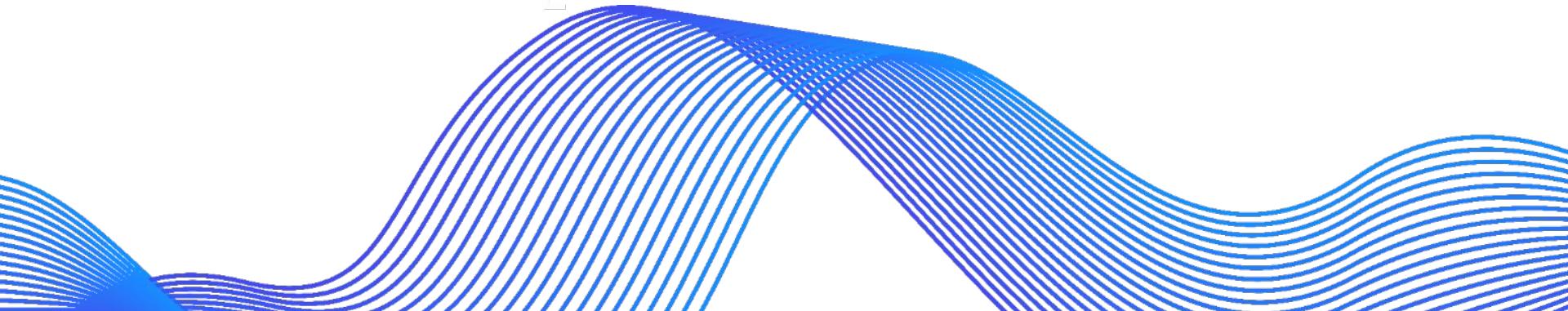




Stream Native

Fast Streaming into Clickhouse
With Apache Pulsar



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, Big Data, Cloud, MXNet, IoT, Python 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

<https://streamnative.io/pulsar-python/>



Hewlett Packard
Enterprise

FLiP Stack Weekly



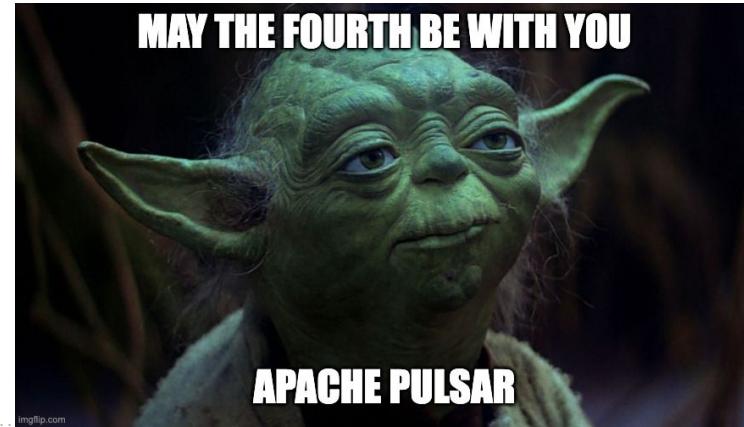
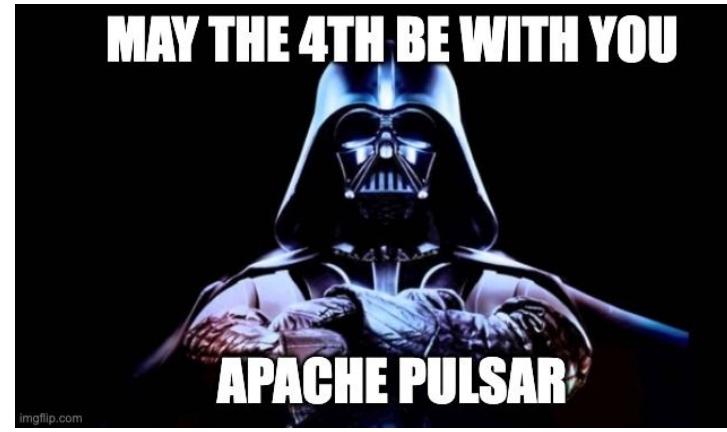
<https://bit.ly/32dAJft>



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

Agenda

- What is Apache Pulsar? 2.10!
- Code / Demonstration.





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



The right API for async



Ideal for high-scale,
mission critical
microservices



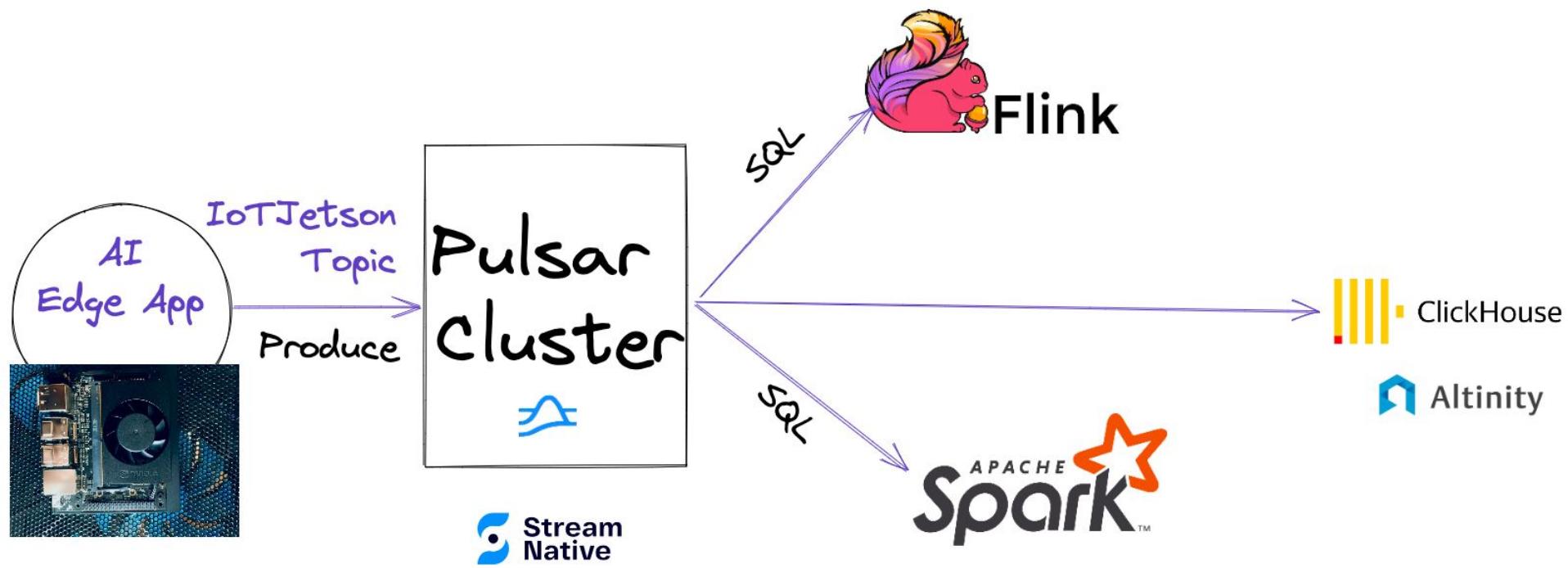
Easy to use, with a
simple pub/sub API



Power and flexibility,
w/ support for
simultaneous streaming
and messaging use cases



Designed for teams, with
built in multi-tenancy



What are the Benefits of Pulsar?



