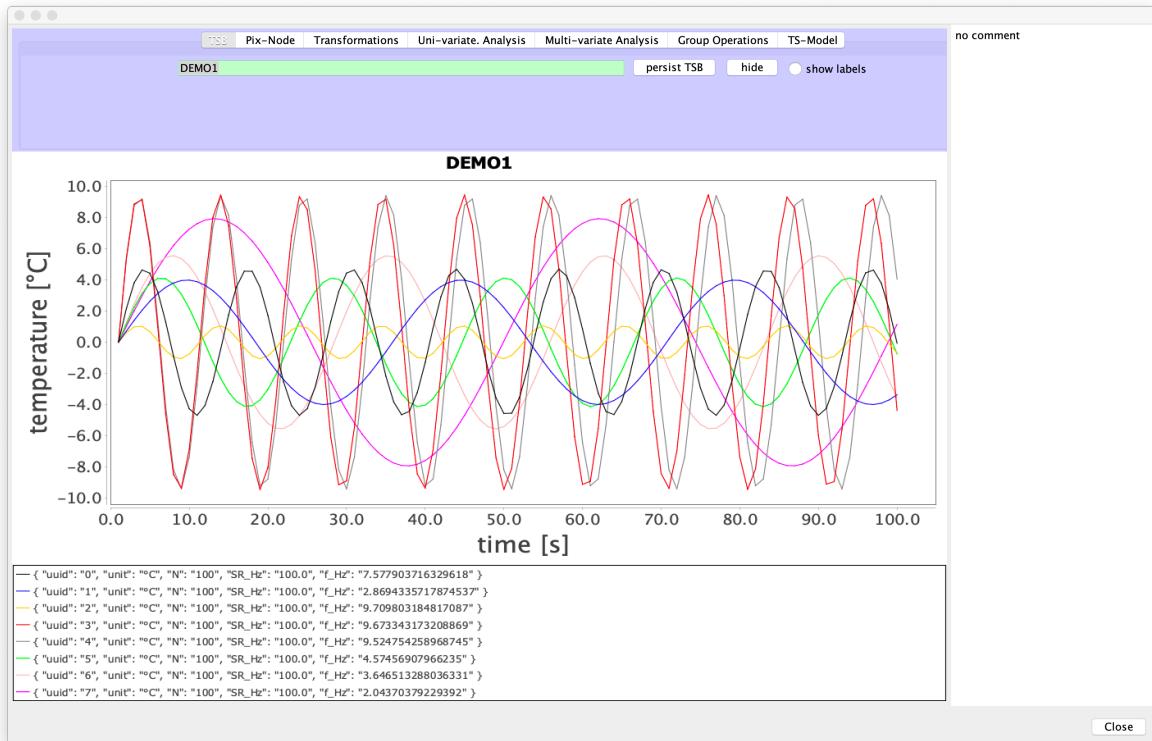


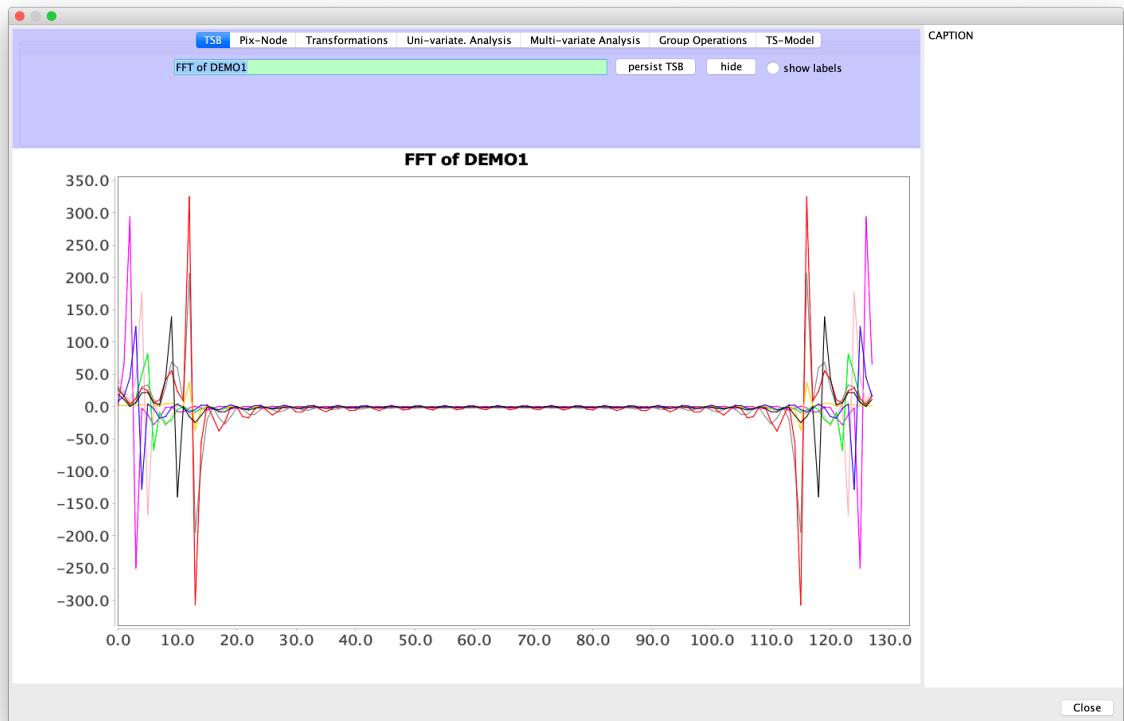
Prepare a Cluster with Demo Data:

Local CP installation and OpenTSx flows

Initial data sets

8 time series with different frequencies, sampling rate 100 Hz





FFT of raw episodes

The screenshot shows the Confluent Control Center interface for a cluster named 'CONTROLCENTER.CLUS...'. The left sidebar has a 'Topics' section selected. The main area is titled 'All topics' and displays a table of five topics:

Topics Topic name	Availability			Throughput	
	Under replicated partitions	Out of sync followers	Out of sync observers	Bytes/sec produced	Bytes/sec consumed
OpenTSx_Episodes_A	0 of 1	0 of 1	0 of 0	0	0
OpenTSx_Episodes_B	0 of 1	0 of 1	0 of 0	0	0
OpenTSx_Event_Flow_State	0 of 1	0 of 1	0 of 0	0	--
OpenTSx_Events	0 of 1	0 of 1	0 of 0	0	0
default_ksql_processing_log	0 of 1	0 of 1	0 of 0	--	--

A search bar at the top of the table says 'Search topics' and a button says '+ Add a topic'.

4 topics with raw data

