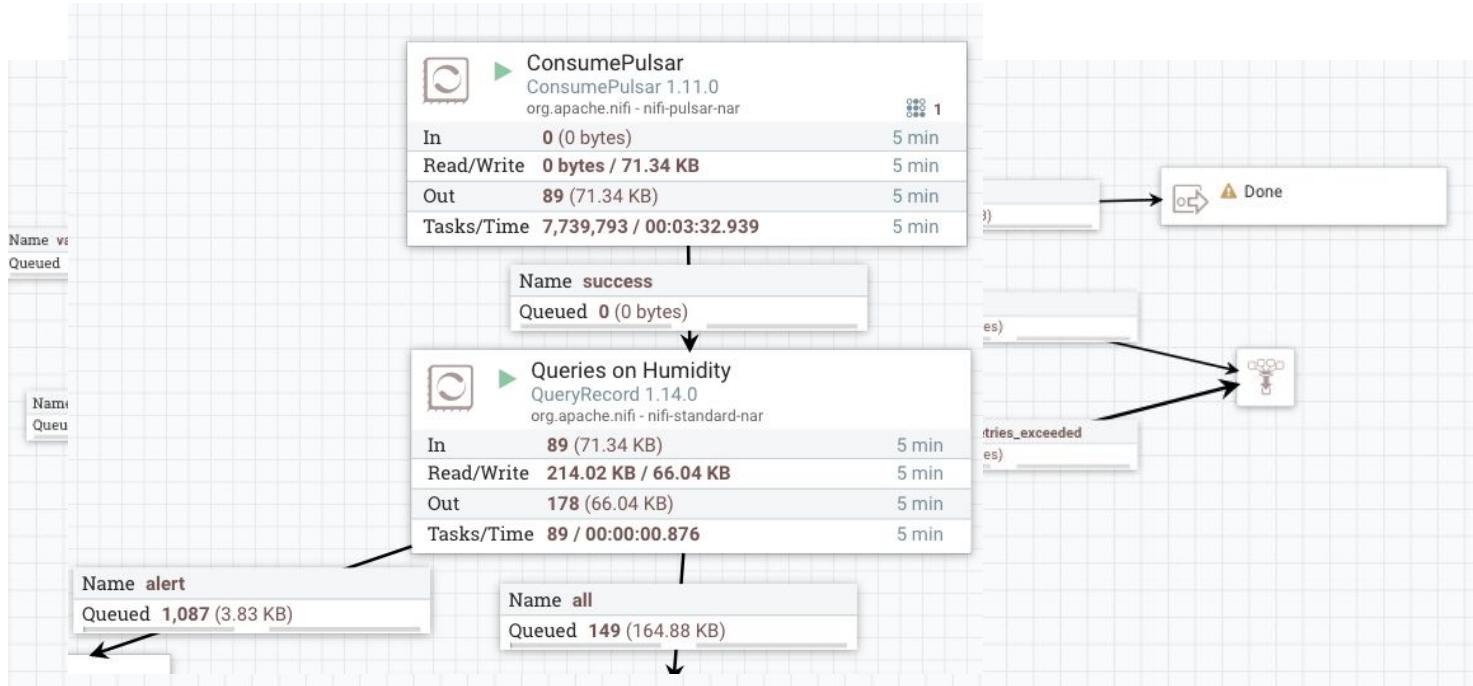
Flink + Pulsar



<https://flink.apache.org/2019/05/03/pulsar-flink.html>
<https://github.com/streamnative/pulsar-flink>
<https://streamnative.io/en/blog/release/2021-04-20-flink-sql-on-streamnative-cloud>