Multi-Tenancy

Scalability

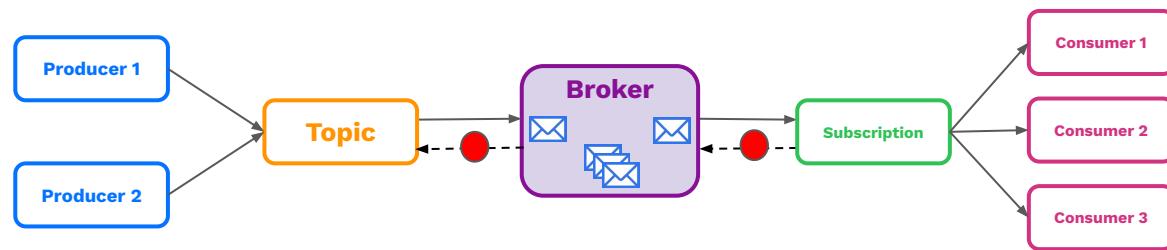
Geo-Replication

Unified Messaging
Model

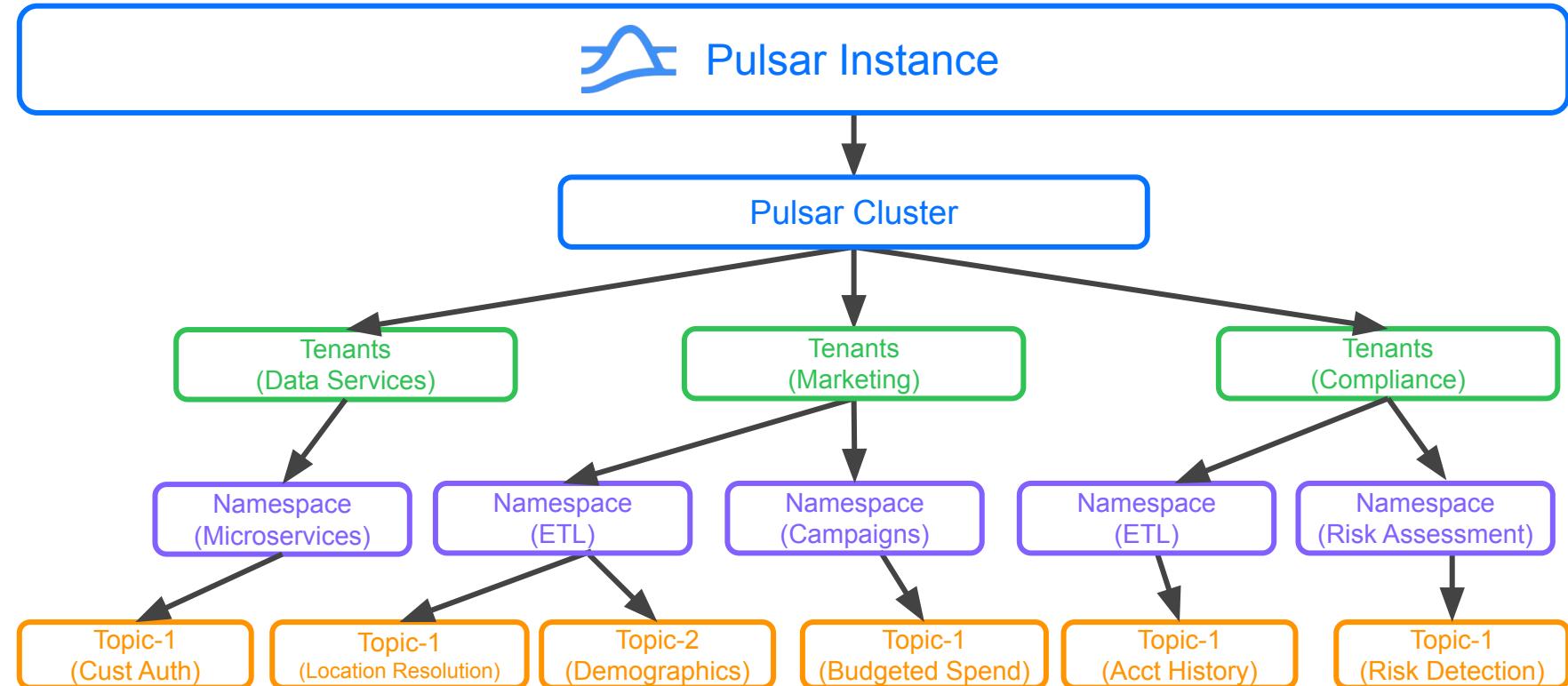
Data Durability

Pulsar's Publish-Subscribe model

- Producers send messages.
- Topics are an ordered, named channel that producers use to transmit messages to subscribed consumers.
- Messages belong to a topic and contain an arbitrary payload.
- Brokers handle connections and routes messages between producers / consumers.
- Subscriptions are named configuration rules that determine how messages are delivered to consumers.
- Consumers receive messages.



Topics



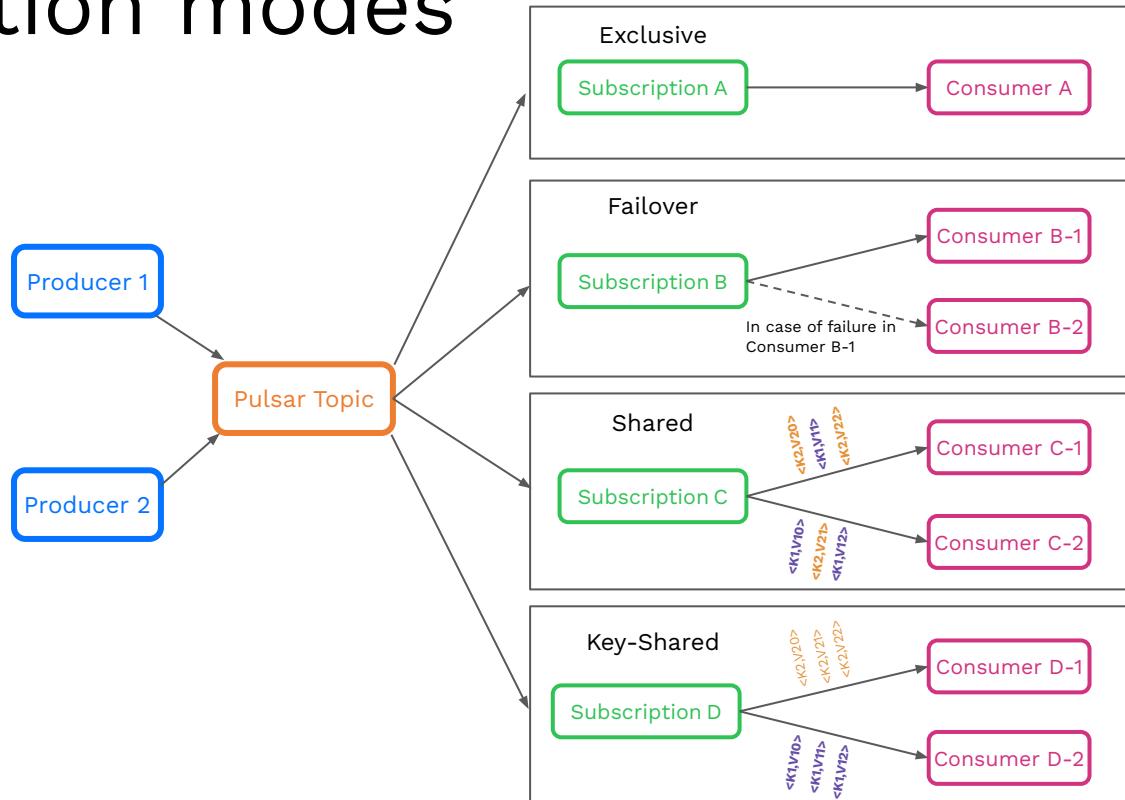
Pulsar subscription modes

Different subscription modes have different semantics:

Exclusive/Failover - guaranteed order, single active consumer

Shared - multiple active consumers, no order

Key_Shared - multiple active consumers, order for given key



Pulsar Terminology

Producer is a process that publishes messages to a topic.

Consumer is a process that establishes a subscription to a topic and processes messages published to that topic.

Subscription: A subscription is a named configuration rule that determines how messages are delivered to consumers.

Brokers handle the connections and routes messages.

Topics are named channels for transmitting messages from producers to consumers.

Messages belong to a topic and contain an arbitrary payload.

BookKeeper log storage system that Pulsar uses for durable storage of all messages.

Bookie Stores messages and cursors. Messages are grouped in segments/ledgers.

ZooKeeper Stores metadata for both Pulsar and BookKeeper, also performs service discovery.

Instance is a group of clusters that act together as a single unit.

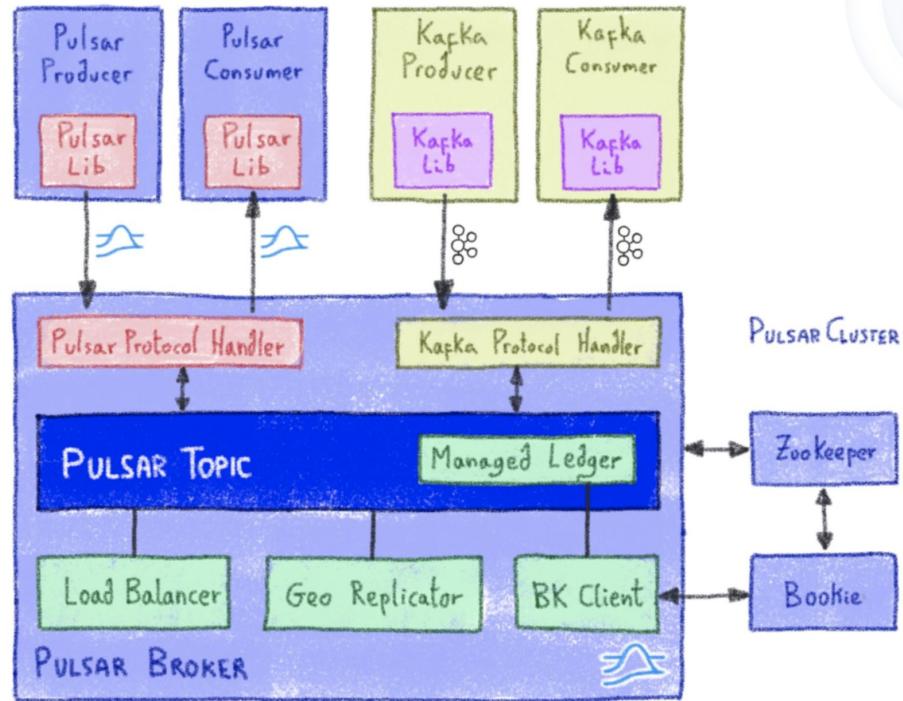
Cluster is a set of Pulsar brokers, ZooKeeper quorum, and an ensemble of BookKeeper bookies.

