Event Processing without Breaking Production

 @nzender  nathanzender.com


EVENTS ...
EVENTS EVERYWHERE


1 7 5 9 3

Nathan Zender

- Developer for over a decade
- Currently Tech Lead with by Cardinal Health fuse
- OSS Contributor to Micronaut and Sonatype
- Cars and family

@nzender nathanzender.com

What are we trying to solve for?

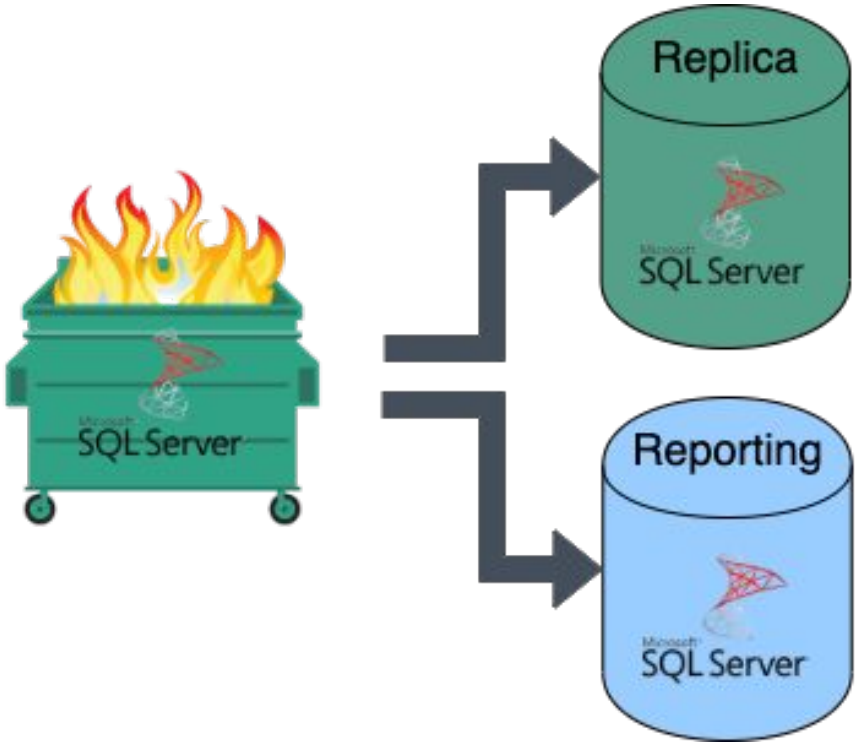
@nzender nathanzender.com

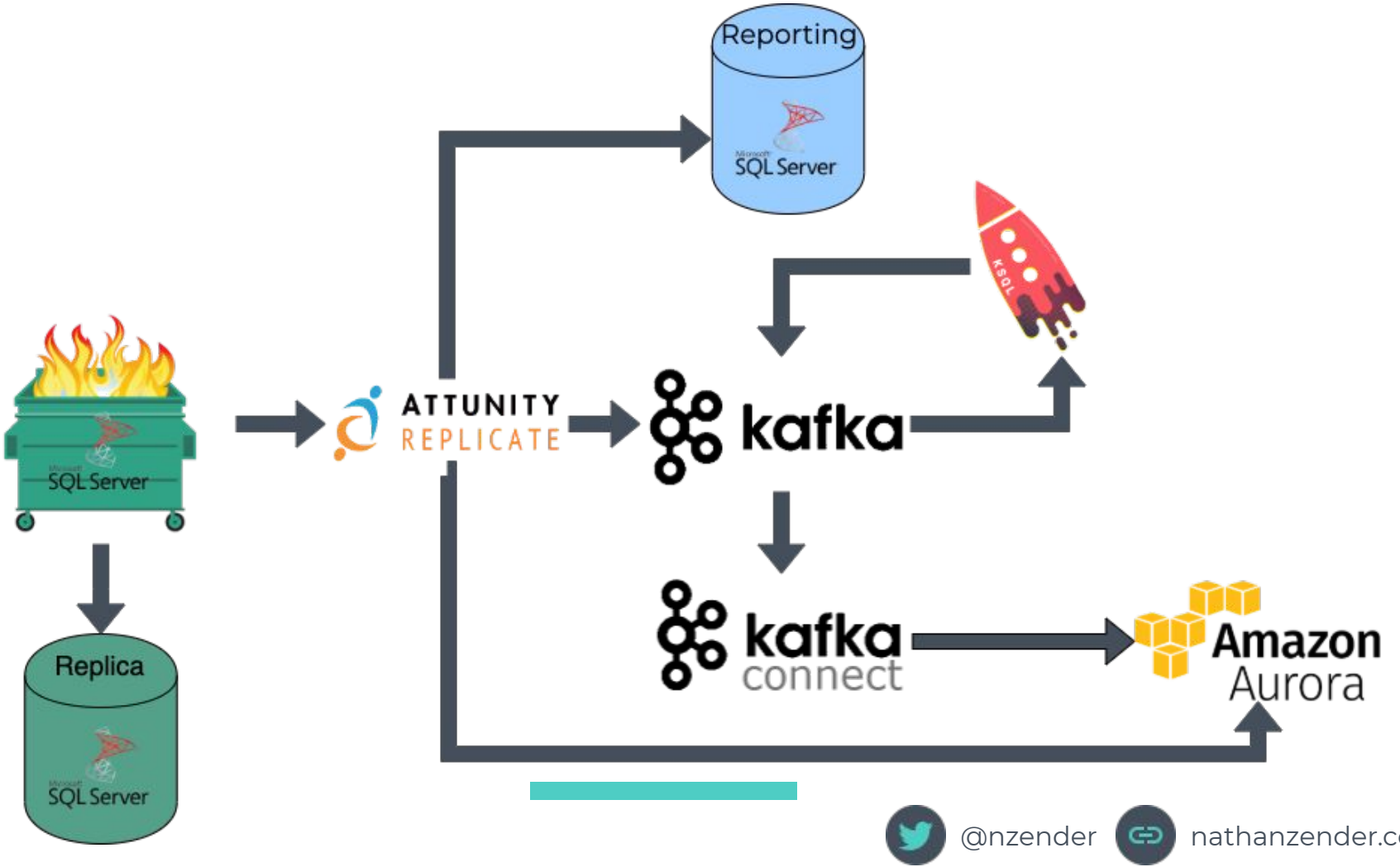
The problems

- Inaccurate reporting due to missing data
- Want to provide deeper insights and analytics
- Cannot impact production performance
- Split the monolith

Architecture

@nzender nathanzender.com

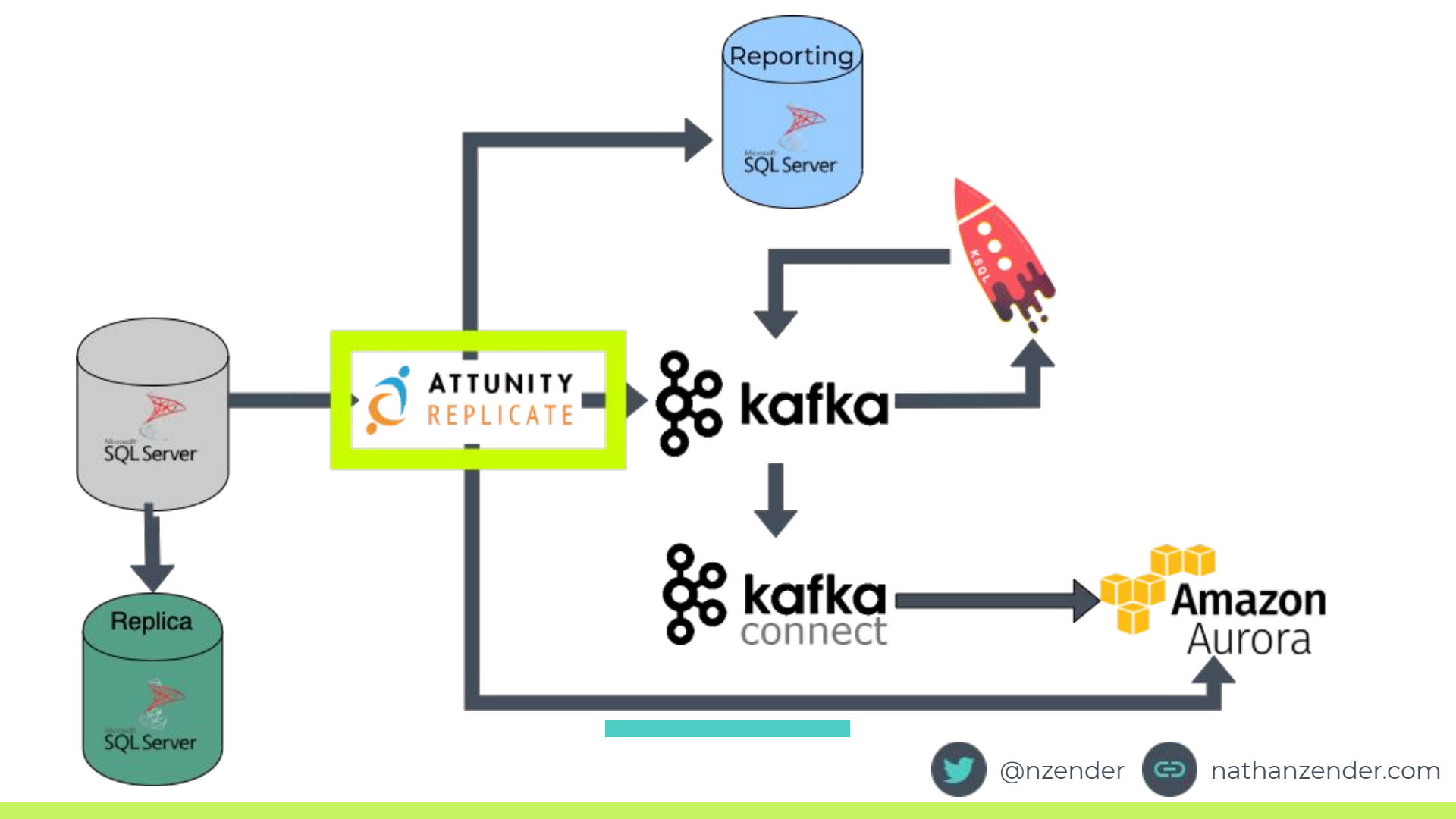

Replica
SQL Server
SQL Server
Reporting
SQL Server



@nzender nathanzender.com

Attunity Replicate

@nzender nathanzender.com



@nzender nathanzender.com

What is it?

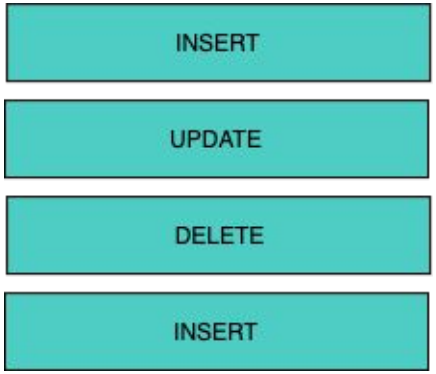
Attunity Replicate provides a unified platform to replicate, synchronize, distribute, consolidate, and ingest data across all major databases, data warehouses and Hadoop, both on-premise and in the cloud.

Why?