Apache NiFi



Apache Pulsar

```
tspann@Timothys-mbp apache-pulsar-2.8.0 % ./influxstatus.sh
{
  "numInstances" : 1,
  "numRunning" : 1,
  "instances" : [ {
    "instanceId" : 0,
    "status" : {
      "running" : true,
      "error" : "",
      "numRestarts" : 0,
      "numReadFromPulsar" : 12,
      "numSystemExceptions" : 0,
      "latestSystemExceptions" : [ ],
      "numSinkExceptions" : 0,
      "latestSinkExceptions" : [ ],
      "numWrittenToSink" : 12,
      "lastReceivedTime" : 1633550560478,
      "workerId" : "c-standalone-fw-127.0.0.1-8080"
    }
  } ]
}
```

StreamNative Cloud

The screenshot displays the StreamNative Cloud interface with five cluster cards:

- aws** (Ready): Overview shows 5 Tenants, 4 Namespaces, 7 Topics, 0 Producers, 0 Subscriptions. Usage shows In Rate 0, Out Rate 0, In Throughput 0, Out Throughput 3.93 Bytes, Storage Size 0.
- gke** (Ready): Overview shows 3 Tenants, 4 Namespaces, 20 Topics, 2 Producers, 8 Subscriptions. Usage shows In Rate 0.09, Out Rate 0.07, In Throughput 4.91 Bytes, Out Throughput 3.93 Bytes, Storage Size 1.54 MB.
- gke-free** (Ready): Overview shows 3 Tenants, 3 Namespaces, 1 Topics, 0 Producers, 0 Subscriptions. Usage shows In Rate 0, Out Rate 0, In Throughput 0, Out Throughput 0, Storage Size 0.
- nyc** (Ready): Overview shows 3 Tenants, 3 Namespaces, 3 Topics, 2 Producers, 1 Subscriptions. Usage shows In Rate 0.09, Out Rate 0.07, In Throughput 5.07 Bytes, Out Throughput 4.07 Bytes, Storage Size 76.5 KB.
- sgconnector** (Ready): Overview shows 3 Tenants, 3 Namespaces, 9 Topics, 2 Producers, 1 Subscriptions. Usage shows In Rate 0.09, Out Rate 0.07, In Throughput 5.5 Bytes, Out Throughput 4.4 Bytes, Storage Size 73.61 KB.

Top navigation includes StreamNative Cloud logo, International, Organization - sndev, Instance, and a search bar. A sidebar on the right shows a profile picture and the email tim.spann@stri.

StreamNative Cloud

The screenshot displays the StreamNative Cloud dashboard. On the left, a sidebar navigation includes 'Tenants' (gke), 'Namespaces', 'Topics', 'SQL', 'Clients', 'Connector', and 'Manage'. The main area has tabs for 'OVERVIEW' (selected) and 'POLICIES'. The 'OVERVIEW' tab shows summary metrics: In Throughput (0 Bytes), Out Throughput (0 Bytes), In Rate (0.00), and Out Rate (0.00). Below this is a 'Subscriptions' section with a table:

Subscription Name	Type	Out Rate	Out Throughput	Msg Expired	Backlog
> reader-536eb1250a	Exclusive	0.00	0 Bytes	0.00	0 : 0
> reader-5ba5588002	Exclusive	0.00	0 Bytes	0.00	0 : 0
> reader-3779c9a496	Exclusive	0.00	0 Bytes	0.00	0 : 0
> reader-a84dd1d8678	Exclusive	0.00	0 Bytes	0.00	0 : 0
> reader-3cc97cac2c	Exclusive	0.00	0 Bytes	0.00	0 : 0
> flink-pulsar-acab44c0-625c-43e9-9313-5e83fbcd877	None	0.00	0 Bytes	0.00	0 : 0

Below the subscriptions is a 'Partitions' section with a table:

Partition #	Producers	Subscriptions	In Rate	Out Rate	In Throughput	Out Throughput	Storage Size
topitems3-partition-4	1	2	0.00	0.00	0 Bytes	0 Bytes	22.07 KB
topitems3-partition-3	1	2	0.00	0.00	0 Bytes	0 Bytes	0 Bytes
topitems3-partition-2	1	2	0.00	0.00	0 Bytes	0 Bytes	0 Bytes
topitems3-partition-1	1	2	0.00	0.00	0 Bytes	0 Bytes	0 Bytes
topitems3-partition-0	1	2	0.00	0.00	0 Bytes	0 Bytes	47.81 KB