The screenshot shows the Confluent Control Center interface for the 'OpenTSX_Events' topic. The left sidebar lists various cluster settings and topics. The main area displays metrics like Bytes in/sec and Bytes out/sec, and a detailed view of message fields. The 'Messages' tab is selected, showing a list of messages with their timestamps, offsets, and keys. Each message is represented by a blue box containing JSON data.

```
1 {  
2   "timestamp": 1599560421900,  
3   "uri": "http://www.w3.org/TR/2004/metrics-owl-20200210/deviceID#0015a",  
4   "value": 3.55122115120929  
5 }  
  
Partition: 0 Offset: 15 Timestamp: 1599560398438  
  
Partition: 0 Offset: 14 Timestamp: 1599560398437  
  
Partition: 0 Offset: 13 Timestamp: 1599560398436  
  
Partition: 0 Offset: 12 Timestamp: 1599560398435  
  
Partition: 0 Offset: 11 Timestamp: 1599560398435
```

Structured value (Avro)

This screenshot shows a single message from the 'OpenTSX_Events' topic. It highlights the 'Key' tab, which displays an Avro-structured key as a JSON object. Below the key, the partition, offset, and timestamp are shown.

```
{ "uuid": "0", "unit": "°C", "N": "100", "SR_Hz": "100.0", "f_Hz": "7.577903716329618" }
```

Partition: 0 Offset: 15 Timestamp: 1599560398438

Structured key (JSON string)