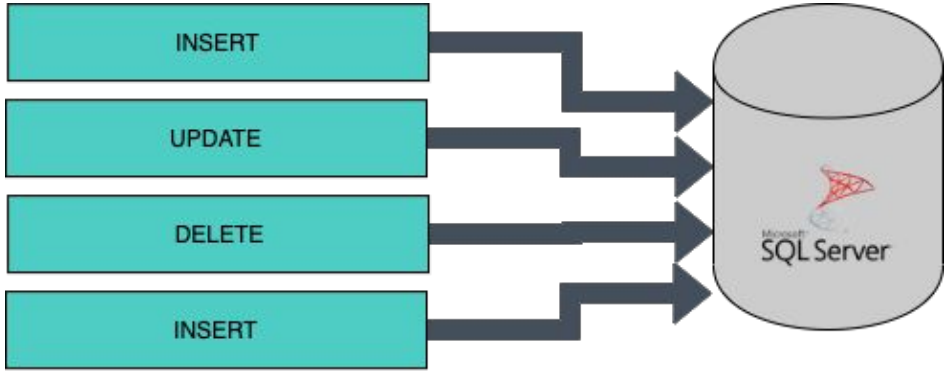
- Extremely low impact to said 

- Ease of use/maintenance
- Supports tons of sources and targets (RDBMS, NoSQL, File, etc, etc)

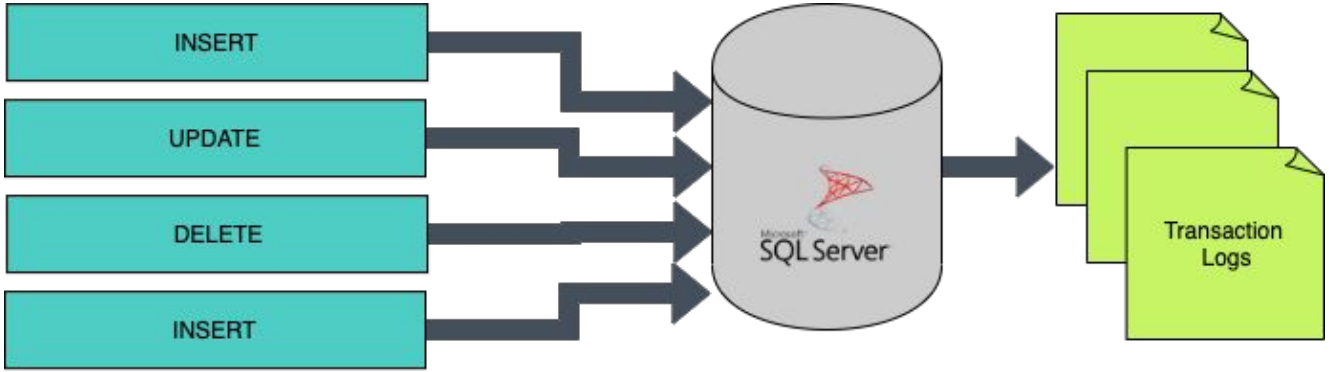
How?



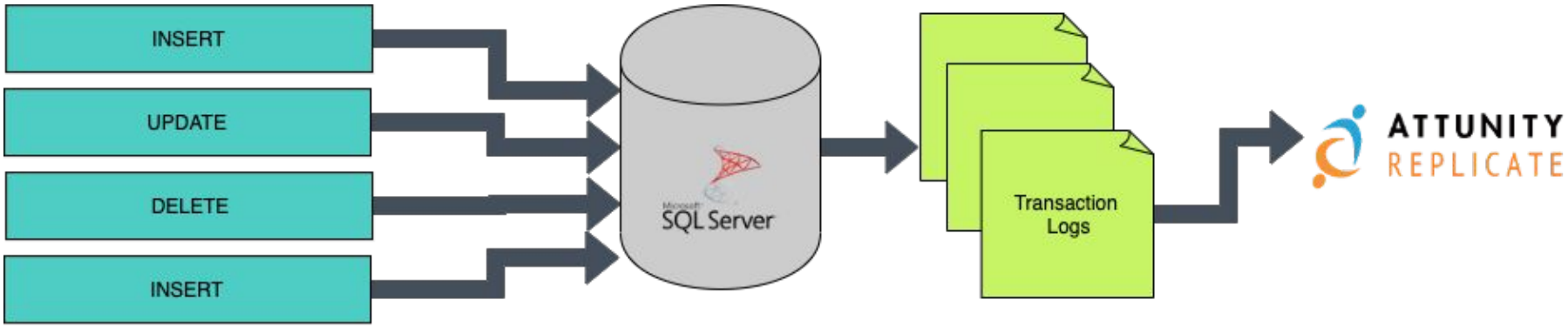
How?



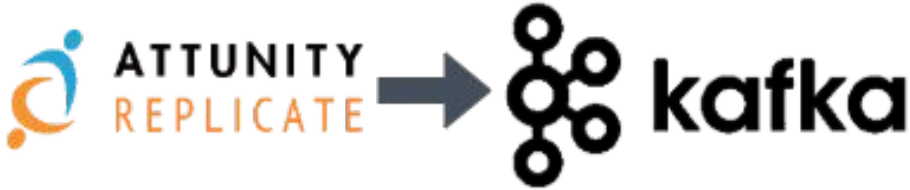
How?



How?

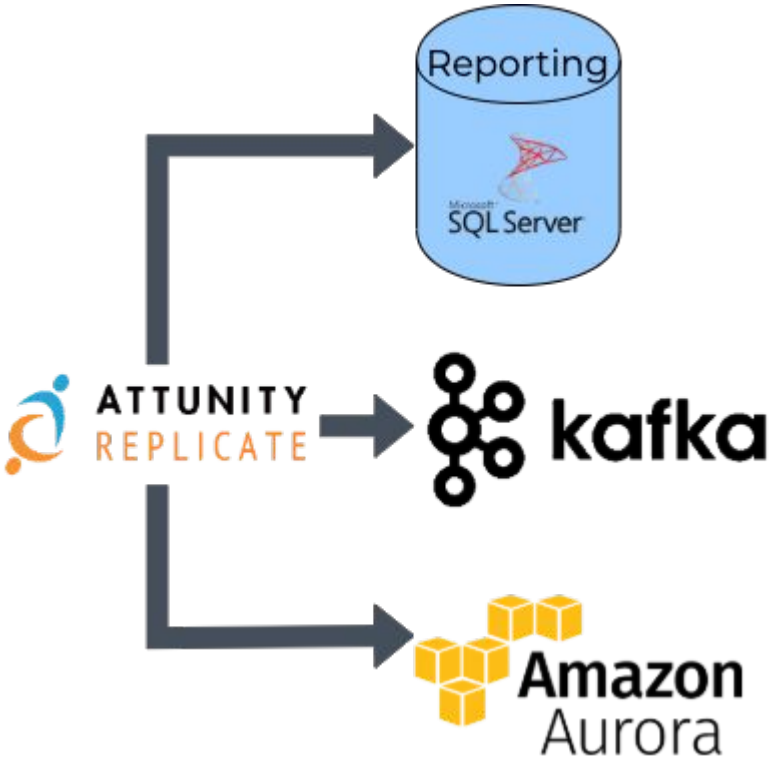

INSERT
UPDATE
DELETE
INSERT
SQL Server
Transaction Logs
ATTUNITY
REPLICATE

How? Part Deux



How? Part Deux

How? Part Deux



But what's the catch?

- Local development environment support
- CPU requirement
  - Strained CPU? Could run into RDBMS issues
  - CPU requirement for reading logs
  - Limited number of tasks
- Upgrading versions has been non-trivial

Open Source Alternative

@nzender nathanzender.com

Debezium

- An open source distributed platform for change data capture
- Can use Change Data Capture to pump data out of MySQL, Postgres or SQL Server into Kafka

Why we didn’t choose it

- At the time SQL Server support was in Alpha status
- CDC had the potential to cause issues with production
- Did not have ability to pump to multiple targets. Would require dev work or finding another tool.

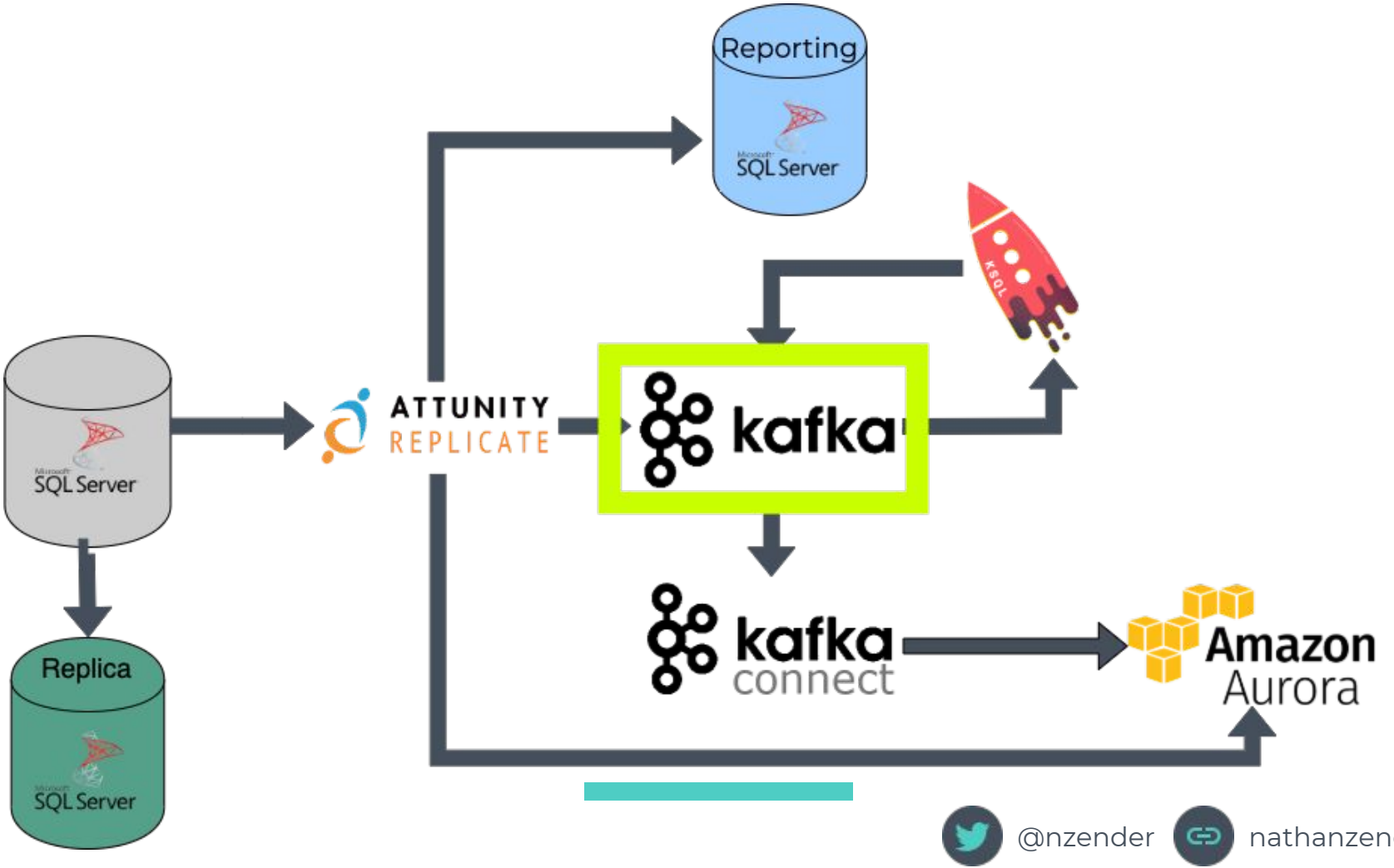
It’s still a good option

- Great talk by Gunnar Morling around the same event data capture topic and using Debezium to do so
  - http://bit.ly/2tpFHEF

Kafka


@nzender

nathanzender.com



@nzender nathanzender.com

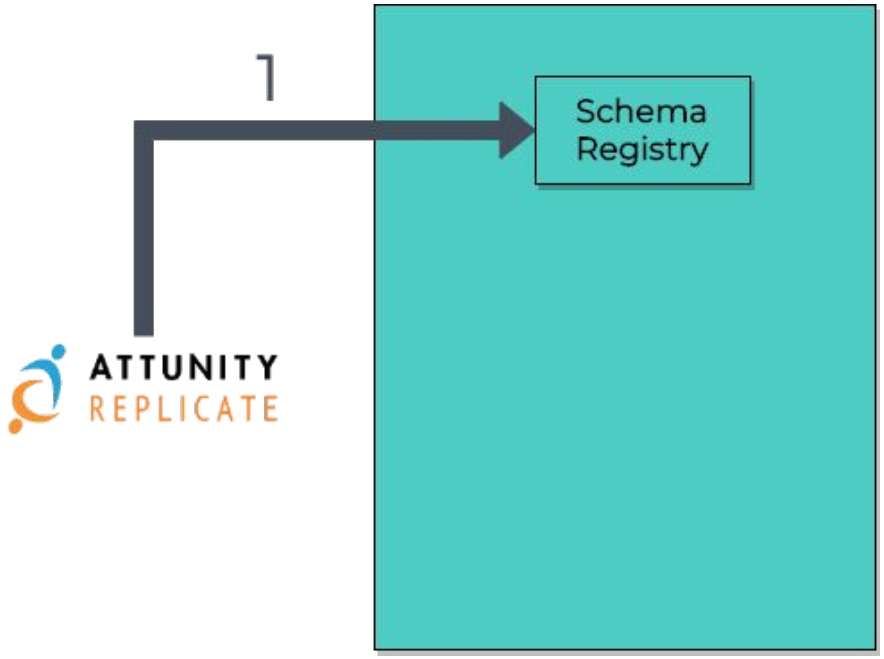
What is it?

