Palo Alto Networks' massive-scale deployment

By Talat Uyarer

BEAM SUMMIT Austin, 2022

Who am I

- Living in San Francisco Bay Area since 2015
- Sr Principal Engineer at Palo Alto Networks (Cortex Data Lake Team)
- Software developer for 10+ years
- Proud Member of Apache Software Foundation
- Passionate about open-source big data projects
- Apache Beam user since early 2019

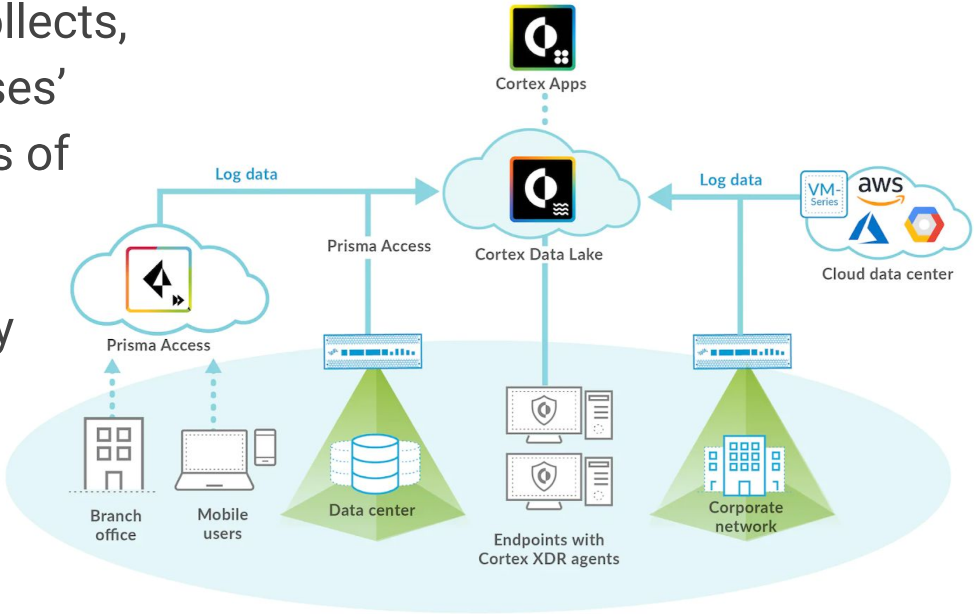
What is Cortex Data Lake

What is Cortex Data Lake

Cortex™ Data Lake infrastructure collects, integrates, and normalizes enterprises' security data combined with trillions of multi-source artifacts.

Most of Palo Alto Networks Security Products are powered by data lake.

That means there is so many consumer for the Data Lake's data



Some Numbers About CDL

- Our data lake is deployed at more than **10 geographical** regions and is highly scalable.
- One of our locations receives more than **10 million records** per second and can be scaled to receive more than **100 million records** per second.
- We store close to **100 Petabytes** at any time and can store much more.
- We serve more than **10 different applications** with **10 thousands** streaming jobs

