

FLiP Into Apache Pulsar Apps with ScyllaDB

Timothy Spann

Developer Advocate











Tim Spann Developer Advocate

- FLiP/FLiPN/FLiPS Stacks
- Ex-Principal Field Engineer Cloudera
- Ex-Senior Solutions Engineer Hortonworks
- Ex-Field Engineer Pivotal

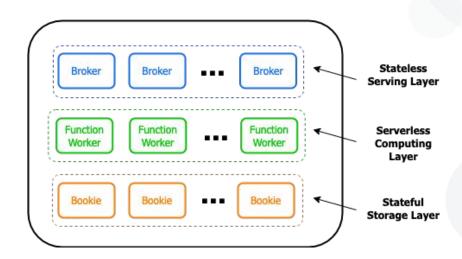




Cloud Streaming

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



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

Streaming FLiPS Apps

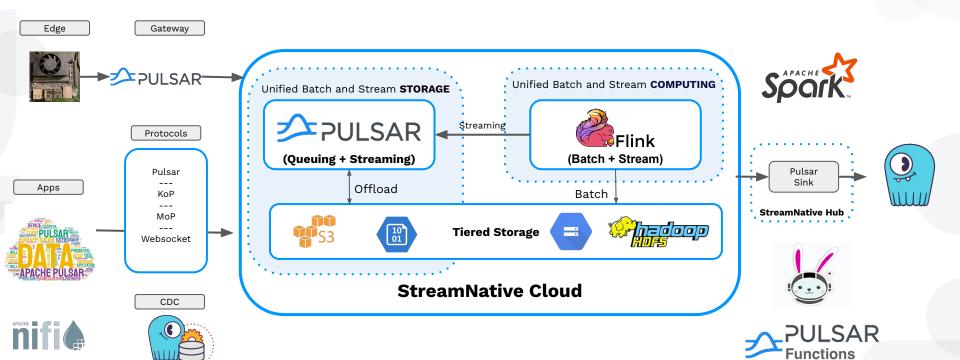


SCYLL∧ <-> Events <-> → PULSAR <-> Events <->







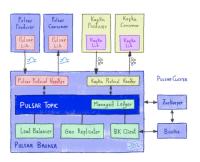


Connect



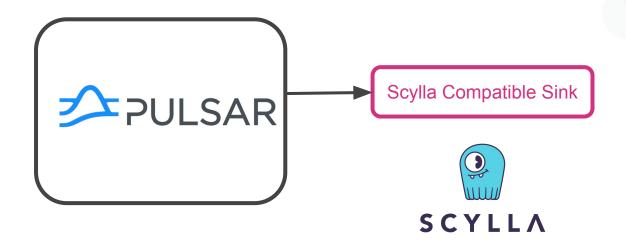
Connectivity

hub.streamnative.io



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

Scylla Compatible Sink Connector



https://pulsar.apache.org/docs/en/io-quickstart/

Build and Run

Config

```
configs:
    roots: "172.17.0.2:9042"
    keyspace: "pulsar_test_keyspace"
    columnFamily: "pulsar_test_table"
    keyname: "key"
    columnName: "col"
```

Define

```
CREATE KEYSPACE pulsar_test_keyspace WITH replication = {'class':'SimpleStrategy', 'replication_factor':1};
CREATE TABLE pulsar_test_table (key text PRIMARY KEY, col text);
CREATE INDEX ON pulsar_test_table(col);
```

Deploy

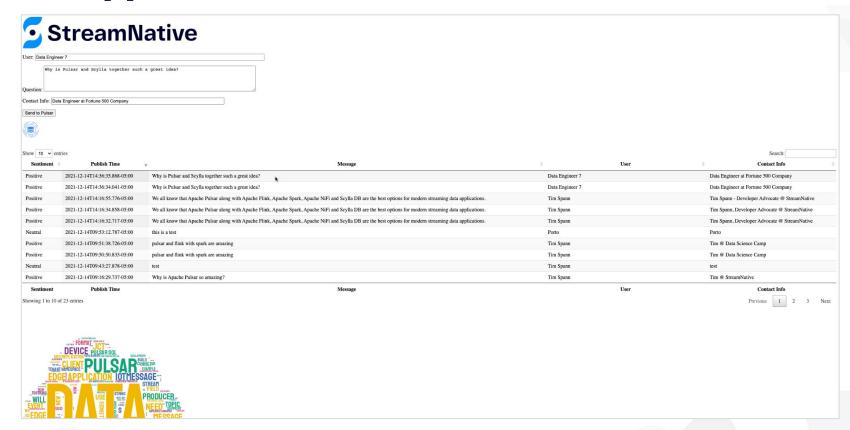
```
bin/pulsar-admin sinks create --tenant public --namespace default --name
"scylla-test-sink" \
    --sink-type cassandra --sink-config-file conf/scylla.yml --inputs
chatresult2
```

Deploy ML Function

```
• • •
bin/pulsar-admin functions stop --name Chat --namespace default --tenant public
bin/pulsar-admin functions delete --name Chat --namespace default --tenant
public
bin/pulsar-admin functions create --auto-ack true --py pulsar-pychat-
function/src/sentiment.py --classname "sentiment.Chat" --inputs
"persistent://public/default/chat2" --log-topic
"persistent://public/default/chatlog2" --name Chat --namespace default --output
"persistent://public/default/chatresult2" --tenant public
```

Your App

https://github.com/tspannhw/ScyllaFLiPSTheStream



Future

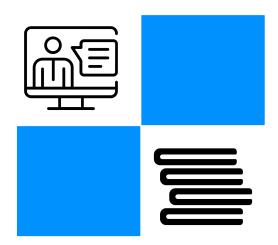
Shard-Aware Pulsar Connector for Scylla



What's Next?

Continue your journey with Apache Pulsar:

- Get started with Pulsar Training from StreamNative Academy
- Watch the 3-part on-demand webinar series Building Event-driven
 Microservices with Pulsar
- Join the Pulsar community on <u>Slack</u>



Thank you!

Stay in touch

Timothy Spann



@PaaSDev



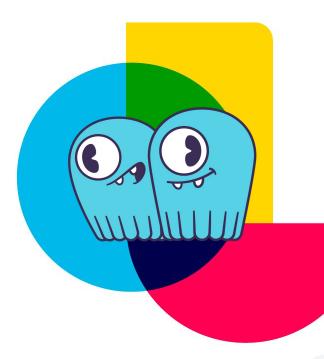
tspannhw



tim@streamnative.io



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



Demo



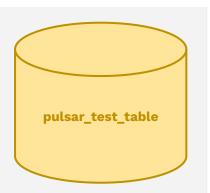








chat



Sink Status



```
. . .
  "numRunning" : 1,
     "instanceId" : 0,
     "status" : {
        "running" : true,
        "numReadFromPulsar" : 6,
       "numSystemExceptions" : 0,
"latestSystemExceptions" : [ ],
        "numSinkÉxceptions" : 0,
"latestSinkExceptions" : [ ],
        "numWrittenToSink" : 6,
"lastReceivedTime" : 1639513221745,
        "workerId" : "c-standalone-fw-127.0.0.1-8080"
```