StreamNative + *influxdata*[®]

streamnative.io

StreamNative Cloud

Tenant public Namespace default Topic iotjetsonjson

OVERVIEW SCHEMA MESSAGES STORAGE POLICIES

Schema Type

JSON

Schema Definition

```
1 {
2   "type": "record",
3   "name": "IoTMessage",
4   "namespace": "io.streamnative.examples.oauth2",
5   "fields": [
6     {
7       "name": "camera",
8       "type": [
9         "null",
10        "string"
11      ],
12      "default": null
13    },
14  {
```

Format Code

Properties

StreamNative Cloud

The screenshot shows the StreamNative Cloud interface. On the left, there's a sidebar with 'Catalogs' (flink-test / gke), 'Create Flink Cluster', and a tree view of 'Tables' under 'public/default'. The tables listed include data-gen-out, iotjetsonjson, jetsoniotz, jetsoniot3, kinesis-input, kinesis-output, orders, product, sensors, test1, test3, and [TENANT_NAMESPACE_...]. In the center, there's a 'Table Schema' panel with a table structure:

Field	Type
id	STRING
uuid	STRING
ir	STRING
end	STRING
lux	STRING
gputemp	STRING
cputemp	STRING
te	STRING
systemtime	STRING
hum	STRING
memory	STRING
gas	STRING
pressure	STRING
host	STRING
diskusage	STRING

Below the schema is a code editor window with the following SQL code:

```
1 CREATE TABLE jetsoniot3
2 (
3     id STRING, uuid STRING, ir STRING,
4     end STRING, lux STRING, gputemp STRING,
5     cputemp STRING, te STRING, systemtime STRING, hum STRING,
6     memory STRING,
7     gas STRING, pressure STRING, host STRING, diskusage STRING, ipaddress STRING, macaddress STRING
8     @cpuinfoof STRING, host_name STRING, camera STRING, filename STRING,
9     @volume STRING, cpu STRING, cputempsof STRING, imageinput STRING,
10    @networktime STRING, tops STRING, top_ipct STRING,
11    publishTime TIMESTAMP) METADATA;
```

At the bottom of the code editor, there are buttons for 'Run' and 'Stop'.

StreamNative Flink SQL

The screenshot shows the StreamNative Cloud interface for a GKE instance. The left sidebar includes sections for Tenants, Namespaces, Topics, SQL (selected), Clients, Connector, and Manage. The main area displays a Catalogs view with a dropdown set to 'flink-test / gke'. A 'Create Flink Cluster' button is present. Below this, under 'public/default', there are tables: data-gen-out, iotjetsonson, and jetsoniotz. The 'jetsoniotz' table is expanded, showing columns: camera (STRING), cpu (DOUBLE), cputemp (STRING), cputempf (STRING), diskusage (STRING), filename (STRING), gpumemp (STRING), gpumempf (STRING), host (STRING), host_name (STRING), imageinput (STRING), and ipaddress (STRING). The 'SQL' tab is selected, showing an 'Execution Type' dropdown set to 'streaming' with 'Run' and 'Stop' buttons. A note indicates 'Currently, Flink SQL is a preview feature.' Below the execution type, a query is displayed:

```
1 select uuid, camera, ipaddress, networktime, topipct, top,
2 cputemp, gpumemp, gpumempf, runtime, host, filename, imageinput,
3 host_name, macaddress, te, systemtime, cpu, diskusage, memory
4 from jetsoniotz /*+ OPTIONS('scan.startup.mode'='earliest') */
5 where top3 in ('monitor', 'crane', 'modem', 'envelope', 'person')
```