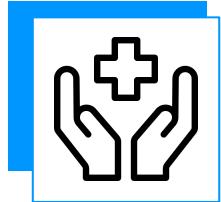
Tenants are the administrative unit for allocating capacity and enforcing an authentication/authorization scheme.

Namespaces are a grouping mechanism for related topics.

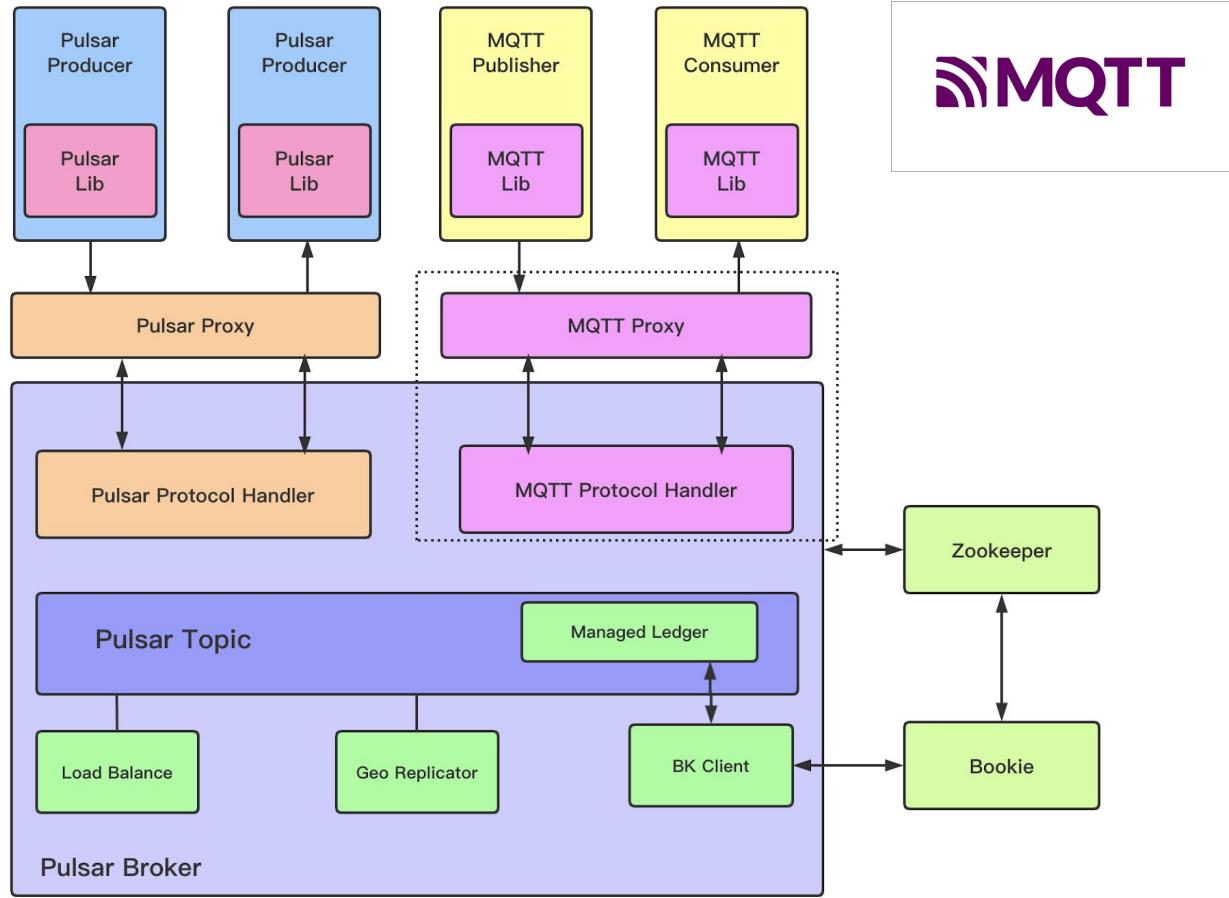


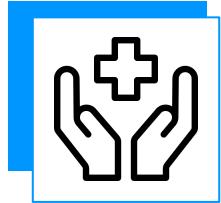
Kafka On Pulsar (KoP)



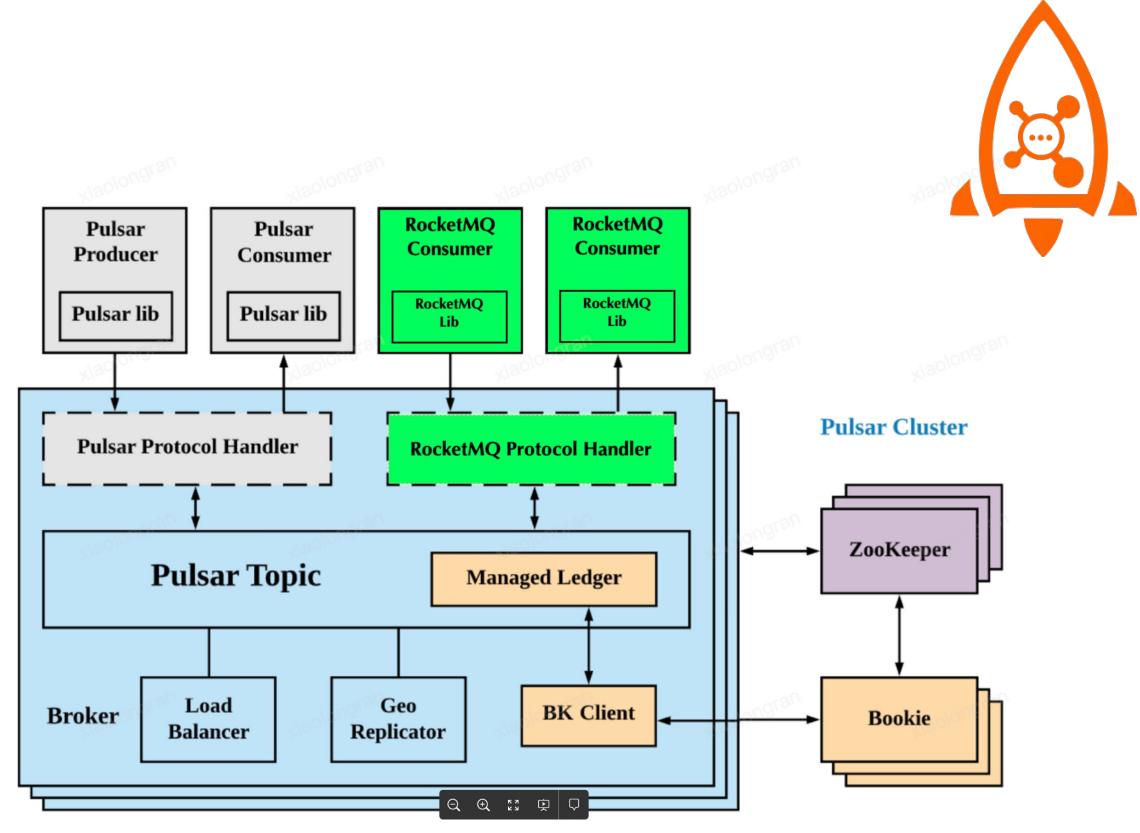


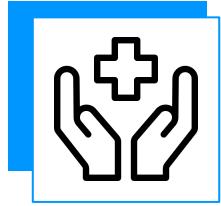
MQTT On Pulsar (MoP)