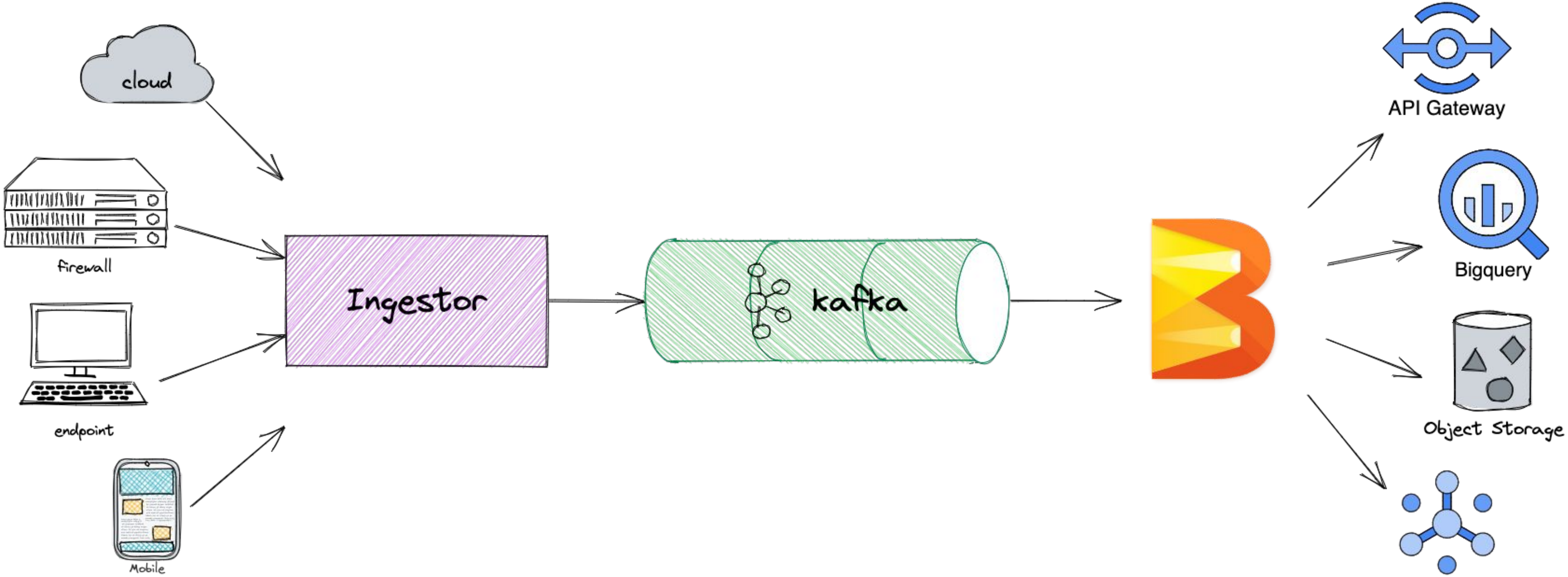
Our Current Use Cases

You can find most common use cases that we server internal teams.

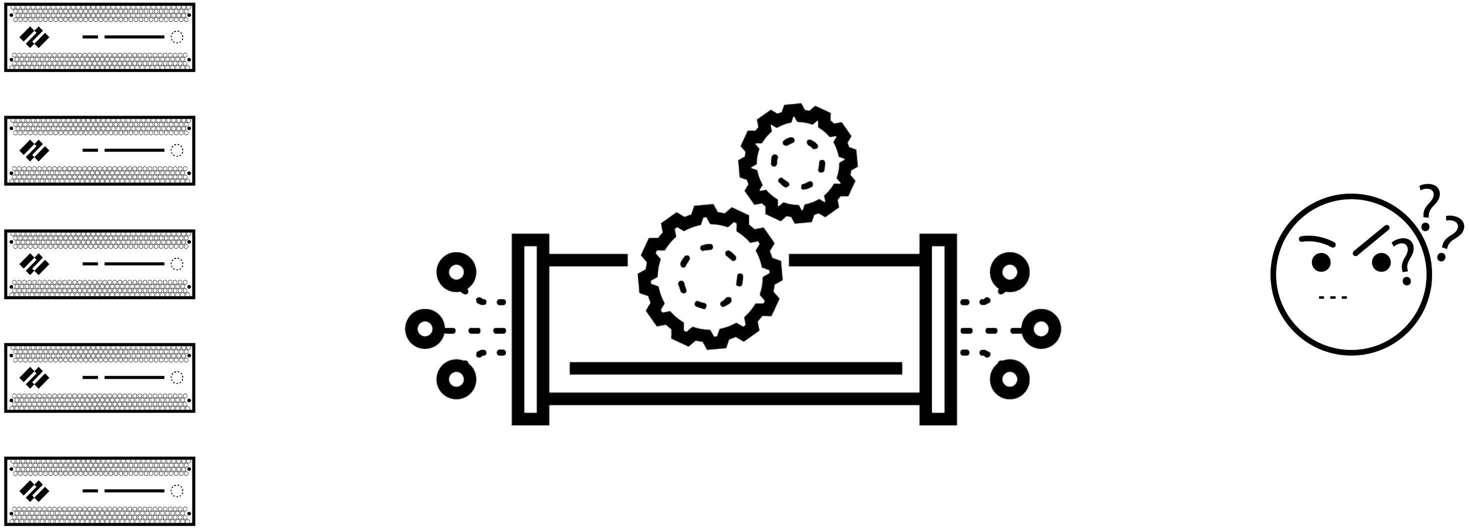
- Data Transformation and Enrichment
- Data Format Change
- Data Aggregation
- Real time Analytics
- Real Time Data Analysis for threat prevention and alarm creation
- Machine Learning