Schema for unified data model

Event schema: AVRO

ALL TOPICS >

OpenTSx_Events

Overview Messages **Schema** Configuration

Value Key

[Edit schema](#) [Version history](#) [Download](#)

```
1  {
2    "fields": [
3      {
4        "name": "timestamp",
5        "type": "long"
6      },
7      {
8        "name": "uri",
9        "type": "string"
10     },
11     {
12       "name": "value",
13       "type": "double"
14     }
15   ],
16   "name": "Event",
17   "namespace": "org.opentsx.data.model",
18   "type": "record"
19 }
```

Episode schema: AVRO

http://127.0.0.1:9021/clusters/tEM8XXu1RDKnllu8clBftA/management/topics/OpenTSx_Episodes_A/schema/value

Prepare Streams over Topics

CONFLUENT

CONTROLCENTER.CLUS... CO Cluster 1

OVERVIEW BROKERS TOPICS CONNECT KSQLDB STREAMS TABLES RUNNING QUERIES

Search streams [+ Add Stream](#)

Stream Name	Kafka topic Topic Name	Partitions	Replication	Data Format
OPENTSX_EVENTS_STREAM_MATERIALIZED	OPENTSX_EVENTS_STREAM_MATERIALIZED	1	1	AVRO
OPENTSX_EVENT_FLOW_STATE_STREAM	OpenTSx_Event_Flow_State	1	1	JSON
OPENTSX_EVENTS_STREAM	OpenTSx_Events	1	1	AVRO
OPENTSX_EPISODES_A_STREAM	OpenTSx_Episodes_A	1	1	AVRO
OPENTSX_EPISODES_B_STREAM	OpenTSx_Episodes_B	1	1	AVRO
KSQL_PROCESSING_LOG	default_ksql_processing_log	1	1	JSON

5 streams available

CONFLUENT

CONTROL CENTER CLUSTER

All topics

Search topics Hide internal topics + Add a topic

Topics	Topic name	Availability	Under replicated partitions	Out of sync followers	Out of sync observers	Throughput	Bytes/sec produced	Bytes/sec consumed
	OPENTSX_EVENTS_STREAM_MATERIALIZED	0 of 1	0 of 1	0 of 0	--	0	0	0
	OpenTSx_Episodes_A	0 of 1	0 of 1	0 of 0	0	0	0	0
	OpenTSx_Episodes_B	0 of 1	0 of 1	0 of 0	0	0	0	0
	OpenTSx_Event_Flow_State	0 of 1	0 of 1	0 of 0	0	--	0	0
	OpenTSx_Events	0 of 1	0 of 1	0 of 0	0	0	0	0
	default_ksql_processing_log	0 of 1	0 of 1	0 of 0	--	--	0	0