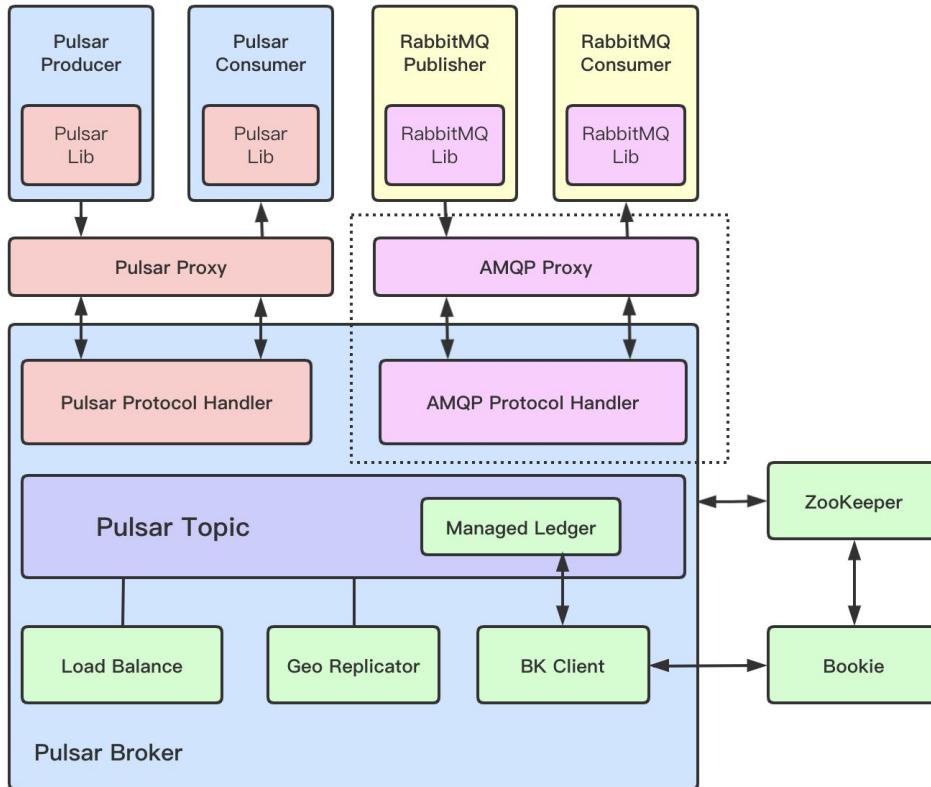


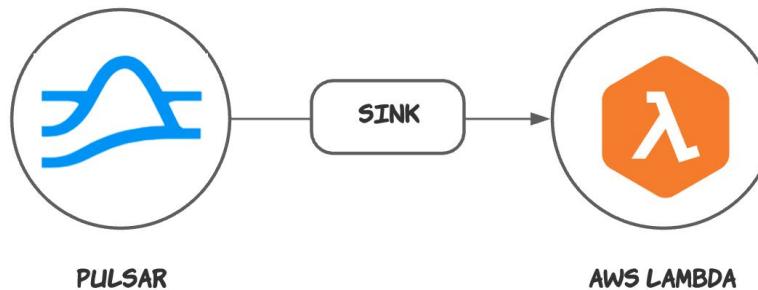
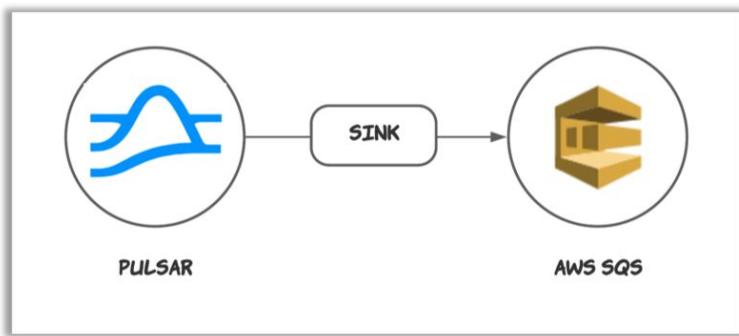
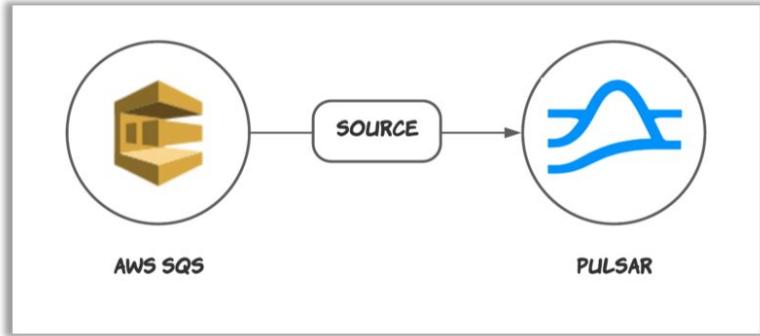
RocketMQ On Pulsar (RoP)





AMQP On Pulsar (AoP)





Moving Data Out of Pulsar to Clickhouse

IO/Connectors are a simple way to integrate with external systems and move data in and out of Pulsar. <https://pulsar.apache.org/docs/en/io-jdbc-sink/>

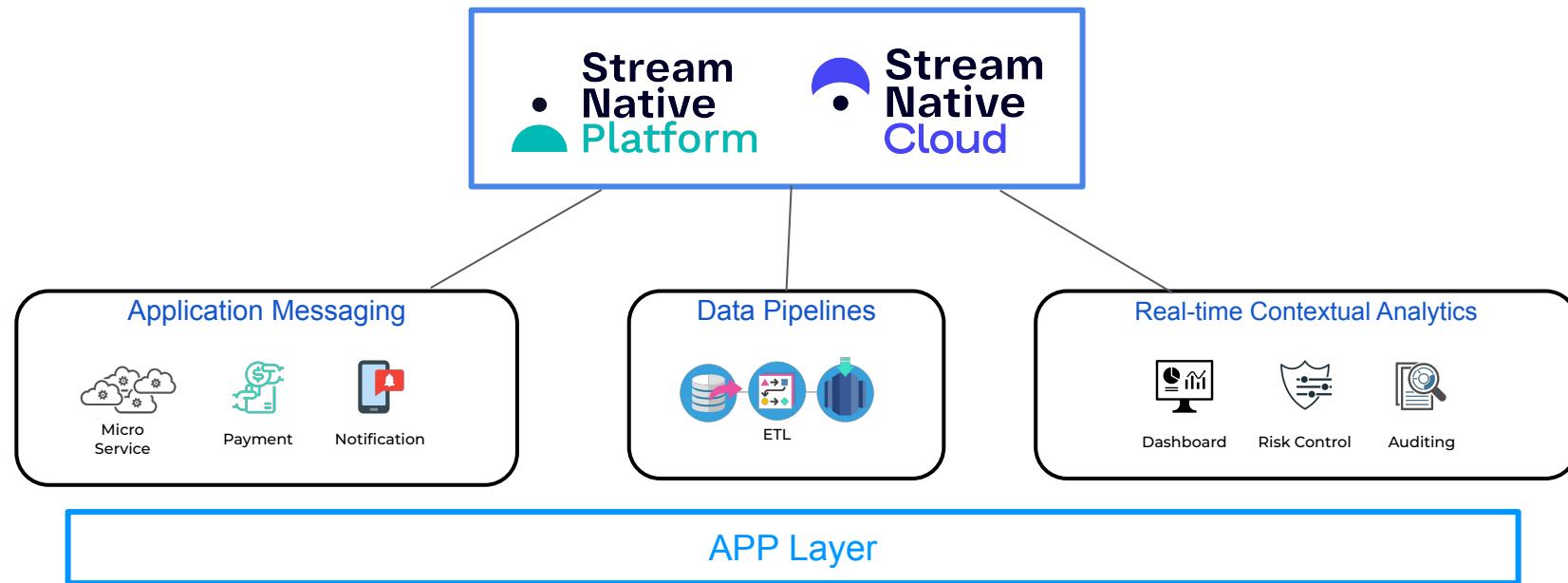
- Built on top of Pulsar Functions
- Built-in connectors - hub.streamnative.io



StreamNative Cloud



How Companies Use StreamNative today



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



Demo



<https://github.com/tspannhw/FLiP-Stream2Clickhouse/>

Clickhouse is fast and easy to insert and query streaming data.