Apache Kafka is an open source project for a distributed publish-subscribe messaging system rethought as a distributed commit log. Kafka stores messages in topics that are partitioned and replicated across multiple brokers in a cluster. Producers send messages to topics from which consumers read.

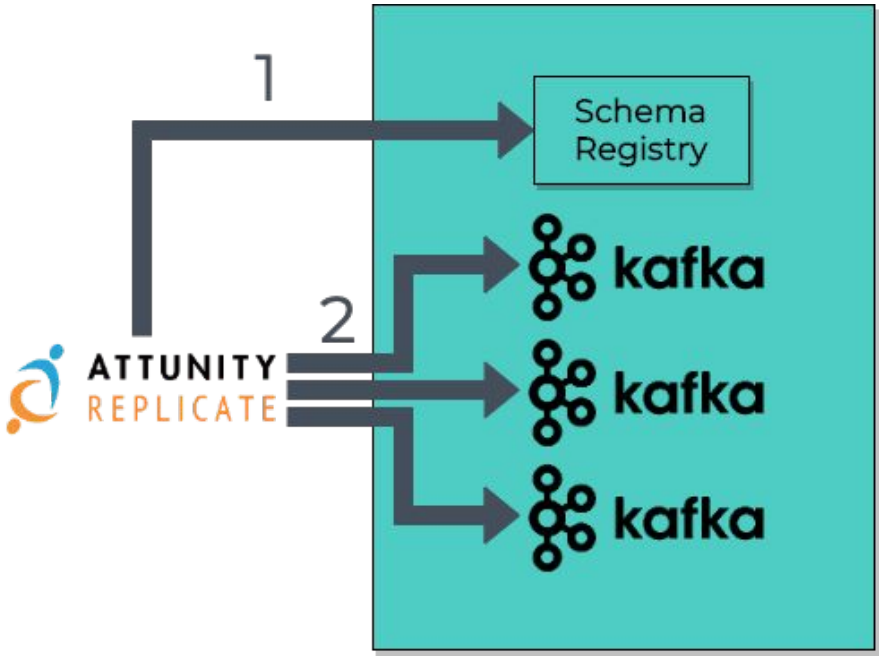
Why?

- Battle tested message system
- Planned to use this as the messaging system between future microservices
- Schema Registry/Avro messages == less disk usage

How?



How?



Schema Registry

- Metadata storage
- Backed by Kafka
- Exposes a REST API to interact with it
- Allows for use of Apache Avro binary messages in Kafka
- Greatly reduces disk usage

Show me some code (Schema)

```
{
   "subject":"dbo.stateCountyCode-value",
   "version":1,
   "id":241,
   "schema":{
      "type":"record",
      "name":"DataRecord",
      "fields":[
         { "name":"data"...},
         { "name":"beforeData"...},
         { "name":"headers"...}
      ]
}}
```


@nzender

nathanzender.com

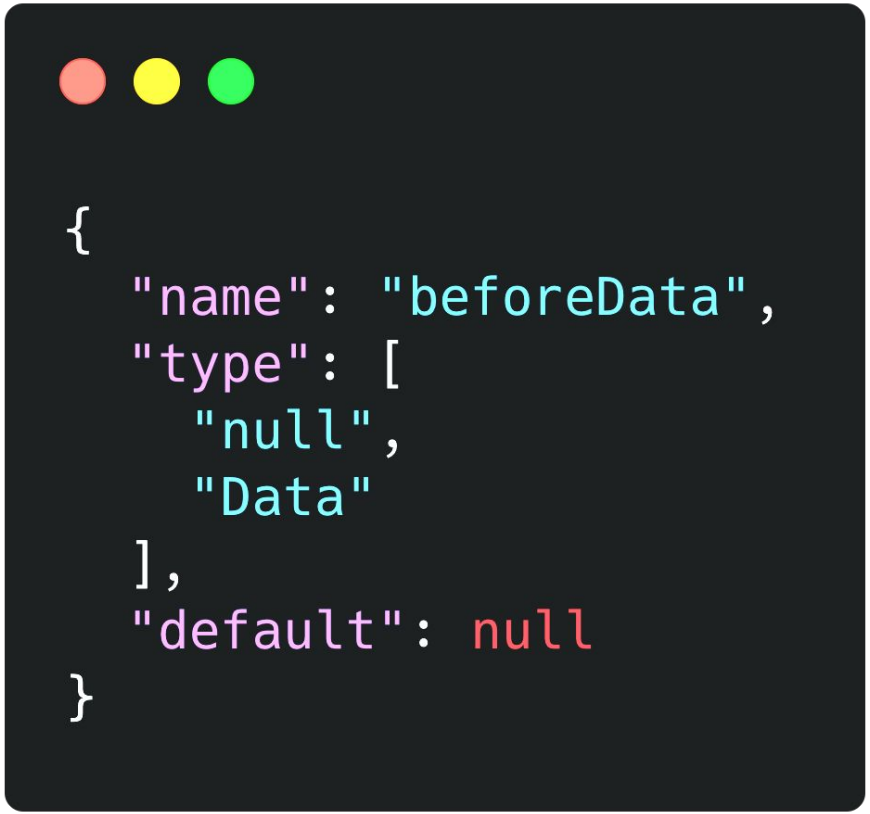
Show me some code (data)

```
{
  "name": "data",
  "type": {
    "type": "record",
    "name": "Data",
    "fields": [
      {"name": "stateCountyCode","type": ["null","int"],    "default": null},
      {"name": "state",          "type": ["null","string"], "default": null },
      {"name": "FIPS",           "type": ["null","string"], "default": null },
      {"name": "CountyName",     "type": ["null","string"], "default": null}
    ]
  }
}
```


@nzender

nathanzender.com

Show me some code (beforeData)



```
{
  "name": "beforeData",
  "type": [
    "null",
    "Data"
  ],
  "default": null
}
```

@nzender nathanzender.com

Show me some code (headers)

```
{
  "name": "headers",
  "type": {
    "type": "record",
    "name": "Headers",
    "fields": [
      {
        "name": "operation",
        "type": {"type": "enum","name": "operation","symbols": ["INSERT","UPDATE","DELETE","REFRESH"]}
      },
      {"name": "changeSequence",  "type": "string"},
      {"name": "timestamp",       "type": "string"},
      {"name": "streamPosition",  "type": "string"},
      {"name": "transactionId",   "type": "string"},
      {"name": "changeMask",      "type": ["null","bytes"], "default": null},
      {"name": "columnMask",      "type": ["null","bytes"], "default": null},
      {"name": "externalSchemaId","type": ["null","string"],"default": null}
    ]
  }
}
```

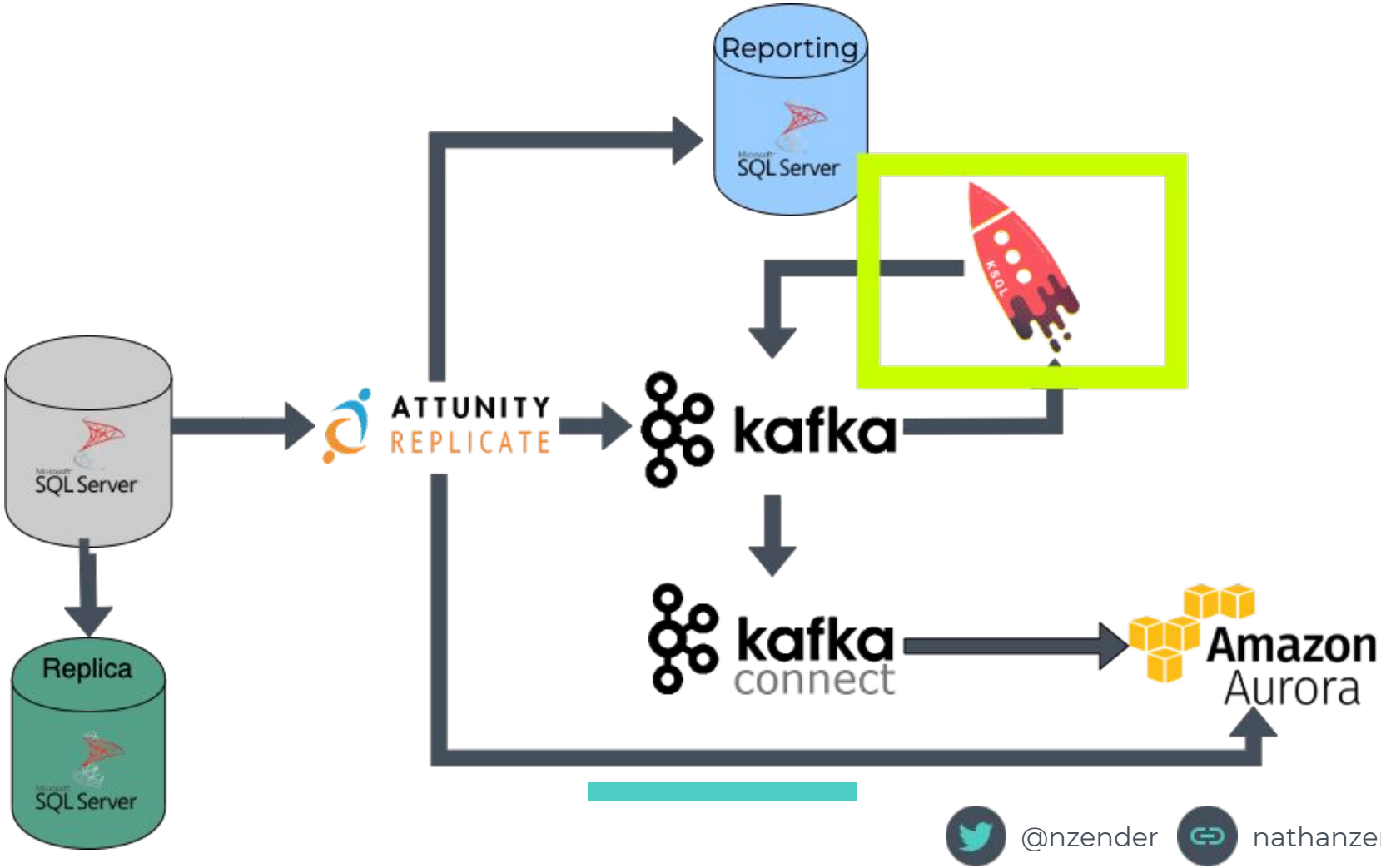

@nzender

nathanzender.com

KSQL


@nzender

nathanzender.com



@nzender nathanzender.com

What is it?