1 new topic for the materialised stream available

Prepare a local KSQLDB instance

```
ksql> show streams;
```

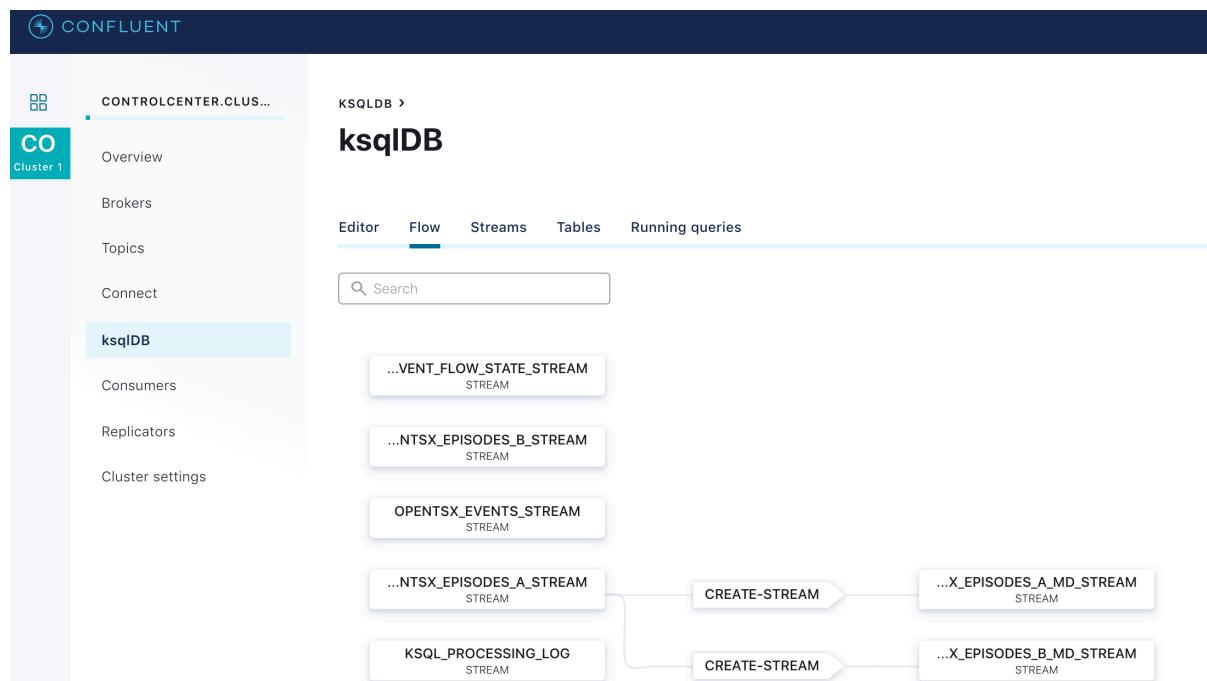
Stream Name	Kafka Topic	Format
KSQl_PROCESSING_LOG	default_ksql_processing_log	JSON
OPENTSX_EPISODES_A_STREAM	OpenTSx_Episodes_A	AVRO
OPENTSX_EPISODES_B_STREAM	OpenTSx_Episodes_B	AVRO
OPENTSX_EVENTS_STREAM	OpenTSx_Events	AVRO
OPENTSX_EVENTS_STREAM_MATERIALIZED	OPENTSX_EVENTS_STREAM_MATERIALIZED	AVRO
OPENTSX_EVENT_FLOW_STATE_STREAM	OpenTSx_Event_Flow_State	JSON

```
ksql> show topics;
```

Kafka Topic	Partitions	Partition Replicas
OPENTSX_EVENTS_STREAM_MATERIALIZED	1	1
OpenTSx_Episodes_A	1	1
OpenTSx_Episodes_B	1	1
OpenTSx_Event_Flow_State	1	1
OpenTSx_Events	1	1
default_ksql_processing_log	1	1

```
ksql> [REDACTED]
```

Inspect streams via CLI



Flows can easily be inspected in the web UI.

Create Kafka Stream Processing applications

```
Topologies:  
Sub-topology: 0  
Source: KSTREAM-SOURCE-0000000000 (topics: [OpenTSx_Events])  
    --> KSTREAM-MAPVALUES-0000000001  
Processor: KSTREAM-MAPVALUES-0000000001 (stores: [])  
    --> KSTREAM-PRINTER-0000000003, KSTREAM-SINK-0000000002  
    <-- KSTREAM-SOURCE-0000000000  
Processor: KSTREAM-PRINTER-0000000003 (stores: [])  
    --> none  
    <-- KSTREAM-MAPVALUES-0000000001  
Sink: KSTREAM-SINK-0000000002 (topic: EVENT_DATA_TRAFO_meetup_09_2020)  
    <-- KSTREAM-MAPVALUES-0000000001
```

TSAExample1

```
Topologies:  
Sub-topology: 0  
Source: KSTREAM-SOURCE-0000000000 (topics: [OpenTSx_Events])  
    --> KSTREAM-FILTER-0000000001  
Processor: KSTREAM-FILTER-0000000001 (stores: [])  
    --> KSTREAM-PRINTER-0000000003, KSTREAM-SINK-0000000002  
    <-- KSTREAM-SOURCE-0000000000  
Processor: KSTREAM-PRINTER-0000000003 (stores: [])  
    --> none  
    <-- KSTREAM-FILTER-0000000001  
Sink: KSTREAM-SINK-0000000002 (topic: EVENT_DATA_ALERTS_meetup_09_2020)  
    <-- KSTREAM-FILTER-0000000001
```