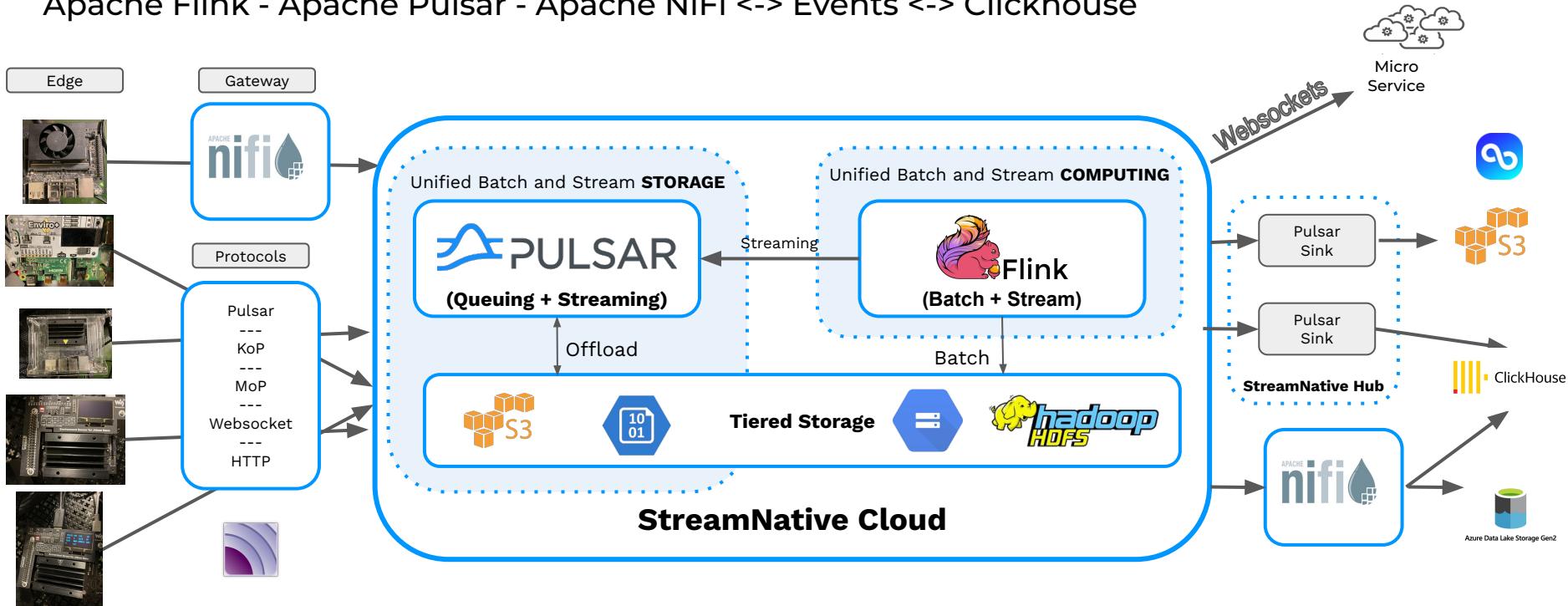
Key Takeaways

Apache Pulsar is a great option for event-based streaming & storage.

Pulsar IO Sinks are easy.

End-to-End Streaming FLiP(N) Apps

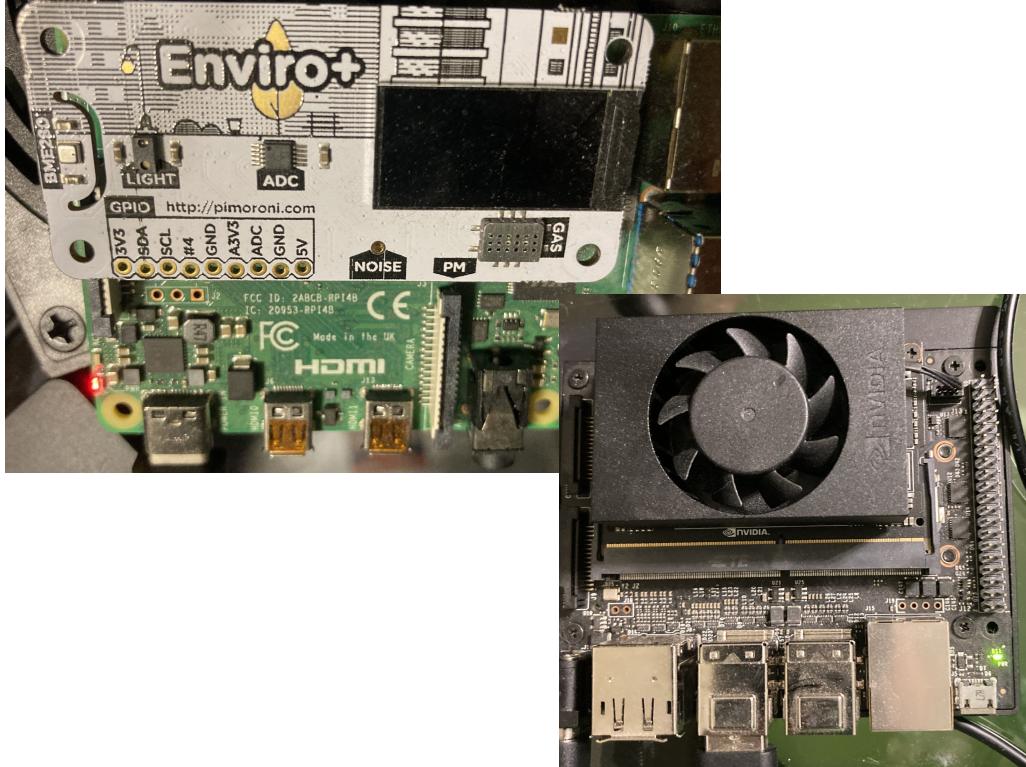
Apache Flink - Apache Pulsar - Apache NiFi <-> Events <-> Clickhouse



IoT Data

IoT Ingestion: High-volume streaming sources, sensors, multiple message formats, diverse protocols and multi-vendor devices creates data ingestion challenges.

Other Sources: Transit data, news, twitter, status feeds, REST data, stock data and more.



Build Cluster in Altinity Cloud

Altinity.Cloud

BACK MANAGE CLUSTERS

streamnative-meetup overview

Type	Kubernetes	Cloud Provider	AWS
ClickHouse Clusters	1	Kubernetes Namespace	streamnative-meetup
ClickHouse Nodes	1	Kubernetes Nodes	4
Zookeeper Clusters	1	Availability Zones	us-east-1a, us-east-1b, us-east-1c
Zookeeper Nodes	1	CH Operator	Version 0.18.3
Node Types	11	Monitoring	View in Grafana

memory

cpu

Node	Type	Zone	memory	memory usage	cpu	cpu usage
ip-10-129-12-9.ec2.internal	t3.large	us-east-1c	7.7 GB	105%	2	125%
ip-10-129-8-189.ec2.internal	t3.large	us-east-1a	7.7 GB	27%	2	66%
ip-10-129-9-32.ec2.internal	m5.large	us-east-1a	7.6 GB	93%	2	90%
ip-10-129-9-51.ec2.internal	t3.large	us-east-1a	7.7 GB	22%	2	56%

Table Details

Table Description Table Schema Sample Rows Column Compression

Copy to Clipboard

```
CREATE TABLE default.iotjetsonjson_local
(
    `uuid` String,
    `camera` String,
    `ipaddress` String,
    `networktime` String,
    `toplct` String,
    `top` String,
    `opitemp` String,
    `gputemp` String,
    `gputempf` String,
    `cpumemp` String,
    `runtime` String,
    `host` String,
    `filename` String,
    `host_name` String
)
```

DONE

<https://docs.altinity.com>

Streaming Events into Altinity Cloud

```
CREATE TABLE iotjetsonjson_local
(
    uuid String, camera String,      ipaddress String,    networktime String,          top1pct String,
    top1 String, cputemp String,     gputemp String,      gputempf String,
    cputempf String,    runtime String,
    host String, filename String,   host_name String,    macaddress String,
    te String,   systemtime String,  cpu String,        diskusage String,
    memory String,    imageinput String
)
ENGINE = MergeTree()
PARTITION BY uuid
ORDER BY (uuid);

CREATE TABLE iotjetsonjson ON CLUSTER '{cluster}' AS iotjetsonjson_local
ENGINE = Distributed('{cluster}', default, iotjetsonjson_local, rand());
```