The 'Result' section shows a table with the following data:

uuid	camera	ipaddress	networktime	topipct	top
xav_uuid_video0_wab_20210927183538	/dev/video0	192.168.1.226	24.956928253173828	67.041015625	mc
xav_uuid_video0_ppz_20210927183613	/dev/video0	192.168.1.226	24.93337631225586	52.490234375	mc
xav_uuid_video0_idn_20210927183647	/dev/video0	192.168.1.226	24.93324851989746	45.605d6875	mc
xav_uuid_video0_gnv_20210927183721	/dev/video0	192.168.1.226	24.91584014892578	51.85546875	mc
xav_uuid_video0_rhh_20210927183754	/dev/video0	192.168.1.226	24.953855514526367	50.48828125	mc
xav_uuid_video0_ztp_20210927183826	/dev/video0	192.168.1.226	25.106399536132812	48.0712890625	mc

streamnative.io

StreamNative Flink SQL

The screenshot shows the StreamNative Cloud interface for a GKE cluster. The left sidebar includes sections for Tenants, Namespaces, Topics, SQL (with a preview tab), Clients, Connector, and Manage. The main area displays a Catalogs section with a dropdown set to 'link-test / gke' and a 'Create Flink Cluster' button. Below this is a 'Tables' section under 'public/default' containing 14 entries: camera, cpu, cputemp, cpusmpf, diskusage, filename, gputemp, gputempf, host, host_name, imageinput, ipaddress, macaddress, memory, networktime, and runtime. An 'Execution Type' dropdown is set to 'streaming', with 'Run' and 'Stop' buttons. A SQL query is entered in the text area:

```
1 select top1pc, top1.gpuTemp,cpuTemp, runtime, systemTime, cpu, diskUsage, memory
2 from jetsoniot2 /*+ OPTIONS('scan.startup.mode'='earliest') */
3 where CAST(cpuTempf as double) > 75
```

The 'Result' section shows a table with 194 rows of data, each with columns: runtime, systemtime, cpu, diskusage, and memory. The data is as follows:

runtime	systemtime	cpu	diskusage	memory
4	09/21/2021 12:05:12	8.5	33077.8 MB	34.1
4	09/21/2021 12:05:18	11	33077.8 MB	34.1
4	09/21/2021 12:05:25	11.7	33077.8 MB	34
4	09/21/2021 12:07:32	12	33077.8 MB	34
4	09/21/2021 12:07:49	12.5	33077.8 MB	34
4	09/21/2021 12:08:07	13	33077.8 MB	34
4	09/21/2021 12:08:24	12.3	33077.8 MB	34.1
4	09/21/2021 12:08:41	12.5	33077.8 MB	34
4	09/21/2021 12:08:58	12.5	33077.8 MB	34
4	09/21/2021 12:09:15	18.3	33077.8 MB	34.1

Pagination controls at the bottom right indicate pages 1 through 194.

StreamNative Flink SQL

The screenshot shows the StreamNative Cloud interface with the following details:

- Top Bar:** StreamNative Cloud, International, Organization - sndev, Instance - gke.
- Left Sidebar:** Tenants, Namespaces, Topics, SQL (selected), Clients, Connector, Manage.
- Middle Panel:**
 - Catalogs:** flink-test / gke. A "Create Flink Cluster" button is present.
 - Tables:** public/default
 - data-gen-out
 - iotjsonson
 - jetsontotext
 - jetsontoreturns
 - kinesis-input
 - kinesis-output
 - order_with_time
 - Orders
 - product
 - sensors
 - test1
 - test3
 - topitems3
 - {TENANT_NAMESPACE}_...
- Execution Type:** streaming. Buttons: Run, Stop.
- SQL Editor:** A preview message says "Currently, Flink SQL is a preview feature." The query entered is:

```
1 SELECT top1, COUNT(*) AS ai_cnt FROM jetsontotext /* OPTIONS('scan.startup.mode':'earliest') */ GROUP BY top1
```
- Result:** A table showing the results of the query:

top1	ai_cnt
web site, website, internet site, site	1
rule, ruler	2
crane	1
binder, ring-binder	7
mouse, computer mouse	1
laptop, laptop computer	10
modem	2
notebook, notebook computer	28