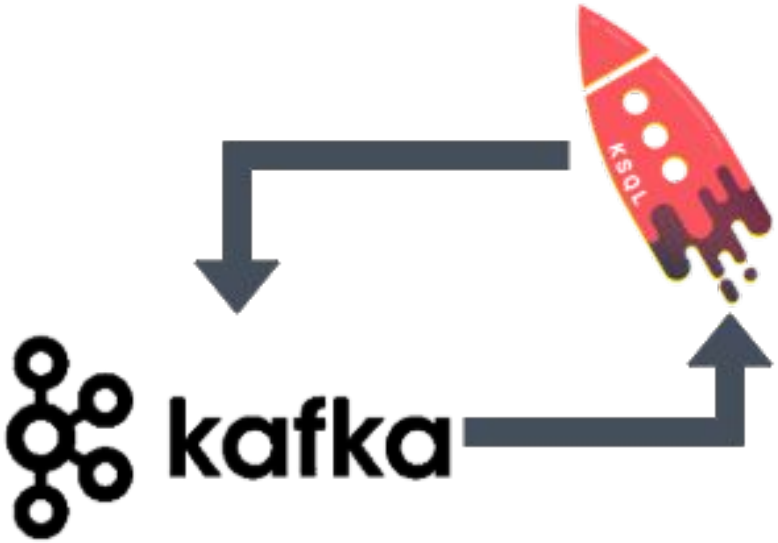
Confluent KSQL is the streaming SQL engine that enables real-time data processing against Kafka. It provides an easy-to-use, yet powerful interactive SQL interface for stream processing on Kafka, without the need to write code in a programming language such as Java or Python.

Why?

- Easy/familiar SQL interface to Kafka topics
- Initial thought was all business logic could live here easily since it was just SQL-like
- Allows adhoc & constantly running queries to be executed

How?

How?



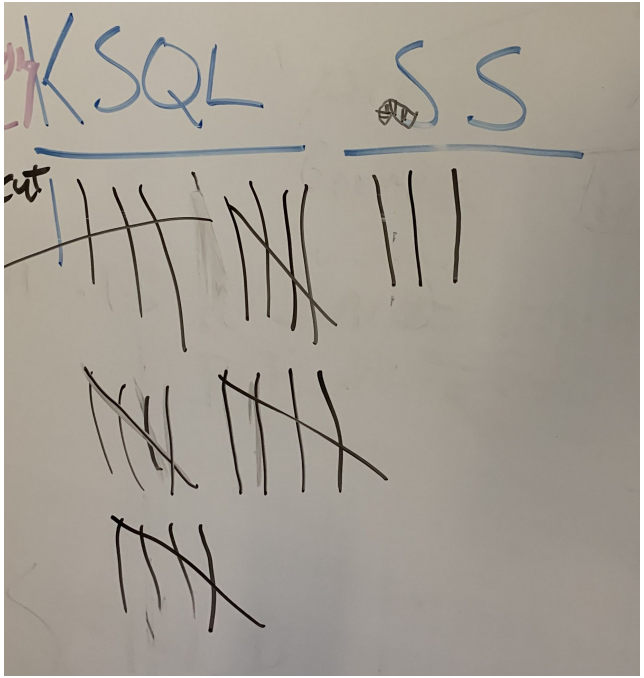
Show me some code

```
CREATE STREAM patient_termed_stream with
    (KAFKA_TOPIC='patient_event', PARTITIONS=3, REPLICAS=16)
AS
    SELECT
        DATA→PATIENTID AS `patientid`,
        'Patient Termed' AS `type`,
        cast(DATA→OUTCOMESTERMDATE as varchar(string)) AS `value`,
        HEADERS→TIMESTAMP AS `effective_timestamp`
    FROM DBO_PATIENT_STREAM
    WHERE ((BEFOREDATA IS NULL)
           OR (BEFOREDATA→OUTCOMESTERMDATE <> DATA→OUTCOMESTERMDATE))
        AND DATA→OUTCOMESTERMDATE IS NOT NULL
        AND DATA→OUTCOMESTERMDATE < '9999-12-31'
        AND HEADERS→OPERATION <> 'REFRESH';
```

But what's the catch??

- KSQL has some rough edges and you will get cut
- The deployment model is not great
  - We did roll our own Flyway adapter to simulate version controlled KSQL streams
  - Hopefully to be open sourced soon
- Adhoc queries can be dangerous 😬

KSQL vs Squid Squad (aka My Team)

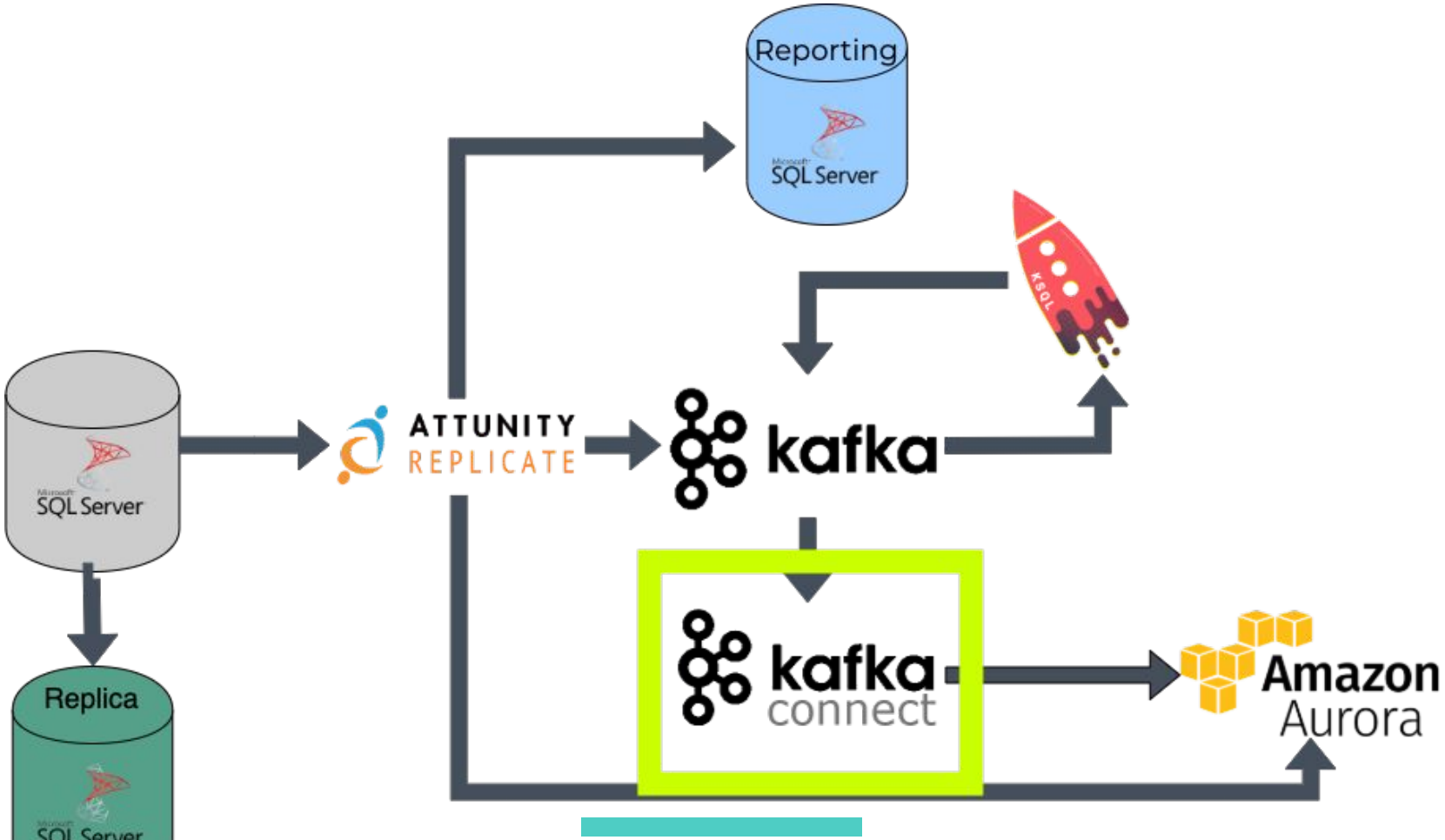


@nzender nathanzender.com

Kafka Connect


@nzender

nathanzender.com



@nzender nathanzender.com

What is it?

Kafka Connect, an open source component of Kafka, is a framework for connecting Kafka with external systems such as databases, key-value stores, search indexes, and file systems.

Why?

- De facto standard in getting data in/out of Kafka via configuration only
- Analytics reports could not be built by querying Kafka directly so we have to get it out somehow

How?

How?

Show me some code

```
{
  "name": "jdbc-sink-patient_event",
  "connector.class": "com.cardinalhealth.connect.PostgreSqlSinkConnector",
  "tasks.max": "1",
  "topics": "patient_event",
  "auto.create": "false",
  "insert.mode": "upsert",
  "pk.mode": "record_value",
  "pk.fields": "effective_timestamp,patientid,type",
  "key.converter.schema.registry.url" : "<CONNECT_KEY_CONVERTER_SCHEMA_REGISTRY_URL>",
  "consumer.security.protocol": "SSL",
  ...
}
```


@nzender

nathanzender.com

But what's the catch??

- Kafka Connect has been pretty solid
- Custom connector to fix with some timestamp conversion issues
- Warning!!!
  - **offset.retention.minutes** will probably bite you at some point

Show us some numbers

@nzender nathanzender.com

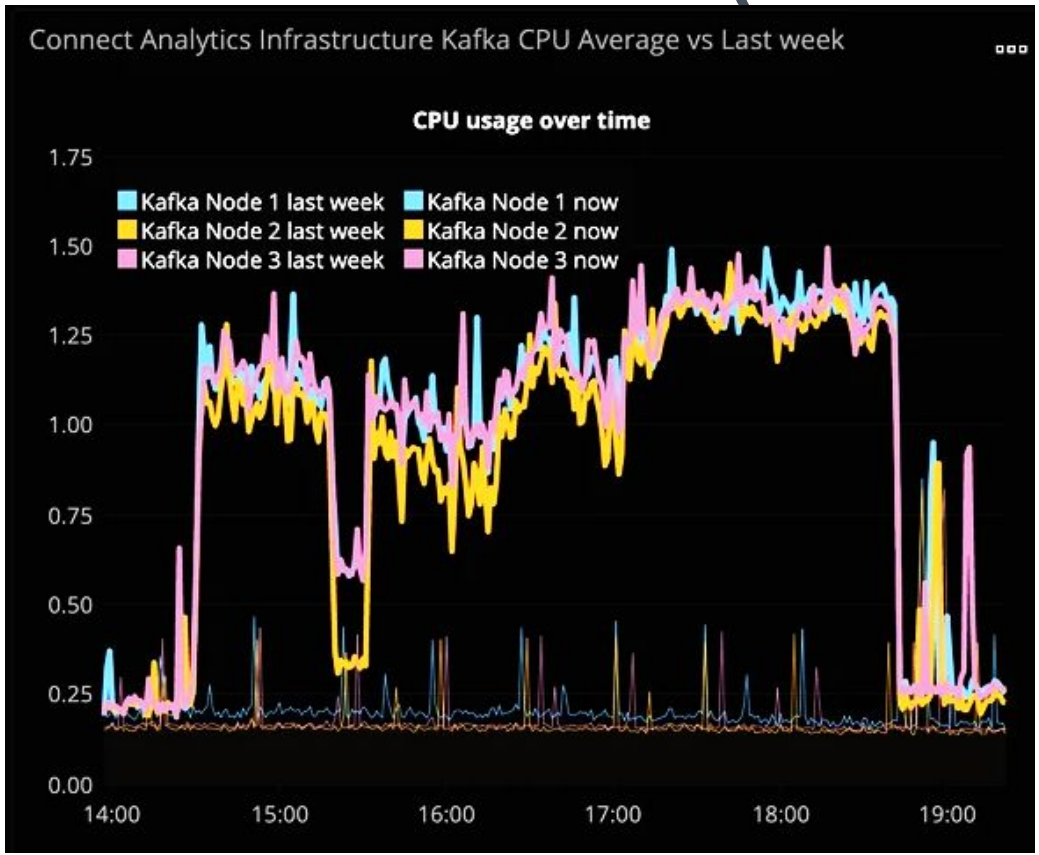
Infra Setup

- **Kafka cluster**
  - m5.large x 3
- **Attunity Replicate server**
  - m4.xlarge
- **KSQL Server**
  - PCF 6GB, 4CPU
- **Schema Registry**
  - m3.xlarge
- **Kafka Connect Server**
  - PCF 6GB, 4CPU
- **Aurora cluster**
  - db.r4.4xlarge x 2
  - 1 writer/1 reader