<https://docs.altinity.com/altinitycloud/quickstartguide/connectclient/>

pulsar-io-jdbc-clickhouse-2.X.nar

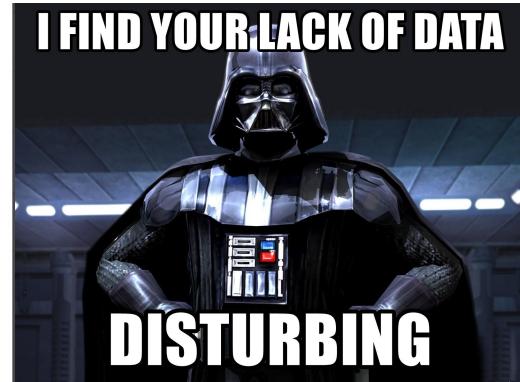
Streaming Events into Altinity Cloud

Table Details

Table Description	Table Schema	Sample Rows	Column Compression
uuid xav_uuid_video0_dlm_20220401140320	camera /dev/video0	ipaddress 192.168.1.216	networktime 24.937440872192383 top1pc 22.045
uuid xav_uuid_video0_wxz_20220401140251	camera /dev/video0	ipaddress 192.168.1.216	networktime 35.330142974853516 top1pc 12.078
uuid xav_uuid_video0_wpo_20220401140233	camera /dev/video0	ipaddress 192.168.1.216	networktime 86.29341125488281 top1pct 9.55810
uuid xav_uuid_video0_rdk_20220401135840	camera /dev/video0	ipaddress 192.168.1.216	networktime 179.46112060546875 top1pc 35.522
uuid xav_uuid_video0_mye_20220401140306	camera /dev/video0	ipaddress 192.168.1.216	networktime 24.9671688079834 top1pct 32.54394

X

DONE



Clickhouse Altinity Cloud Table

Table Details

Table Description Table Schema Sample Rows Column Compression

database	default
name	iotjetjson_local
uuid	00000000-0000-0000-0000-000000000000
engine	MergeTree
is_temporary	0
data_paths	/var/lib/clickhouse/data/default/iotjetjson_local/
	/var/lib/clickhouse/metadata/default/iotjetjson_local.sql
metadata_modification_time	2022-04-01 13:44:26
dependencies_database	
dependencies_table	
partition_key	uuid
sorting_key	uuid
primary_key	uuid
sampling_key	



StreamNati

Monitor The Cluster in Altinity Cloud



Monitor The Cluster in Altinity Cloud

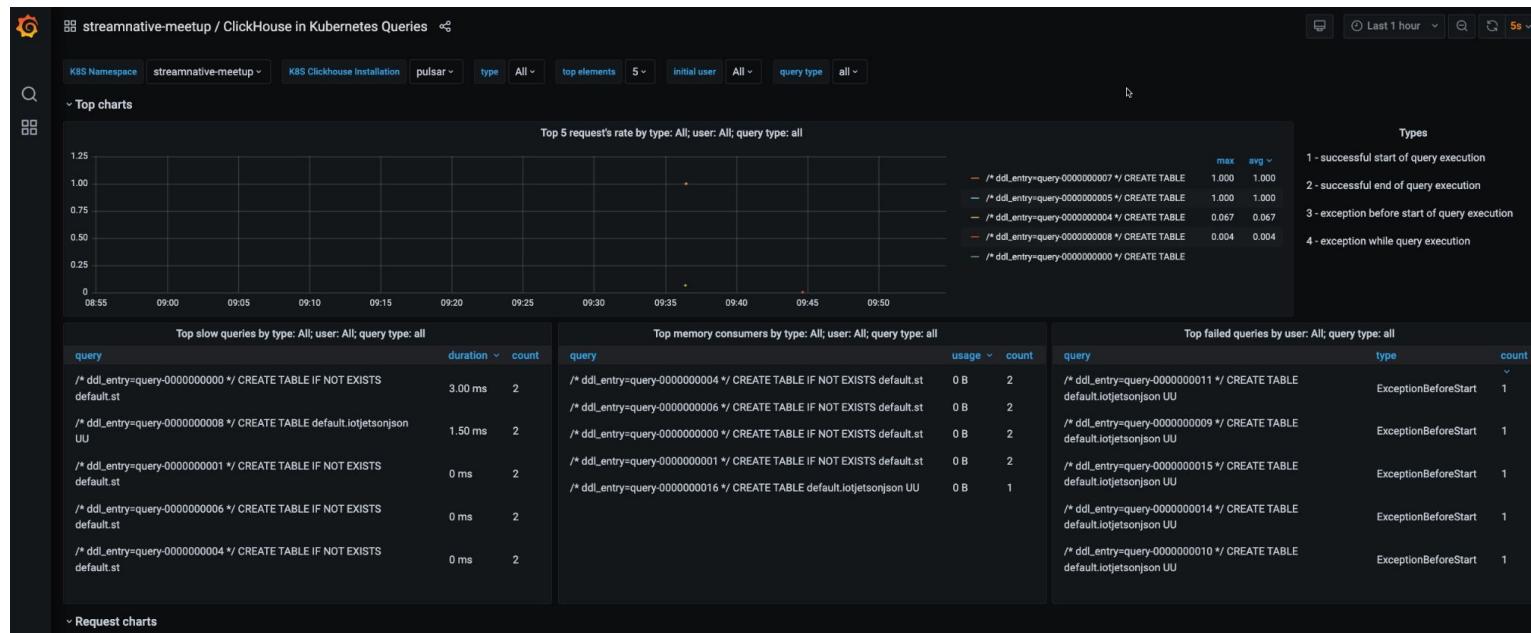
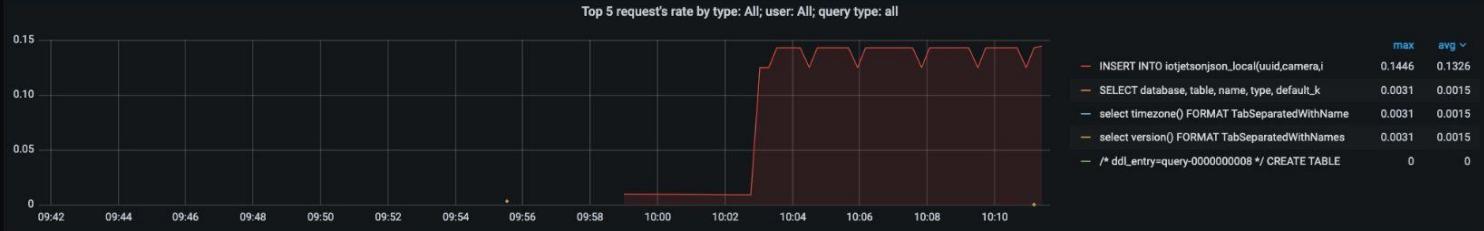


Table Details

[Table Description](#)[Table Schema](#)[Sample Rows](#)[Column Compression](#)[Copy to Clipboard](#)

```
CREATE TABLE default.iotjetsonjson_local
(
    `uuid` String,
    `camera` String,
    `ipaddress` String,
    `networktime` String,
    `top1pct` String,
    `top1` String,
    `cputemp` String,
    `gputemp` String,
    `gputempf` String,
    `cputempf` String,
    `runtime` String,
    `host` String,
    `filename` String,
    `host_name` String
)
```

[DONE](#)



Types

- 1 - successful start of query execution
- 2 - successful end of query execution
- 3 - exception before start of query execution
- 4 - exception while query execution

Top slow queries by type: All; user: All; query type: all

query	duration	count
/* ddl_entry=query-0000000008 */ CREATE TABLE default.iotjetjson UU	1.50 ms	2
INSERT INTO iotjetjson_local(uuid,camera,ipaddress,netw	1.07 ms	76
SELECT database, table, name, type, default_kind as default_type, defa	0.50 ms	2
select database, name, engine from system.tables where 1 = 1 and name	0.50 ms	2
/* ddl_entry=query-0000000012 */ CREATE TABLE default.iotjetjson UU	0 ms	1

Top memory consumers by type: All; user: All; query type: all

query	usage	count
INSERT INTO iotjetjson_local(uuid,camera,ipaddress,netw	2.09 MiB	76
SELECT database, table, name, type, default_kind as default_type, defa	796.00 B	2
select database, name, engine from system.tables where 1 = 1 and name	348.00 B	2
select timezone() FORMAT TabSeparatedWithNamesAndTypes;	316.00 B	2
select version() FORMAT TabSeparatedWithNamesAndTypes;	316.00 B	2

Top failed queries by user: All; query type: all

query	type	count
/* ddl_entry=query-0000000011 */ CREATE TABLE default.iotjetjson UU	ExceptionBeforeStart	1
/* ddl_entry=query-0000000009 */ CREATE TABLE default.iotjetjson UU	ExceptionBeforeStart	1
/* ddl_entry=query-0000000015 */ CREATE TABLE default.iotjetjson UU	ExceptionBeforeStart	1
/* ddl_entry=query-0000000014 */ CREATE TABLE default.iotjetjson UU	ExceptionBeforeStart	1
/* ddl_entry=query-0000000010 */ CREATE TABLE default.iotjetjson UU	ExceptionBeforeStart	1