StreamNative Flink SQL

The screenshot shows the StreamNative Cloud interface for a GKE instance. On the left, a sidebar navigation includes: Tenants, Namespaces, Topics, SQL (selected), Clients, Connector, and Manage. The main area displays a catalog named "flink-test / gke". Under "Tables", there are entries for "data-gen-out", "iotjetsonjson", "jetsoniot2", and "jetsoniotresults". The "jetsoniotresults" table is expanded, showing columns: uuid (STRING), camera (STRING), ipaddress (STRING), networktime (STRING), top1pcpt (DOUBLE), top1 (STRING), cputemp (STRING), gputemp (STRING), gputempf (STRING), cputempf (STRING), and runtime (STRING). A SQL query is being run against the "jetsoniotresults" table:

```
1 select top1,
2   min(CAST(cputempf as double)) as avgcputempf, min(gputempf) as avggputempf
3 from jetsoniot2 /*+ OPTIONS('scan.startup.mode'='earliest') */
4 group by top1
```

The "Result" section shows the output of the query:

top1	avgcputempf	avggputempf
binder,ring-binder	85	84
web site,website,internet site,site	88	87
rule,ruler	88	88
crane	88	87
monitor	80	100
notebook,notebook computer	81	81
laptop,laptop computer	82	82
mouse,computer mouse	83	83
modem	87	87
radio,wireless	87	87

Influx Data Explorer

Weather

This dashboard doesn't have any cells with defined variables. [Learn How](#)