Real world first End 2 End

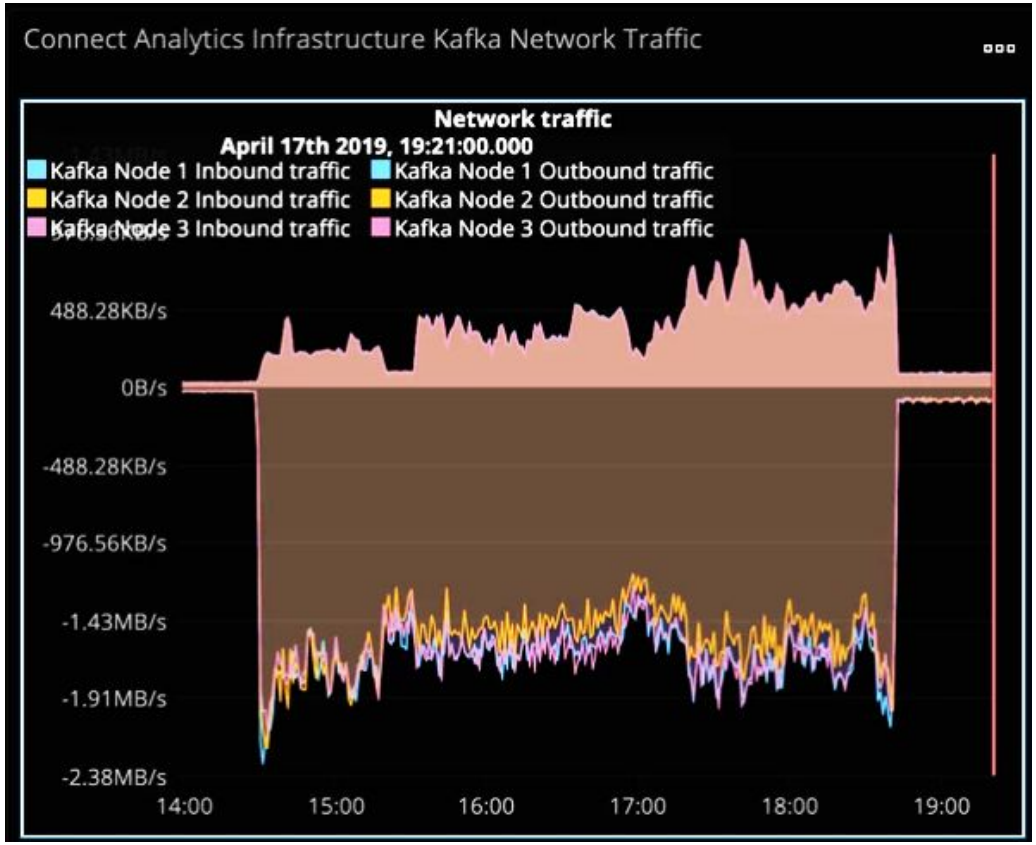
- ~ 400 tables watching for changes
- ~ 6 months worth of changes
- ~ 350 million Kafka messages in total processed
- Time to do **Kafka -> KSQL -> Kafka**
  - ~ 4 hours (25k messages per second)
- Time to do **Kafka -> Kafka Connect -> Aurora**
  - ~ 1 hour
  - ~ 30.5 million rows (filtered events)

Real world first E2E (Kafka while KSQLing)



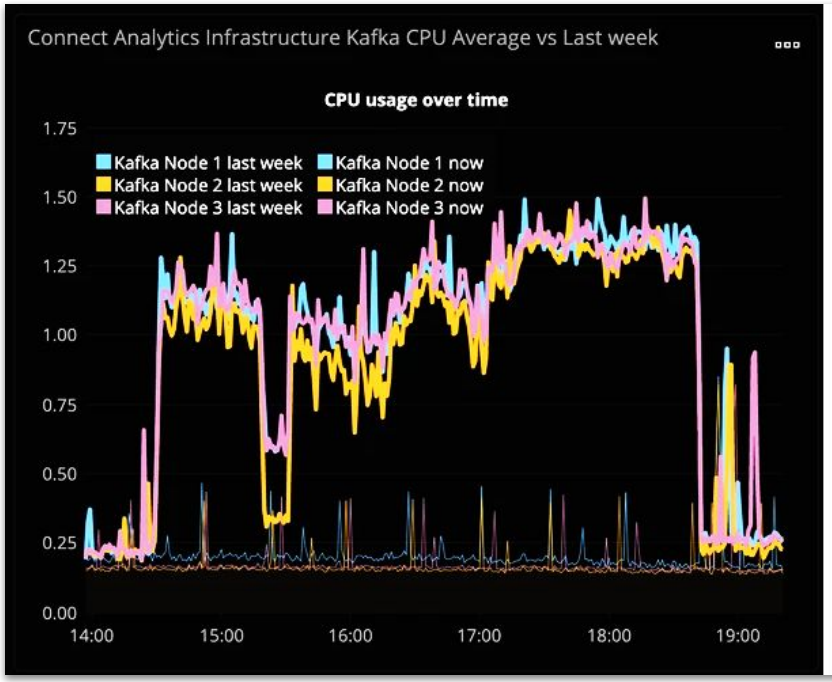
@nzender nathanzender.com

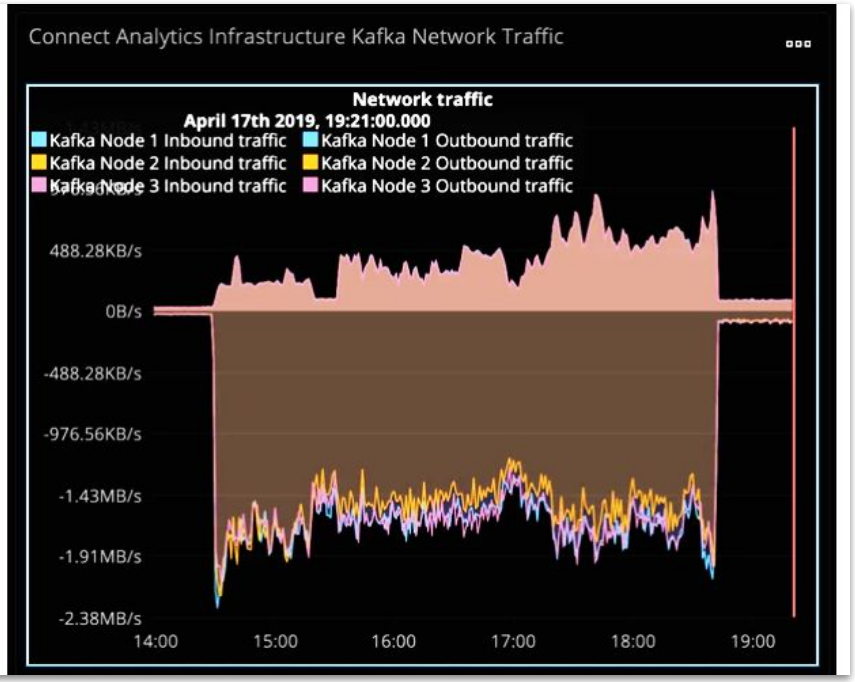
Real world first E2E (Kafka while KSQLing)



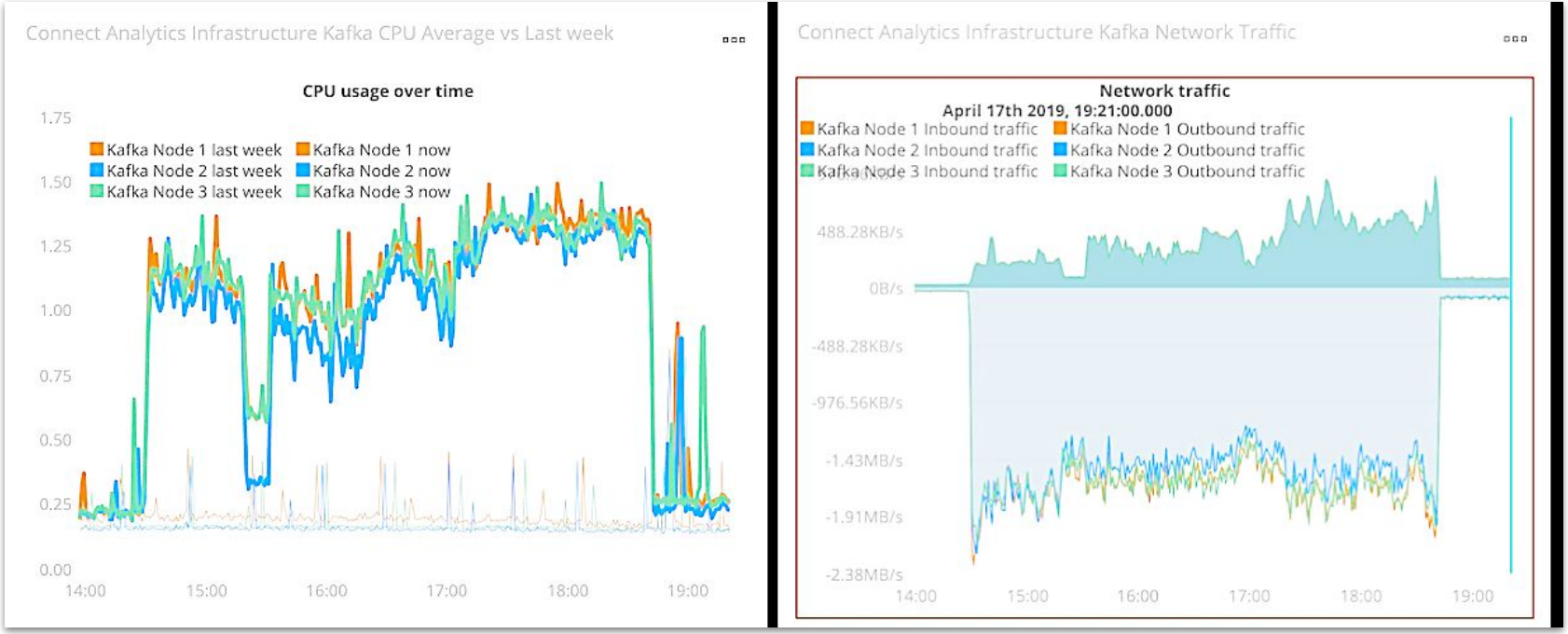
@nzender nathanzender.com

Real world first E2E (Kafka while KSQLing)



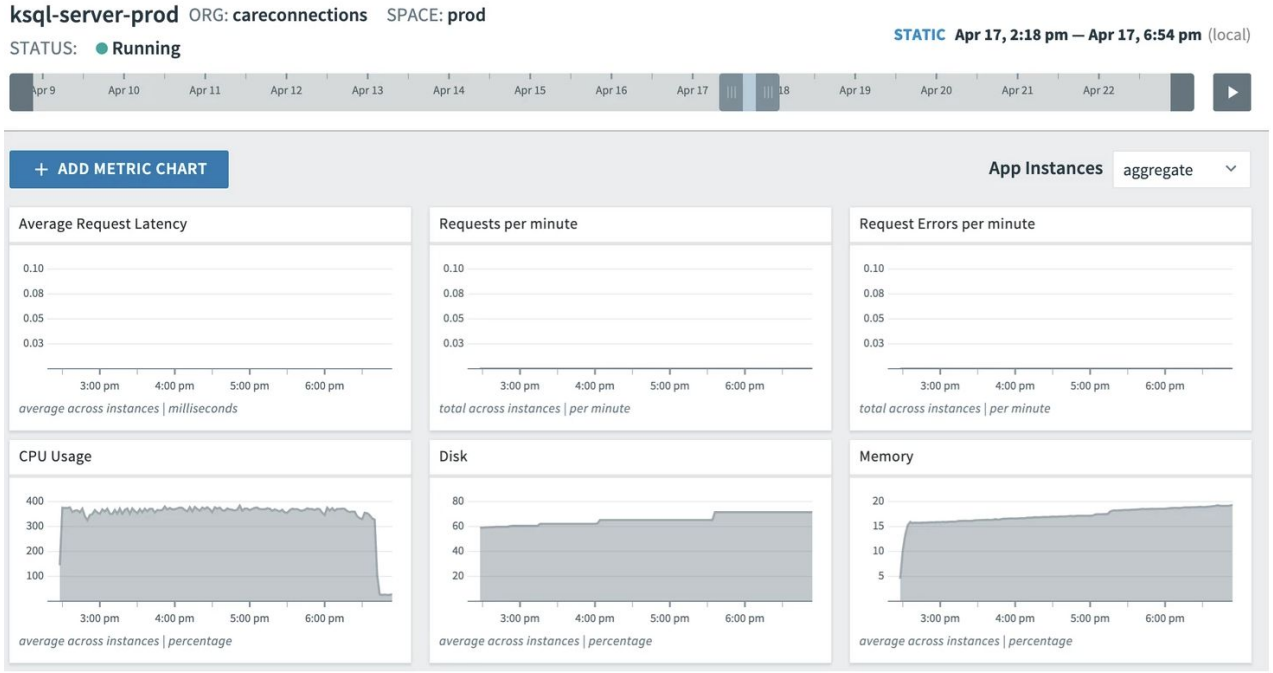


Real world first E2E (Kafka while KSQLing)