TSAExample2

```
Topologies:  
Sub-topology: 0  
Source: KSTREAM-SOURCE-0000000000 (topics: [EVENT_DATA_TRAFO_meetup_09_2020])  
    --> KSTREAM-FILTER-0000000001  
Processor: KSTREAM-FILTER-0000000001 (stores: [])  
    --> KSTREAM-PRINTER-0000000003, KSTREAM-SINK-0000000002  
    <-- KSTREAM-SOURCE-0000000000  
Processor: KSTREAM-PRINTER-0000000003 (stores: [])  
    --> none  
    <-- KSTREAM-FILTER-0000000001  
Sink: KSTREAM-SINK-0000000002 (topic: EVENT_DATA_COMPACT_ALERTS_meetup_09_2020)  
    <-- KSTREAM-FILTER-0000000001
```

TSAExample3

Topologies:

Sub-topology: 0

Source: KSTREAM-SOURCE-0000000000 (topics: [OpenTSx_Episodes_B])

---> KSTREAM-MAPVALUES-0000000001

Processor: KSTREAM-MAPVALUES-0000000001 (stores: [])

---> KSTREAM-PRINTER-0000000003, KSTREAM-SINK-0000000002

<-- KSTREAM-SOURCE-0000000000

Processor: KSTREAM-PRINTER-0000000003 (stores: [])

---> none

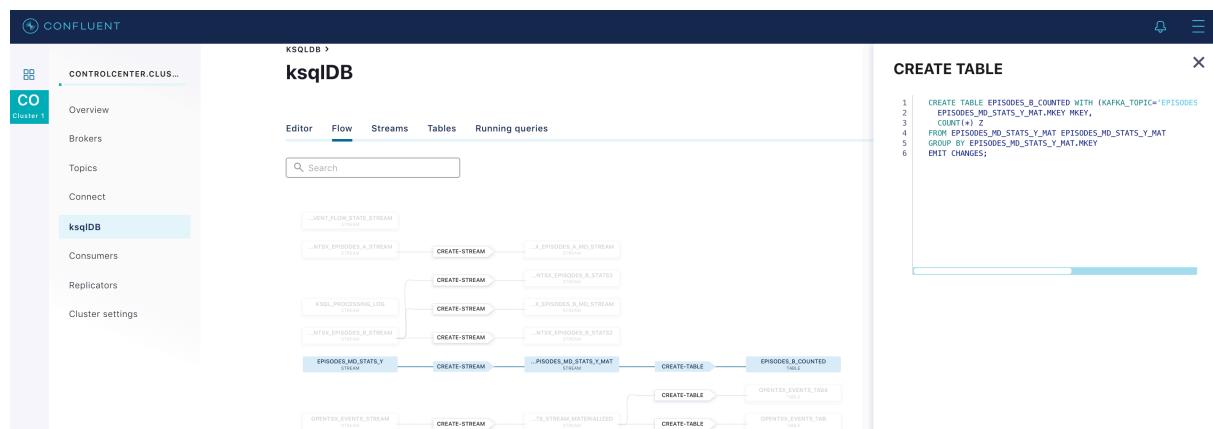
<-- KSTREAM-MAPVALUES-0000000001

Sink: KSTREAM-SINK-0000000002 (topic: EPISODES_DATA_B_SIMPLE_STATS_meetup_09_2020)

<-- KSTREAM-MAPVALUES-0000000001

TSAExample10

Data flow visualization



Episodes statistics based on custom properties, calculated in a Kafka streams application.

```

1 CREATE STREAM OPENTSX_EPISODES_B_STATS2 WITH (KAFKA_TOPIC='OPENTSX_EPISODES_B_STATS2', PARTITIONS=1, REPLICAS=1) AS SELECT
2   OPENTSX_EPISODES_B_STREAM.LABEL LABEL,
3   INDEXOF(OPENTSX_EPISODES_B_STREAM.TSTART, OPENTSX_EPISODES_B_STREAM.TEND) DT
4   FROM OPENTSX_EPISODES_B_STREAM OPENTSX_EPISODES_B_STREAM
5   EMIT CHANGES;
6
7
8