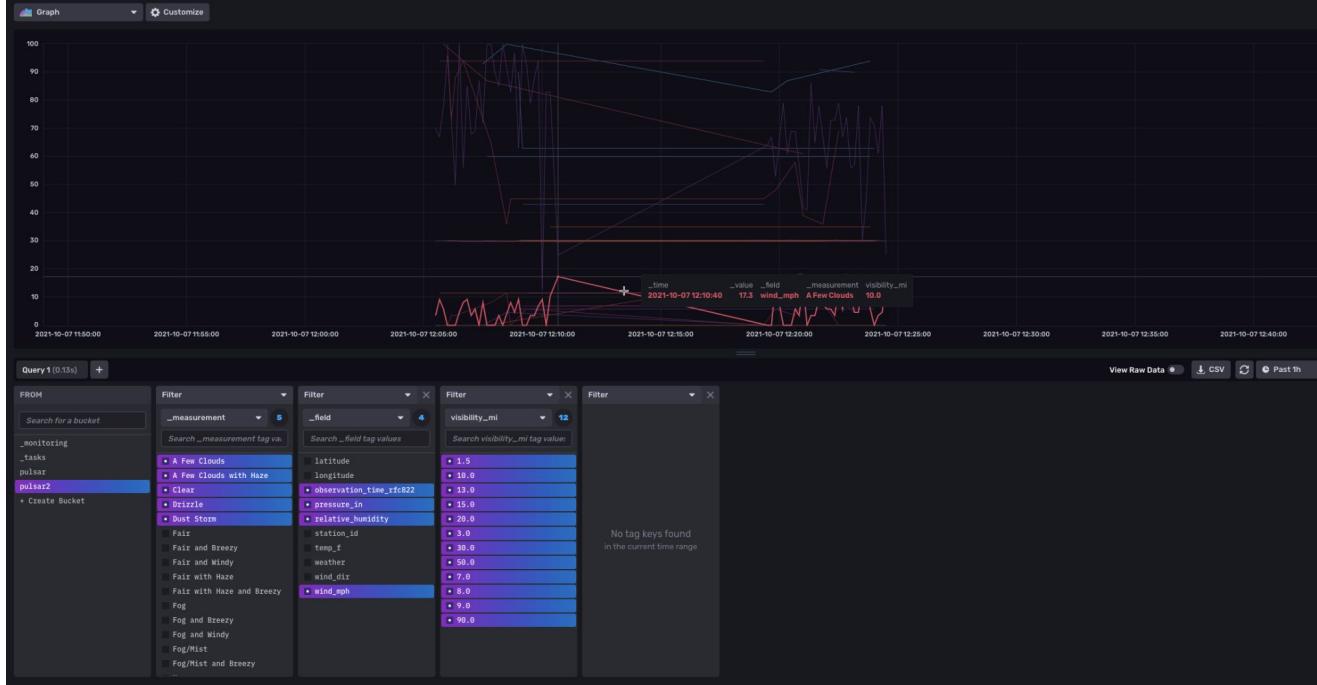
Name this Cell	_start	_stop	_time	_value	_field	_measurement	visibility_mi
Filter tables...	2021-10-07 11:23:04 EDT	2021-10-07 12:23:04 EDT	2021-10-07 12:05:30 EDT	30.00	pressure_in	Fair	7.0
_field = wind_mph .._measurement = Party Cloudy visibility_mi = 10.0	2021-10-07 11:23:04 EDT	2021-10-07 12:23:04 EDT	2021-10-07 12:05:50 EDT	30.02	pressure_in	Fair	7.0
_field = relative_humidity .._measurement = A Few Clouds visibility_mi = 30.0	2021-10-07 11:23:04 EDT	2021-10-07 12:23:04 EDT	2021-10-07 12:06:00 EDT	29.93	pressure_in	Fair	7.0
_field = pressure_in .._measurement = Fog/Mist visibility_mi = 7.0	2021-10-07 11:23:04 EDT	2021-10-07 12:23:04 EDT	2021-10-07 12:06:40 EDT	30.05	pressure_in	Fair	7.0
_field = pressure_in .._measurement = Mostly Cloudy visibility_mi = 10.0	2021-10-07 11:23:04 EDT	2021-10-07 12:23:04 EDT	2021-10-07 12:07:00 EDT	30.07	pressure_in	Fair	7.0
_field = temp_f .._measurement = A Few Clouds visibility_mi = 7.0	2021-10-07 11:23:04 EDT	2021-10-07 12:23:04 EDT	2021-10-07 12:07:50 EDT	29.96	pressure_in	Fair	7.0
_field = relative_humidity .._measurement = Fog/Mist visibility_mi = 12.5	2021-10-07 11:23:04 EDT	2021-10-07 12:23:04 EDT	2021-10-07 12:08:00 EDT	29.98	pressure_in	Fair	7.0
_field = pressure_in .._measurement = Thunderstorm in Vicinity Rain visibility_mi = 0.0	2021-10-07 11:23:04 EDT	2021-10-07 12:23:04 EDT	2021-10-07 12:08:40 EDT	30.06	pressure_in	Fair	7.0
_field = wind_mph .._measurement = Fog visibility_mi = 0.25	2021-10-07 11:23:04 EDT	2021-10-07 12:23:04 EDT	2021-10-07 12:09:30 EDT	29.88	pressure_in	Fair	7.0
_field = relative_humidity .._measurement = Fog/Mist visibility_mi = 1.0	2021-10-07 11:23:04 EDT	2021-10-07 12:23:04 EDT	2021-10-07 12:09:50 EDT	29.98	pressure_in	Fair	7.0
_field = wind_mph .._measurement = Thunderstorm in Vicinity Fog/Mist visibility_mi = 0.0	2021-10-07 11:23:04 EDT	2021-10-07 12:23:04 EDT	2021-10-07 12:10:00 EDT	30.04	pressure_in	Fair	7.0
_field = relative_humidity .._measurement = Mostly Cloudy and Breezy visibility_mi = 0.0	2021-10-07 11:23:04 EDT	2021-10-07 12:23:04 EDT	2021-10-07 12:10:20 EDT	30.03	pressure_in	Fair	7.0
_field = wind_mph .._measurement = Thunderstorm in Vicinity Rain visibility_mi = 0.0	2021-10-07 11:23:04 EDT	2021-10-07 12:23:04 EDT	2021-10-07 12:10:30 EDT	29.96	pressure_in	Fair	7.0