@nzender nathanzender.com

Real world first End 2 End (KSQL)

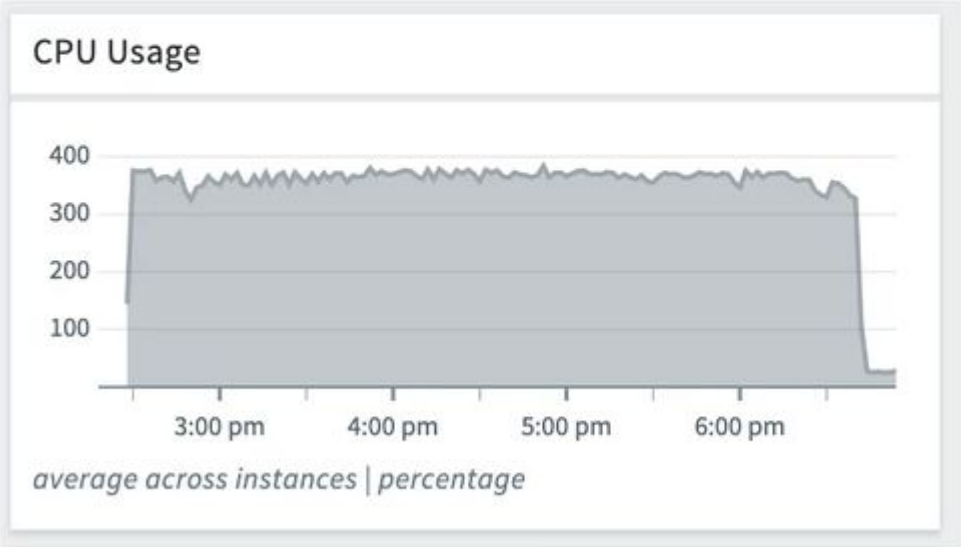


@nzender nathanzender.com

Real world first End 2 End (KSQL)

ksql-server-prod ORG: **careconnections** SPACE: **prod**

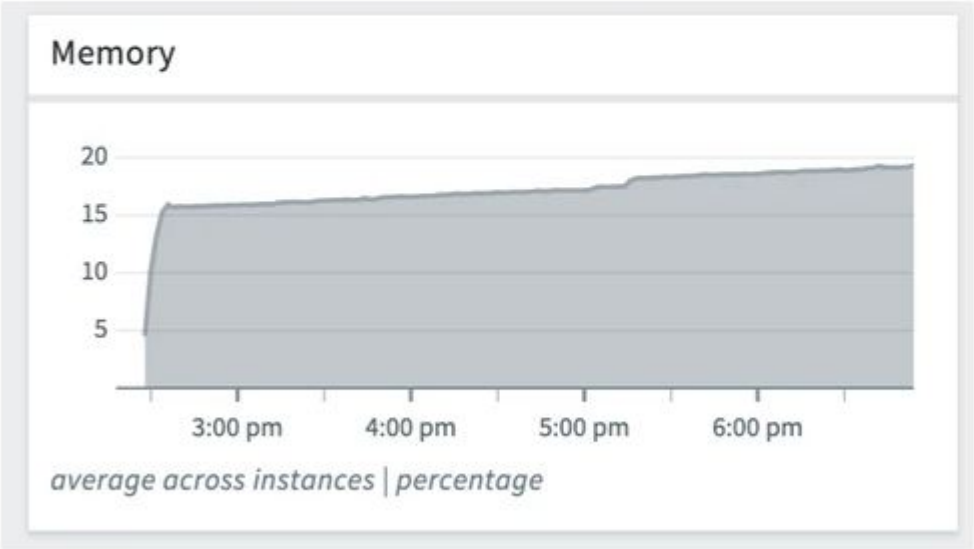
STATUS: ● **Running**



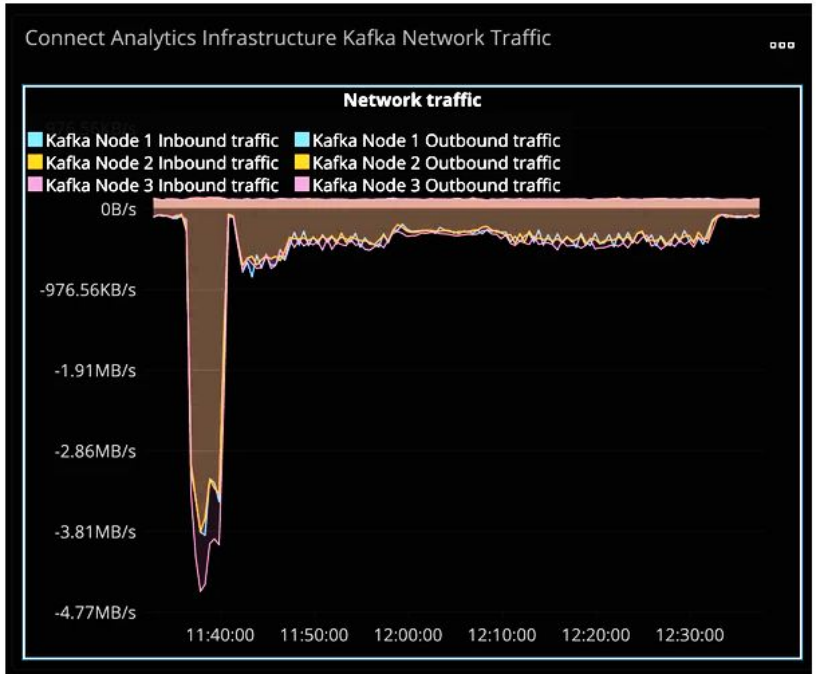
Real world first End 2 End (KSQL)

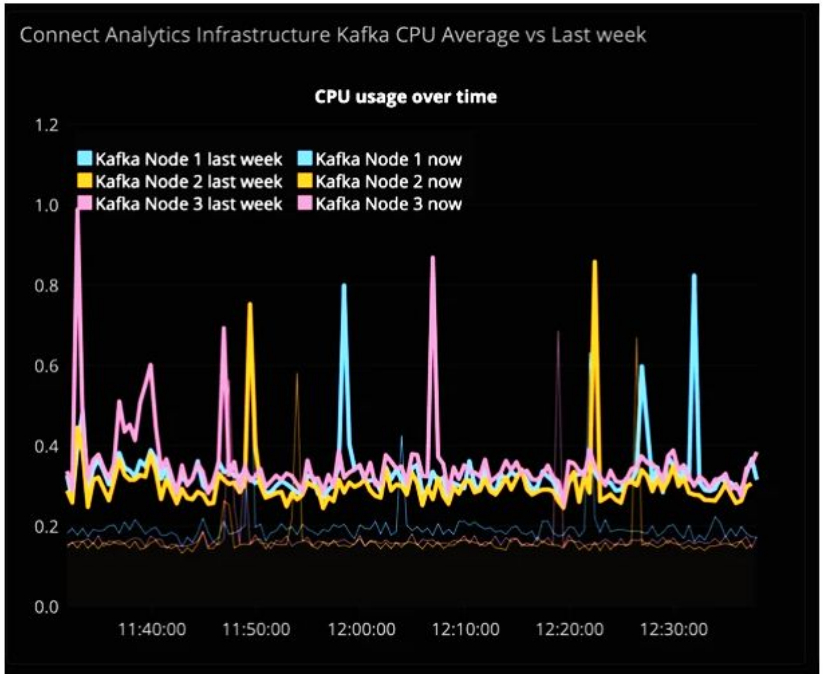
ksql-server-prod ORG: **careconnections** SPACE: **prod**

STATUS: **Running**



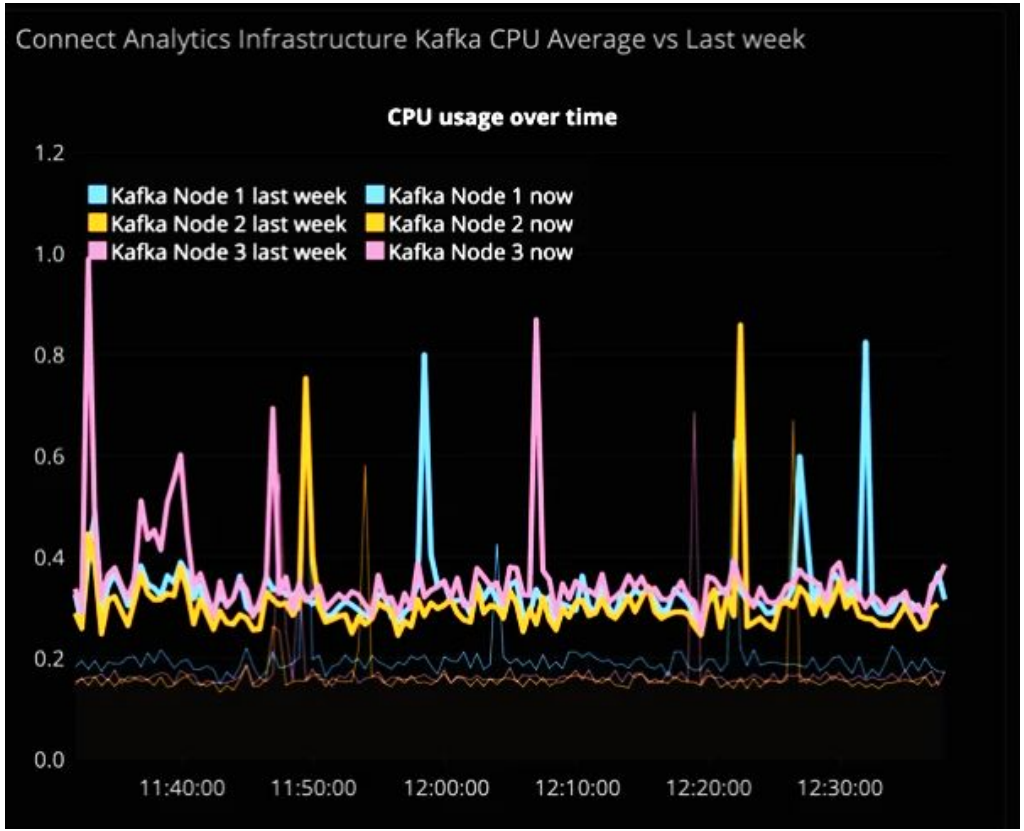
Real world first End 2 End (Kafka Connect)