Request charts

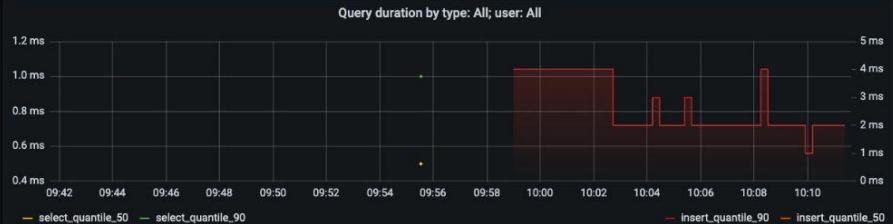
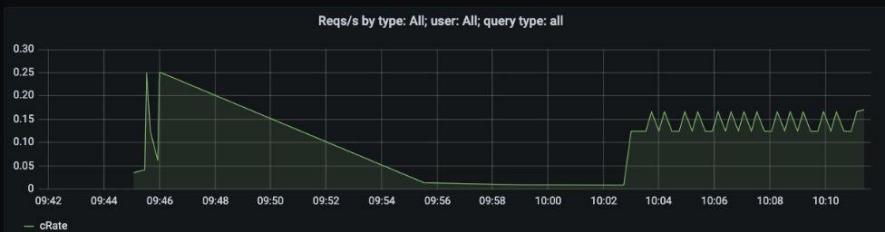


Table Details

X

Table Description	Table Schema	Sample Rows	Column Compression	
uuid xav_uuid_video0_dlm_20220401140320	camera /dev/video0	ipaddress 192.168.1.216	networktime 24.937440872192383	top1pc 22.045
uuid xav_uuid_video0_wxz_20220401140251	camera /dev/video0	ipaddress 192.168.1.216	networktime 35.330142974853516	top1pc 12.078
uuid xav_uuid_video0_wpo_20220401140233	camera /dev/video0	ipaddress 192.168.1.216	networktime 86.29341125488281	top1pc 9.55810
uuid xav_uuid_video0_rdk_20220401135840	camera /dev/video0	ipaddress 192.168.1.216	networktime 179.46112060546875	top1pc 35.522
uuid xav_uuid_video0_mye_20220401140306	camera /dev/video0	ipaddress 192.168.1.216	networktime 24.9671688079834	top1pc 32.54394



DONE



K8S Namespace

streamnative-meetup

K8S Clickhouse Installation

pulsar

Server

All

prometheus alerts



Uptime

Current

Failed Pods

0

ReadOnly replicas

0

Version

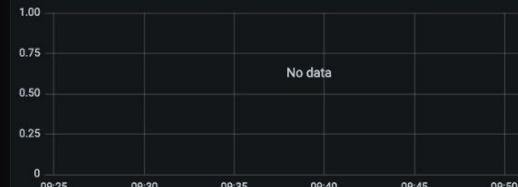
Current

Metric

chi-pulsar-pulsar-0-0.streamnative-meetup.svc.cluster.local

21008013

DNS and Distributed Connection Errors



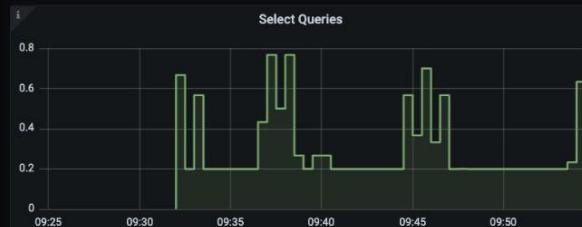
Replication and ZooKeeper Exceptions



Delayed/Rejected/Pending Inserts



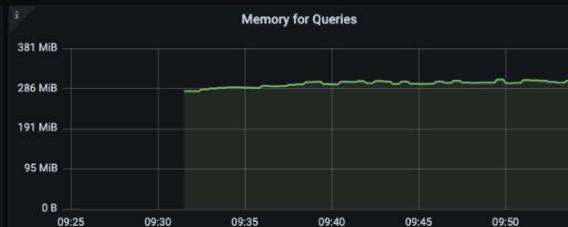
Select Queries



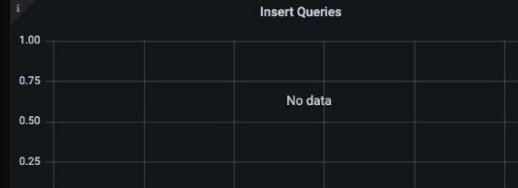
Read Bytes



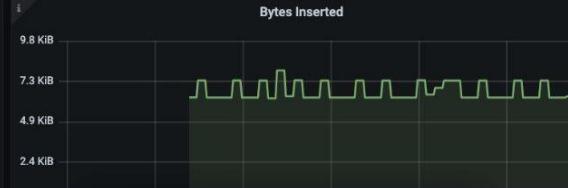
Memory for Queries



Insert Queries



Bytes Inserted



Rows Inserted





<<

BACK

MANAGE CLUSTERS

Accounts

Environments

Clusters

Activate Subscription

streamnative-meetup overview

Type Kubernetes

ClickHouse Clusters 1

Cloud Provider

AWS

Kubernetes Namespace

streamnative-meetup

ClickHouse Nodes 1

Kubernetes Nodes

4

Zookeeper Clusters 1

Availability Zones

us-east-1a, us-east-1b, us-east-1c

Zookeeper Nodes 1

CH Operator

Version 0.18.3

Node Types 11

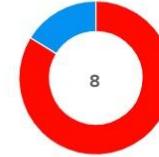
Monitoring

View in Grafana

memory

● Used
● Free

cpu

● Used
● Free

Node

Type

Zone

memory

memory usage

cpu

cpu usage

ip-10-129-12-9.ec2.internal

t3.large

us-east-1c

7.7 GB

105%

2

125%

ip-10-129-8-189.ec2.internal

t3.large

us-east-1a

7.7 GB

27%

2

66%

ip-10-129-9-32.ec2.internal

m5.large

us-east-1a

7.6 GB

93%

2

90%

ip-10-129-9-51.ec2.internal

t3.large

us-east-1a

7.7 GB

22%

2

56%

Table Details

X

Table Description Table Schema Sample Rows Column Compression

database	default
name	iotjetsonjson_local
uuid	00000000-0000-0000-0000-000000000000
engine	MergeTree
is_temporary	0
data_paths	/var/lib/clickhouse/data/default/iotjetsonjson_local/
	/var/lib/clickhouse/metadata/default/iotjetsonjson_local.sql
metadata_modification_time	2022-04-01 13:44:26
dependencies_database	
dependencies_table	
partition_key	uuid
sorting_key	uuid
primary_key	uuid
sampling_key	

DONE

streamnative.io

StreamNative Cloud International Organization - sdev Instance

tim.spann@streamnative.io

Search Instances Create instance

aws Ready

aws

[Cluster Details](#)

[Overview](#)

Tenants	Namespaces	Topics
5	4	7

Producers	Subscriptions
0	0

Usage

In Rate	Out Rate	In Throughput
0	0	0

Out Throughput	Storage Size
0	0

gke Ready

gke

[Cluster Details](#)

[Overview](#)

Tenants	Namespaces	Topics
3	4	21

Producers	Subscriptions
0	0

Usage

In Rate	Out Rate	In Throughput
0	0	0

Out Throughput	Storage Size
0	0

gke-free Ready

gke-free

[Cluster Details](#)

[Overview](#)

Tenants	Namespaces	Topics
3	3	1

Producers	Subscriptions
0	0

Usage

In Rate	Out Rate	In Throughput
0	0	0

Out Throughput	Storage Size
0	0

nyc Ready

harbs

[Cluster Details](#)

[Overview](#)

Tenants	Namespaces	Topics
4	4	7

Producers	Subscriptions
0	0

Usage

In Rate	Out Rate	In Throughput
0	0	0

Out Throughput	Storage Size
0	0

sgconnector Ready

connectors

[Cluster Details](#)

[Overview](#)

Tenants	Namespaces	Topics
3	3	9

Producers	Subscriptions
0	0

Usage

In Rate	Out Rate	In Throughput
0	0	0

Out Throughput	Storage Size
0	0

StreamNative Cloud @International ▾ Organization - sndev ▾ Instance - gke ▾ tim.spann@streamnative.io ▾

gke

- Tenants
- Namespaces
- Topics
- SQL
- Clients
- Connector
- Manage
- Service Accounts
- Flink Clusters
- Pulsar Clusters
- Instance Setting