_field = temp_f .._measurement = Thunderstorm Heavy Rain and Breezy visibility_mi = 0.0	2021-10-07 11:23:04 EDT	2021-10-07 12:23:04 EDT	2021-10-07 12:10:40 EDT	30.07	pressure_in	Fair	7.0
_field = wind_mph .._measurement = Thunderstorm Light Rain Fog/Mist visibility_mi = 0.0	2021-10-07 11:23:04 EDT	2021-10-07 12:23:04 EDT	2021-10-07 12:10:50 EDT	29.95	pressure_in	Fair	7.0
_field = wind_mph .._measurement = Clear visibility_mi = 20.0	2021-10-07 11:23:04 EDT	2021-10-07 12:23:04 EDT	2021-10-07 12:10:50 EDT	29.97	pressure_in	Fair	7.0
_field = temp_f .._measurement = Fog/Mist visibility_mi = 1.0	2021-10-07 11:23:04 EDT	2021-10-07 12:23:04 EDT	2021-10-07 12:10:50 EDT	29.89	pressure_in	Fair	7.0
_field = pressure_in .._measurement = Heavy Rain visibility_mi = 1.5	2021-10-07 11:23:04 EDT	2021-10-07 12:23:04 EDT	2021-10-07 12:10:50 EDT	30.03	pressure_in	Fair	7.0
_field = relative_humidity .._measurement = Mostly Cloudy visibility_mi = 15.0	2021-10-07 11:23:04 EDT	2021-10-07 12:23:04 EDT	2021-10-07 12:10:50 EDT	29.99	pressure_in	Fair	7.0
_field = wind_mph .._measurement = Party Cloudy with Haze visibility_mi = 6.0	2021-10-07 11:23:04 EDT	2021-10-07 12:23:04 EDT	2021-10-07 12:10:50 EDT	30.09	pressure_in	Fair	7.0
_field = relative_humidity .._measurement = Light Drizzle visibility_mi = 10.0	2021-10-07 11:23:04 EDT	2021-10-07 12:23:04 EDT	2021-10-07 12:10:50 EDT	30.02	pressure_in	Fair	7.0
_field = pressure_in .._measurement = Mostly Cloudy visibility_mi = 8.0	2021-10-07 11:23:04 EDT	2021-10-07 12:23:04 EDT	2021-10-07 12:10:50 EDT	30.05	pressure_in	Fair	7.0
_field = pressure_in .._measurement = Thunderstorm Light Rain Fog/Mist visibility_mi = 0.0	2021-10-07 11:23:04 EDT	2021-10-07 12:23:04 EDT	2021-10-07 12:10:50 EDT	30.06	pressure_in	Fair	7.0
_field = relative_humidity .._measurement = Overcast with Haze and Windy visibility_mi = 0.0	2021-10-07 11:23:04 EDT	2021-10-07 12:23:04 EDT	2021-10-07 12:10:50 EDT	30.29	pressure_in	Fair	7.0
_field = wind_mph .._measurement = Fair and Windy visibility_mi = 1.75	2021-10-07 11:23:04 EDT	2021-10-07 12:23:04 EDT	2021-10-07 12:20:10 EDT	30	pressure_in	Fair	7.0
_field = wind_mph .._measurement = Fair and Windy visibility_mi = 10.0	2021-10-07 11:23:04 EDT	2021-10-07 12:23:04 EDT	2021-10-07 12:20:20 EDT	30.11	pressure_in	Fair	7.0
_field = pressure_in .._measurement = Mostly Cloudy with Haze visibility_mi = 1.75	2021-10-07 11:23:04 EDT	2021-10-07 12:23:04 EDT	2021-10-07 12:20:30 EDT	30.07	pressure_in	Fair	7.0