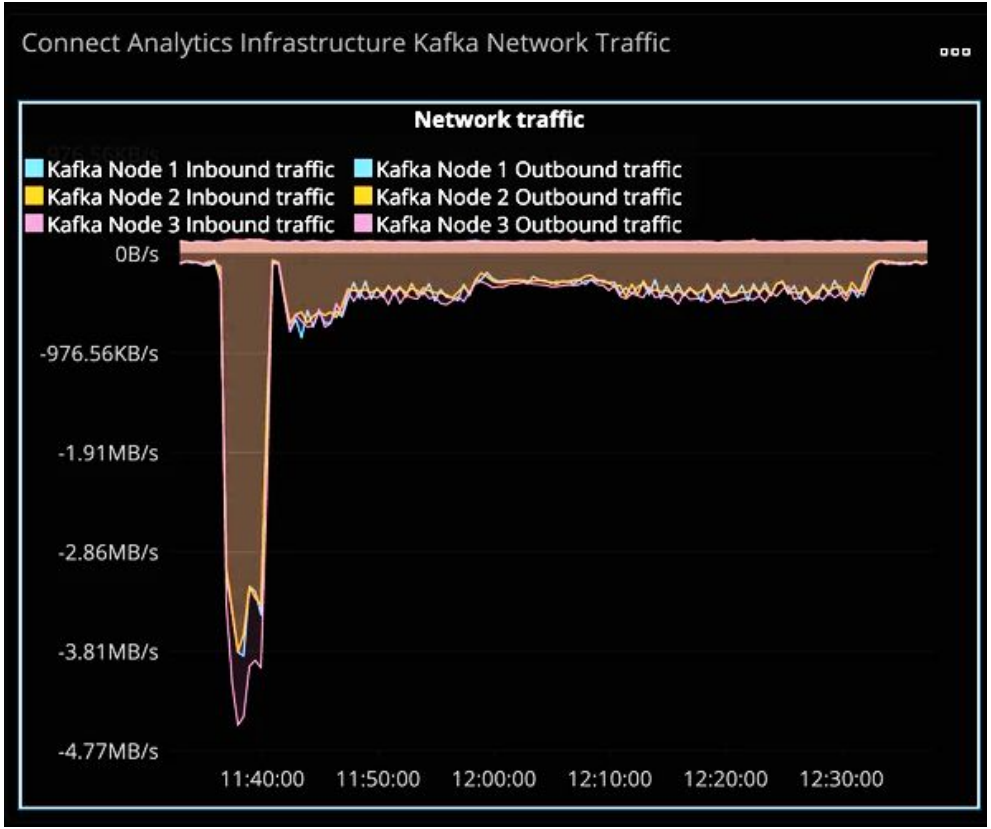
@nzender nathanzender.com

Real world first E2E (Kafka while Kafka Connect)



@nzender nathanzender.com

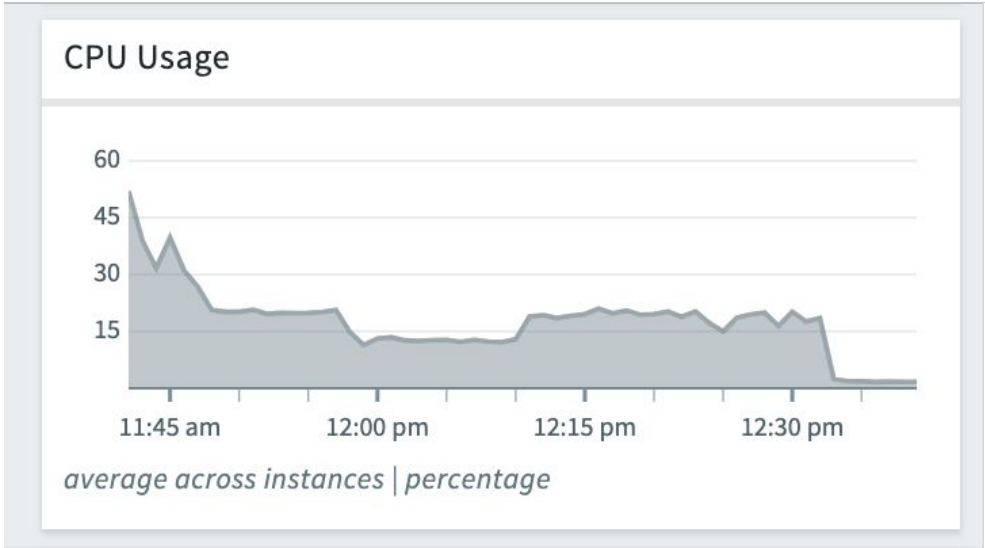
Real world first E2E (Kafka while Kafka Connect)



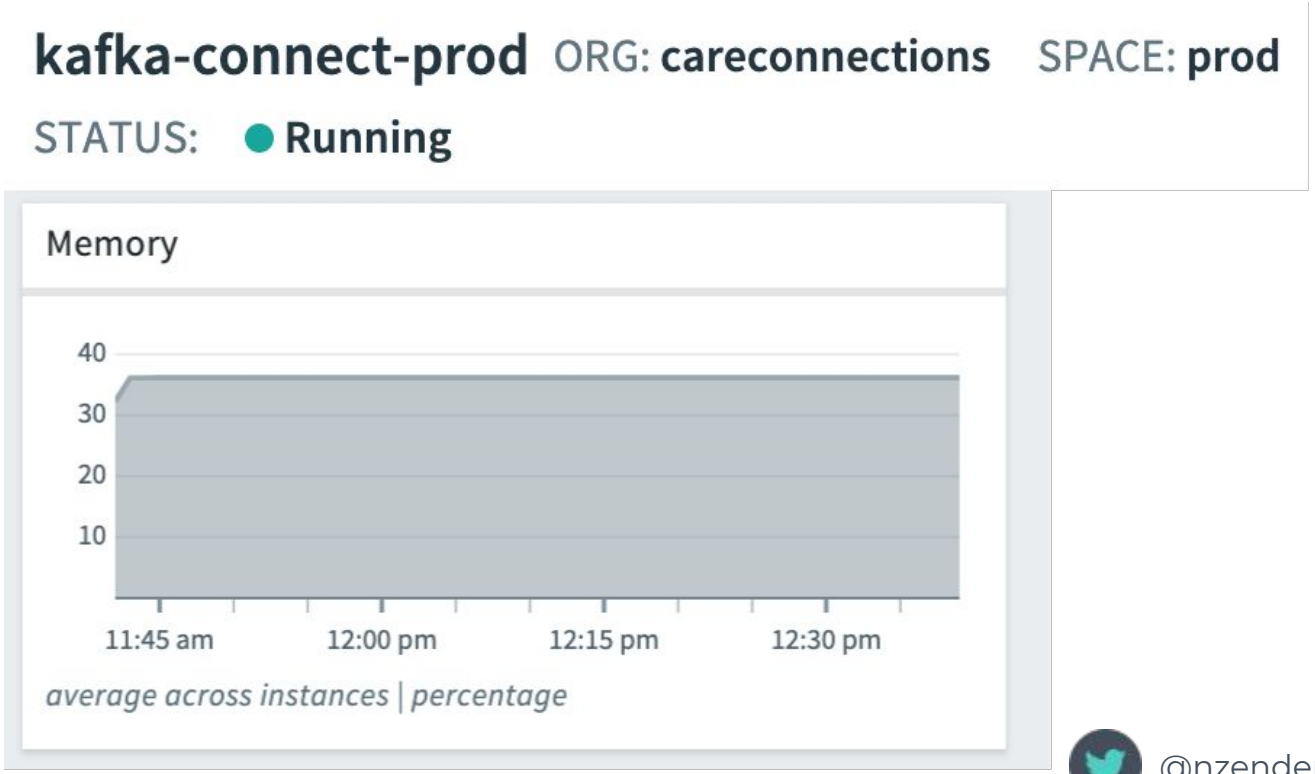
Real world first E2E (Kafka Connect)

kafka-connect-prod ORG: **careconnections** SPACE: **prod**

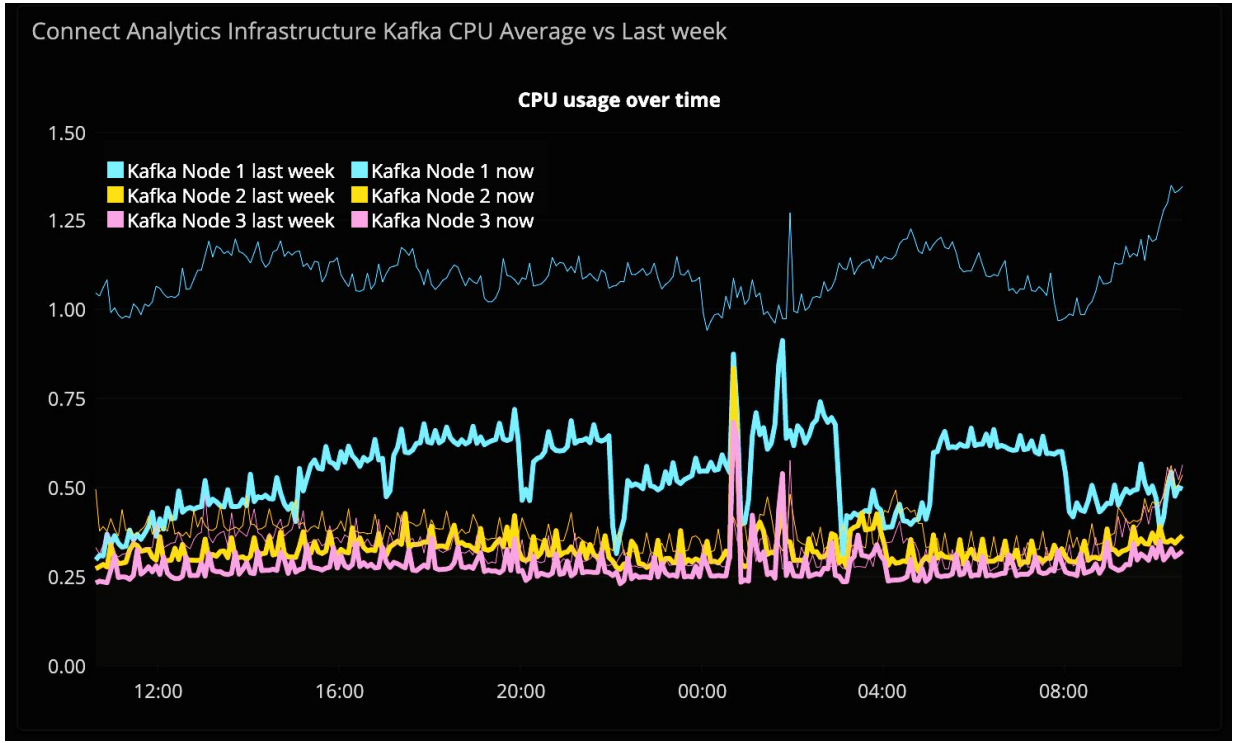
STATUS: ● **Running**



Real world first E2E (Kafka Connect)

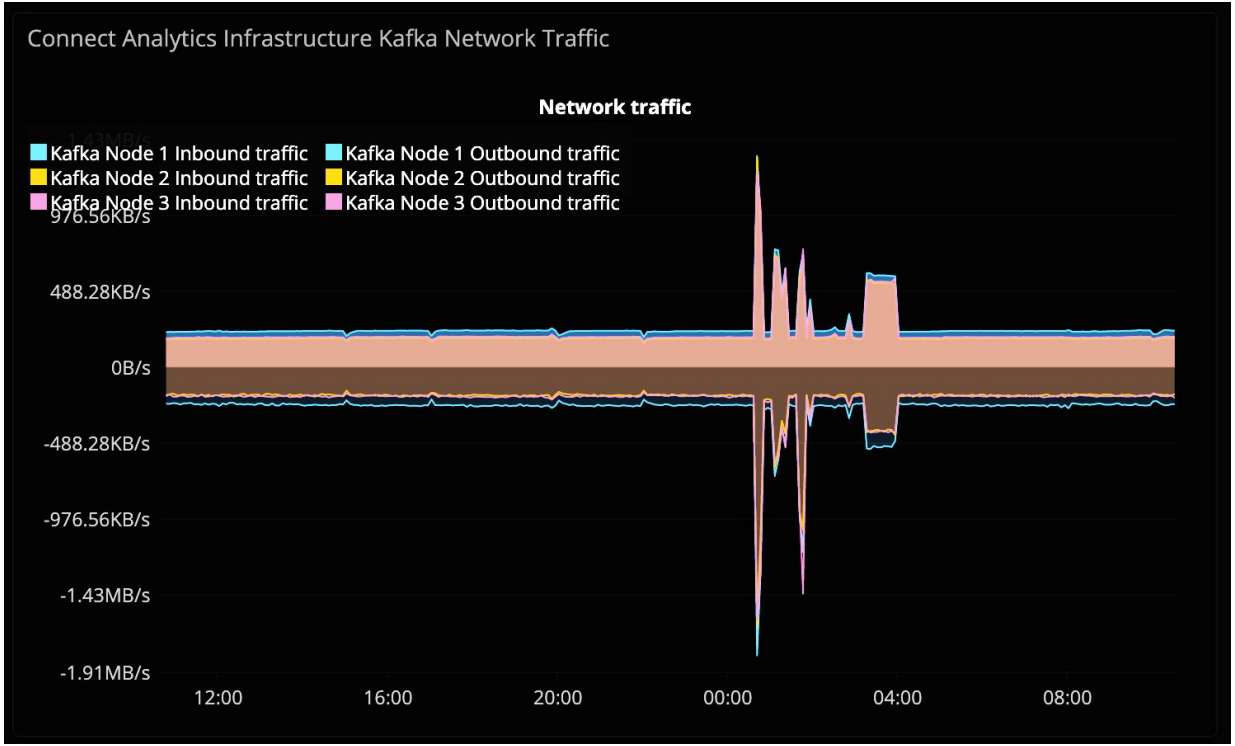


Real world Day to Day (Kafka)