Tenant: public Namespace: default

OVERVIEW TOPICS POLICY

+ New Topic

Topic	Partitions	Domain	Producers	Subscriptions	In Rate	Out Rate	In Throughput	Out Throughput	Storage Size
> product	5	persistent	0	0	0	0	0	0	0
> iotjetsonjson	0	persistent	0	0	0	0	0	0	0
> jetsoniottss2	5	persistent	0	0	0	0	0	0	0
> kinesis-output	0	persistent	0	0	0	0	0	0	0
> test1	0	persistent	0	0	0	0	0	0	0
> jetsoniotts	5	persistent	0	0	0	0	0	0	0
> kinesis-input	0	persistent	0	0	0	0	0	0	0
> sensors	0	persistent	0	0	0	0	0	0	0
> test3	0	persistent	0	0	0	0	0	0	0
> [TENANT_NAMESPACE_TO_PIC]	0	persistent	0	0	0	0	0	0	0
> topitems3	5	persistent	0	0	0	0	0	0	0
> data-gen-out	0	persistent	0	0	0	0	0	0	0

StreamNative Cloud International Organization - sndev Instance - gke tim.spann@streamnative.io

gke

- Tenants
- Namespaces
- Topics**
- SQL
- Clients
- Connector
- Manage
- Service Accounts
- Flink Clusters
- Pulsar Clusters
- Instance Setting

Tenant: public Namespace: default Topic: iotjetsonjson

OVERVIEW SCHEMA MESSAGES **STORAGE** POLICIES

Storage Size: 2 MB

Entries: 2K

Segments: 17

Segments

Ledger ID	Entries	Size	Status	Offload
1342	5	4.51K	closing	false
1353	1	882.00	closing	false
1365	1	940.00	closing	false
1378	2	1.9K	closing	false
1386	1	898.00	closing	false
1975	3	2.79K	closing	false
2007	1	898.00	closing	false
2284	1	805.00	closing	false

SOURCE CODE



```
DATE=$(date +"%Y-%m-%d_%H%M")
python3 -W ignore /home/nvidia/nvme/minifi-jetson-xavier/demo.py --camera /dev/video0 --network
googlenet /home/nvidia/nvme/images/$DATE.jpg 2>/dev/null
java -jar /home/nvidia/nvme/minifi-jetson-xavier/IoTProducer.jar --serviceUrl pulsar://pulsar1:6650
--topic 'persistent://public/default/iotjetsonjson' --message "`tail -1
/home/nvidia/nvme/logs/demo1.log`"
```

Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
Setting default log level to "WARN".

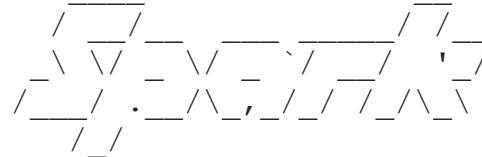
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).

Spark context Web UI available at http://pulsar1.fios-router.home:4040

Spark context available as 'sc' (master = spark://pulsar1:7077, app id = app-20220204140604-0000).

Spark session available as 'spark'.

Welcome to

The logo consists of a stylized 'S' shape formed by a series of diagonal lines and brackets, followed by the text "version 3.2.0".

version 3.2.0

Using Scala version 2.12.15 (OpenJDK 64-Bit Server VM, Java 1.8.0_312)



```
val dfPulsar = spark.readStream
  .format("pulsar")
  .option("service.url", "pulsar://localhost:6650")
  .option("admin.url", "http://localhost:8080")
  .option("topic", "persistent://public/default/iotjetsonjson")
  .load()
```



```
dfPulsar.printSchema()
root
|-- camera: string (nullable = true)
|-- cpu: double (nullable = false)
|-- cputemp: string (nullable = true)
|-- cputempf: string (nullable = true)
|-- diskusage: string (nullable = true)
|-- filename: string (nullable = true)
|-- gputemp: string (nullable = true)
|-- gputempf: string (nullable = true)
|-- host: string (nullable = true)
|-- host_name: string (nullable = true)
|-- imageinput: string (nullable = true)
|-- ipaddress: string (nullable = true)
|-- macaddress: string (nullable = true)
|-- memory: double (nullable = false)
|-- networktime: double (nullable = false)
|-- runtime: string (nullable = true)
|-- systemtime: string (nullable = true)
|-- te: string (nullable = true)
|-- top1: string (nullable = true)
|-- top1pct: double (nullable = false)
|-- uuid: string (nullable = true)
```

```
|-- __key: binary (nullable = true)
|-- __topic: string (nullable = true)
|-- __messageId: binary (nullable = true)
|-- __publishTime: timestamp (nullable = true)
|-- __eventTime: timestamp (nullable = true)
|-- __messageProperties: map (nullable = true)
|   |-- key: string
|   |-- value: string (valueContainsNull = true)
```

```
val pQuery = dfPulsar.selectExpr("*")
.writeStream.format("console")
.option("truncate", "false").start()

|camera      |cpu    |cputemp|cputempf|diskusage |filename
|gputemp|gputempf|host          |host_name        |imageinput
|macaddress     |memory|networktime       |runtime|systemtime
|uuid           |__key
|__topic
|__eventTime|__messageProperties|
+-----+-----+-----+-----+
| /dev/video0|37.2|29.5   |85      |24512.6 MB|/home/nvidia/nvme/images/out_video0_fax_20220204191405.jpg|30.0   |186
|nvidia-desktop|nvidia-desktop|/home/nvidia/nvme/images/img_video0_cok_20220204191405.jpg|192.168.1.228|70:66:55:15:b4:a5|77.9  |25.066272735595703|6      |02/04/2022 14:14:10|5.91699481010437|window
screen|26.8310546875|xav_uuid_video0_jfh_20220204191405|[78 61 76 5F 75 75 69 64 5F 76 69 64 65 6F 30 5F 6A 66 68 5F 32
30 32 32 30 32 34 31 39 31 34 30 35]|persistent://public/default/iotjetsonjson|[08 D1 A0 05 10 00 20 00]|2022-02-04
14:14:16.517|null      |{}|
```





```
CREATE TABLE iotjetsonjson
(
    `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, ipaddress STRING, macaddress STRING,
    gputempf STRING, host_name STRING, camera STRING, filename STRING,
    `runtime` STRING, cpu STRING, cputempf STRING, imageinput STRING,
    `networktime` STRING, top1 STRING, top1pct STRING,
    publishTime TIMESTAMP(3) METADATA,
    WATERMARK FOR publishTime AS publishTime - INTERVAL '5' SECOND
) WITH (
    'connector' = 'pulsar',
    'topic' = 'persistent://public/default/iotjetsonjson',
    'value.format' = 'json',
    'scan.startup.mode' = 'earliest',
    'service-url' = 'pulsar://pulsar1:6650',
    'admin-url' = 'http://pulsar1:8080'
);
```



```
bin/pulsar sql
```

```
show tables in pulsar."public/default";
```

```
select * from pulsar."public/default".iotjetsonjson  
order by systemtime desc;
```

ARTICLE



Q&A



[Webinar]
Building Microservices

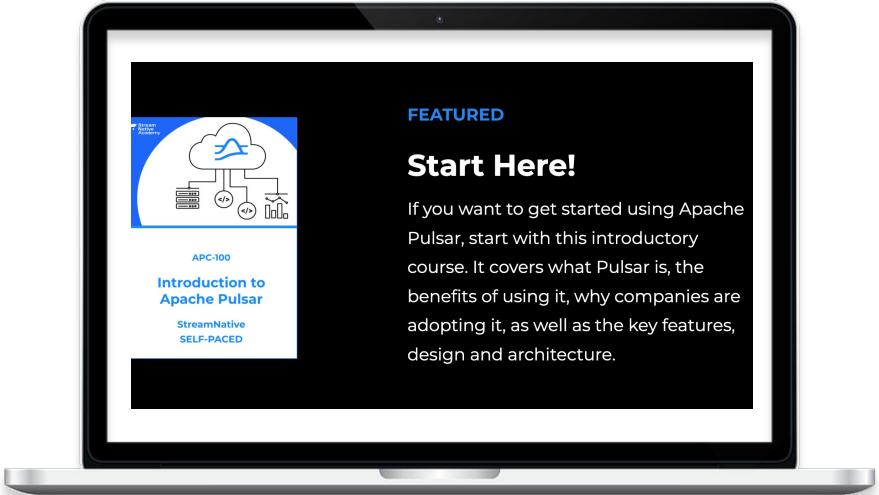
[Watch Now](#)

Developing Event-Driven
Microservices with
[Apache Pulsar: Part I](#)



[Blog post]
Event-Driven Microservices

[Learn More](#)



Now Available

On-Demand Pulsar Training

Academy.StreamNative.io

Let's Keep in Touch!



Tim Spann

Developer Advocate



@PassDev



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



<https://github.com/tspannhw>