```

Query ID: CSAS_OPENTSX_EPISODES_B_STATS2 Sink: OPENTSX_EPISODES_B_STATS2 Source: OPENTSX_EPISODES_B_STREAM Explain Terminate

```

1 CREATE TABLE EPISODES_B_COUNTED WITH (KAFKA_TOPIC='EPISODES_B_COUNTED', PARTITIONS=1, REPLICAS=1) AS SELECT
2   EPISODES_MO_STATS_Y_MAT.MKEY MKEY,
3   COUNT(*) Z
4   FROM EPISODES_MO_STATS_Y_MAT EPISODES_MO_STATS_Y_MAT
5   GROUP BY EPISODES_MO_STATS_Y_MAT.MKEY
6   EMIT CHANGES;
7
8

```

Query ID: CTAS_EPISODES_B_COUNTED_27 Sink: EPISODES_B_COUNTED Source: EPISODES_MO_STATS_Y_MAT Explain Terminate

```

1 CREATE TABLE OPENTSX_EVENTS_TAB WITH (KAFKA_TOPIC='OPENTSX_EVENTS_TAB', PARTITIONS=1, REPLICAS=1) AS SELECT
2   OPENTSX_EVENTS_STREAM_MATERIALIZED.URI URI,
3   COUNT(*) Z
4   FROM OPENTSX_EVENTS_STREAM_MATERIALIZED OPENTSX_EVENTS_STREAM_MATERIALIZED
5   WINDOW TUMBLING ( SIZE 10 SECONDS )
6   GROUP BY OPENTSX_EVENTS_STREAM_MATERIALIZED.URI
7   EMIT CHANGES;
8

```

Query ID: CTAS_OPENTSX_EVENTS_TAB_17 Sink: OPENTSX_EVENTS_TAB Source: OPENTSX_EVENTS_STREAM_MATERIALIZED Explain Terminate

Some running queries ... without data

```

1 CREATE STREAM OPENTSX_EVENTS_STREAM_MATERIALIZED WITH (KAFKA_TOPIC='OPENTSX_EVENTS_STREAM_MATERIALIZED', PARTITIONS=1, REPLICAS=1) AS SELECT *
2 FROM OPENTSX_EVENTS_STREAM OPENTSX_EVENTS_STREAM
3
4
5
6
7
8

```

Query ID: CSAS_OPENTSX_EVENTS_STREAM_MATERIALIZED_0 Sink: OPENTSX_EVENTS_STREAM_MATERIALIZED
Messages/sec: 2408.40 Source: OPENTSX_EVENTS_STREAM