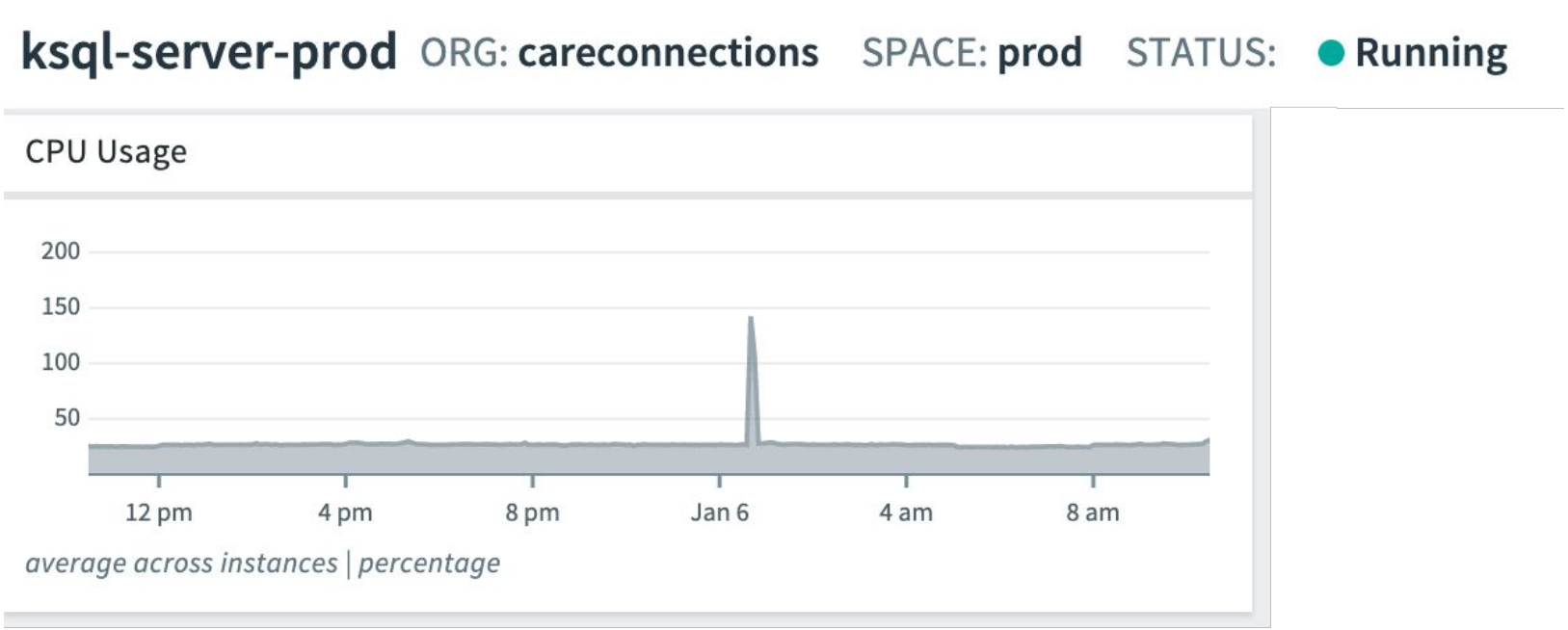


@nzender nathanzender.com

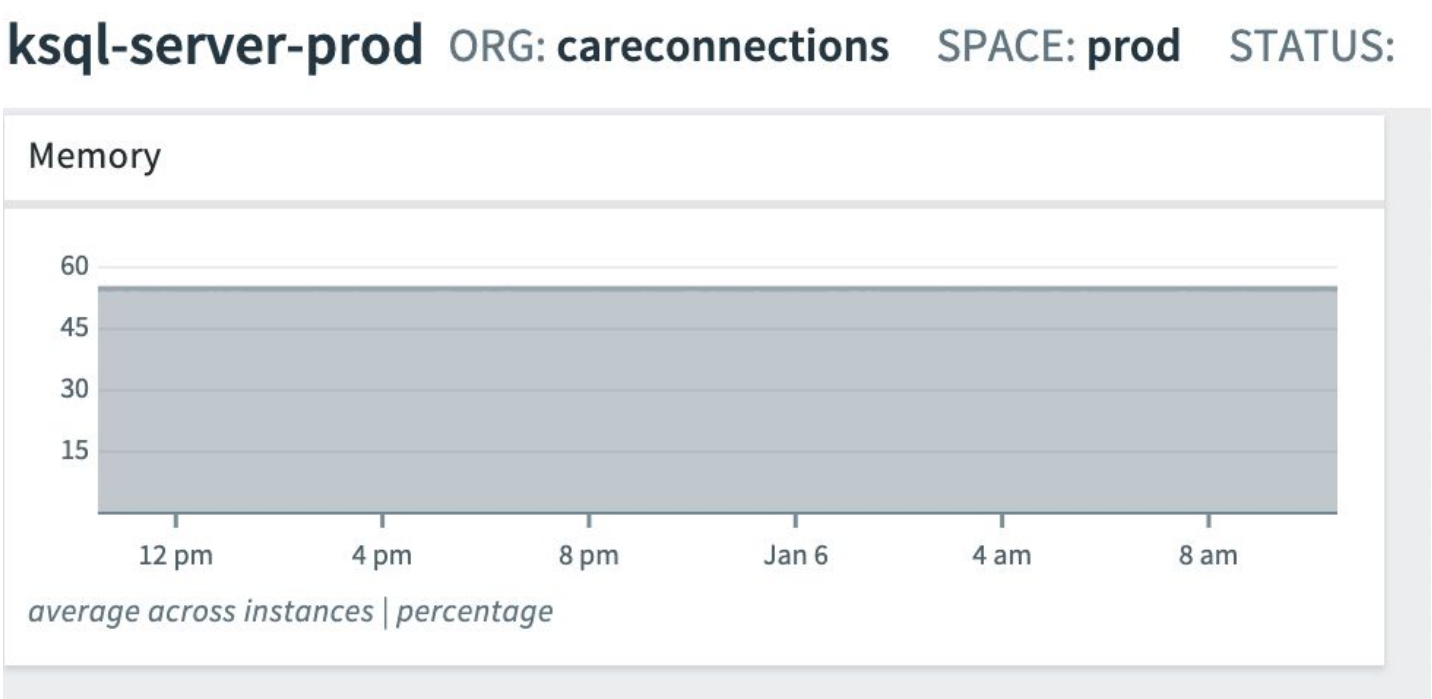
Real world Day to Day (Kafka)



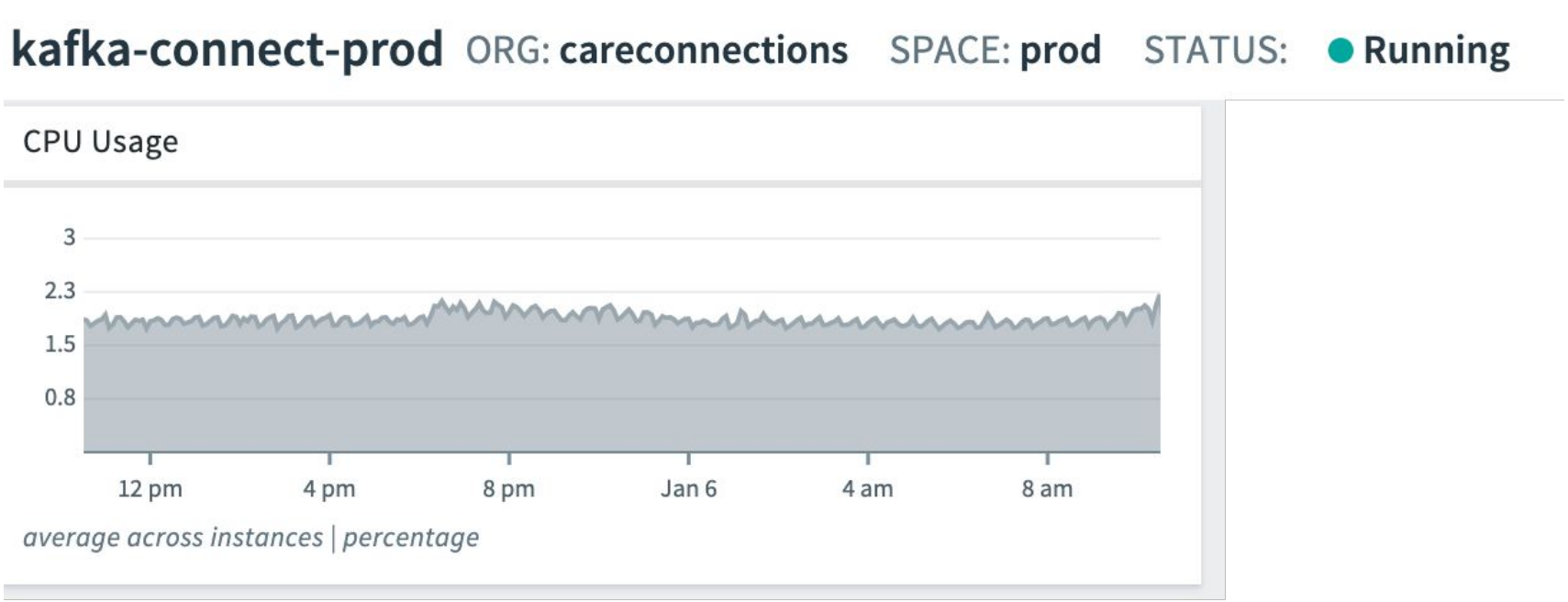
Real world Day to Day (KSQL)



Real world Day to Day (KSQL)



Real world Day to Day (Kafka Connect)



Real world Day to Day (Kafka Connect)

What have we accomplished?

@nzender nathanzender.com

Current State

- Delivered our first event based report
- Triaged issues that required an audit trail on tables that had no audit trail
- Break out one piece from the monolith
- **Did not bring down production**

What's Next for Us??


@nzender

nathanzender.com

The Future

- Continue to extract more and more analytics/insights from the events tracking
- Continue to use the event data to break the monolith
- Explore pushing our biz logic out of Aurora -> KSQL

Helpful Links

- Attunity Replicate
  - https://www.qlik.com/us/products/attunity-replicate
  - http://bit.ly/2PDmdoz - full support matrix
- Kafka
  - https://kafka.apache.org/quickstart
  - https://cwiki.apache.org/confluence/display/KAFKA/KIP-186%3A+Increase+offsets+retention+default+to+7+days
- KSQL
  - https://docs.confluent.io/current/ksql/docs/index.html
- Schema Registry
  - https://docs.confluent.io/current/schema-registry/index.html
- Kafka Connect
  - https://docs.confluent.io/current/connect/index.html
- Debezium
  - http://bit.ly/2tpFHEF
  - https://debezium.io/
- Confluent on CDC and Kafka
  - https://www.confluent.io/blog/no-more-silos-how-to-integrate-your-databases-with-apache-kafka-and-cdc/

@nzender nathanzender.com

Q&A


@nzender

nathanzender.com