StreamNative + influxdata[®]

streamnative.io

Influx Data Explorer



Web Socket Visualization



User: Data Engineer 7

Why is Pulsar and Scylla together such a great idea?

Question:

Contact Info: Data Engineer at Fortune 500 Company

Send to Pulsar



Show 10 entries

Sentiment	Publish Time	Message	User	Contact Info
Positive	2021-12-14T14:36:35.868-05:00	Why is Pulsar and Scylla together such a great idea?	Data Engineer 7	Data Engineer at Fortune 500 Company
Positive	2021-12-14T14:36:34.041-05:00	Why is Pulsar and Scylla together such a great idea?	Data Engineer 7	Data Engineer at Fortune 500 Company
Positive	2021-12-14T14:16:55.776-05:00	We all know that Apache Pulsar along with Apache Flink, Apache Spark, Apache NiFi and Scylla DB are the best options for modern streaming data applications.	Tim Spann	Tim Spann - Developer Advocate @ StreamNative
Positive	2021-12-14T14:16:34.858-05:00	We all know that Apache Pulsar along with Apache Flink, Apache Spark, Apache NiFi and Scylla DB are the best options for modern streaming data applications.	Tim Spann	Tim Spann, Developer Advocate @ StreamNative
Positive	2021-12-14T14:16:32.717-05:00	We all know that Apache Pulsar along with Apache Flink, Apache Spark, Apache NiFi and Scylla DB are the best options for modern streaming data applications.	Tim Spann	Tim Spann, Developer Advocate @ StreamNative
Neutral	2021-12-14T09:53:12.787-05:00	this is a test	Porto	Porto
Positive	2021-12-14T09:51:38.726-05:00	pulsar and flink with spark are amazing	Tim Spann	Tim @ Data Science Camp
Positive	2021-12-14T09:50:50.835-05:00	pulsar and flink with spark are amazing	Tim Spann	Tim @ Data Science Camp
Neutral	2021-12-14T09:43:27.878-05:00	test	Tim Spann	test
Positive	2021-12-14T09:16:29.737-05:00	Why is Apache Pulsar so amazing?	Tim Spann	Tim @ StreamNative

Sentiment Publish Time Message User Contact Info

Showing 1 to 10 of 23 entries

Previous 1 2 3 Next



streamnative.io



- <https://github.com/tspannhw/FLiP-InfluxDB>
- <https://pulsar.apache.org/docs/en/io-influxdb-sink/>
- <https://pulsar.apache.org/docs/en/io-overview/>
- <https://github.com/tspannhw/Flip-solr>
- <https://github.com/tspannhw/Flip-transit>
- <https://github.com/tspannhw/Flip-iot>
- <https://www.datainmotion.dev/2020/10/flank-streaming-edgeai-on-new-nvidia.html>
- <https://github.com/tspannhw/minifi-jetson-nano>
- <https://github.com/tspannhw/SpeakerProfile/>
- <https://hub.streamnative.io/connectors/influxdb-sink/2.5.1/>
- <https://github.com/tspannhw/minifi-xaviernx/>

Resources



StreamNative



influx^{data}®

**Now Available
On-Demand
Pulsar Training**

[Academy.StreamNative](#)

**Live 3-day
Developers Training**

[Save Your Spot!](#)

**Free ebook
Pulsar in Action**

[Read it now](#)

Let's Keep in Touch!



Timothy Spann
Developer Advocate



@[PaasDev](#)



<https://www.linkedin.com/in/timothyspann>



<https://github.com/tspannhw>

Questions?