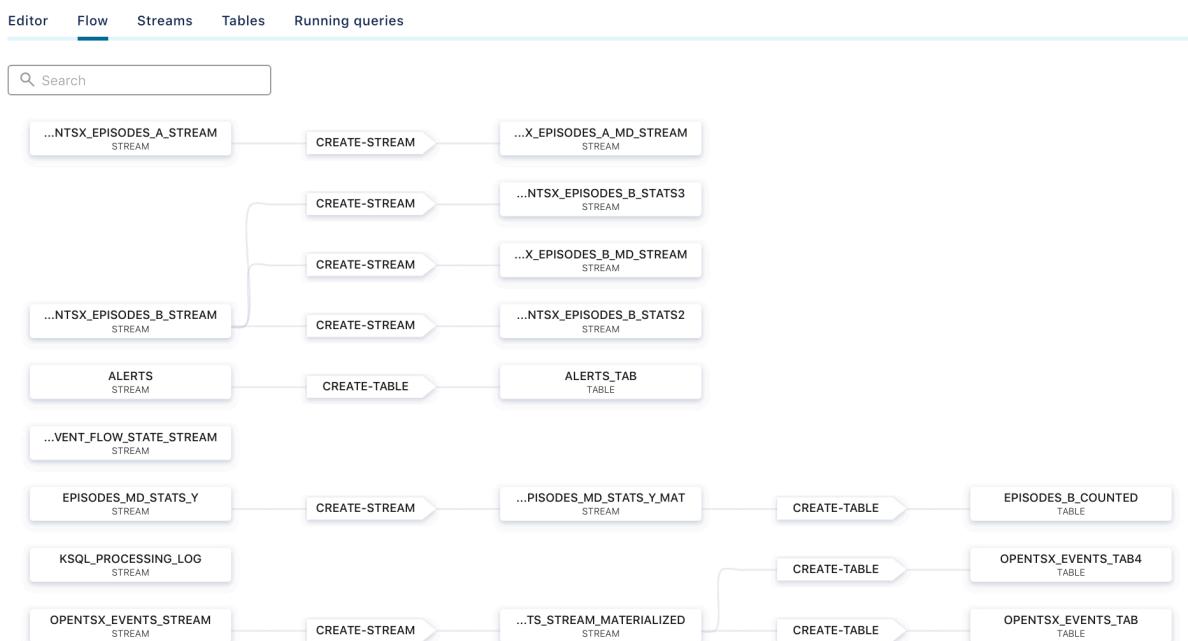
[Explain](#) [Terminate](#)

... and now some data flows through the system, 2400 events per second.

Alert processing

Final overview of flows:

ksqldb



Topologies:

Sub-topology: 0

Source: KSTREAM-SOURCE-0000000000 (topics: [OpenTSx_Events])

--> KSTREAM-FILTER-0000000001

Processor: KSTREAM-FILTER-0000000001 (stores: [])

--> KSTREAM-PRINTER-0000000003, KSTREAM-SINK-0000000002

<-- KSTREAM-SOURCE-0000000000

Processor: KSTREAM-PRINTER-0000000003 (stores: [])

--> none

<-- KSTREAM-FILTER-0000000001

Sink: KSTREAM-SINK-0000000002 (topic: EVENT_DATA_ALERTS_meetup_09_2020)

<-- KSTREAM-FILTER-0000000001

TSAExample2

```
#####
#
# Count alerts by sensor and time-window
#
#####

CREATE STREAM alerts WITH (kafka_topic='EVENT_DATA_ALERTS_meetup_09_2020', value_format='AVRO');
select * from alerts emit changes;
CREATE TABLE ALERTS_TAB AS SELECT URI, count(*) as z FROM alerts WINDOW TUMBLING (SIZE 15 SECONDS) Group by URI;

#
# Finally, we can use the PULL queries to get the alert counter per sensor instantly.
#
select * from alerts_tab where URI='http://www.w3.org/TR/2004/metrics-owl-20200210/deviceID#0815a';
select * from alerts_tab where URI='http://www.w3.org/TR/2004/metrics-owl-20200210/deviceID#17u4';
```

URI	WINDOWSTART	WINDOWEND	Z
http://www.w3.org/TR/2004/metrics-owl-20200210/deviceID#0815a	1599496845000	1599496845000	234
http://www.w3.org/TR/2004/metrics-owl-20200210/deviceID#0815a	1599496860000	1599496860000	9677
http://www.w3.org/TR/2004/metrics-owl-20200210/deviceID#0815a	1599496875000	1599496875000	11785
http://www.w3.org/TR/2004/metrics-owl-20200210/deviceID#0815a	1599497835000	1599497835000	234
http://www.w3.org/TR/2004/metrics-owl-20200210/deviceID#0815a	1599497850000	1599497850000	3378
http://www.w3.org/TR/2004/metrics-owl-20200210/deviceID#0815a	1599497865000	1599497865000	5613
http://www.w3.org/TR/2004/metrics-owl-20200210/deviceID#0815a	1599497880000	1599497880000	2628
http://www.w3.org/TR/2004/metrics-owl-20200210/deviceID#0815a	1599560400000	1599560400000	234
http://www.w3.org/TR/2004/metrics-owl-20200210/deviceID#0815a	1599560415000	1599560415000	5602