Let's Design a Datalake

In General Datalake Archtecture

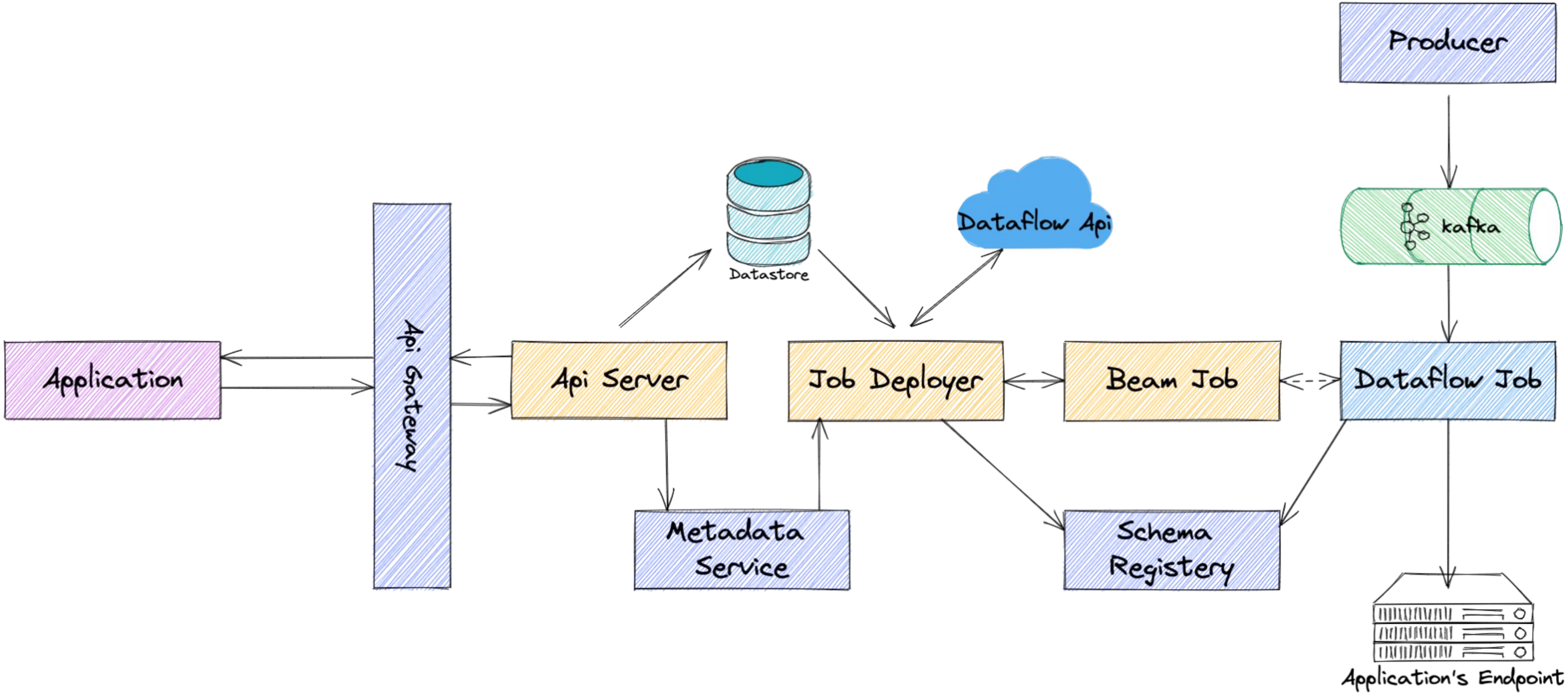


How would like to process data ?



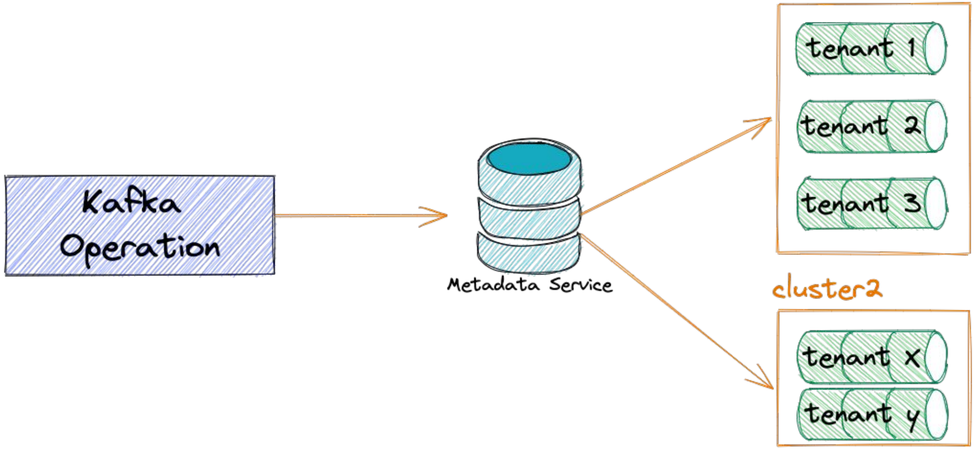
Streaming Infrastructure

Our Streaming Infrastructure

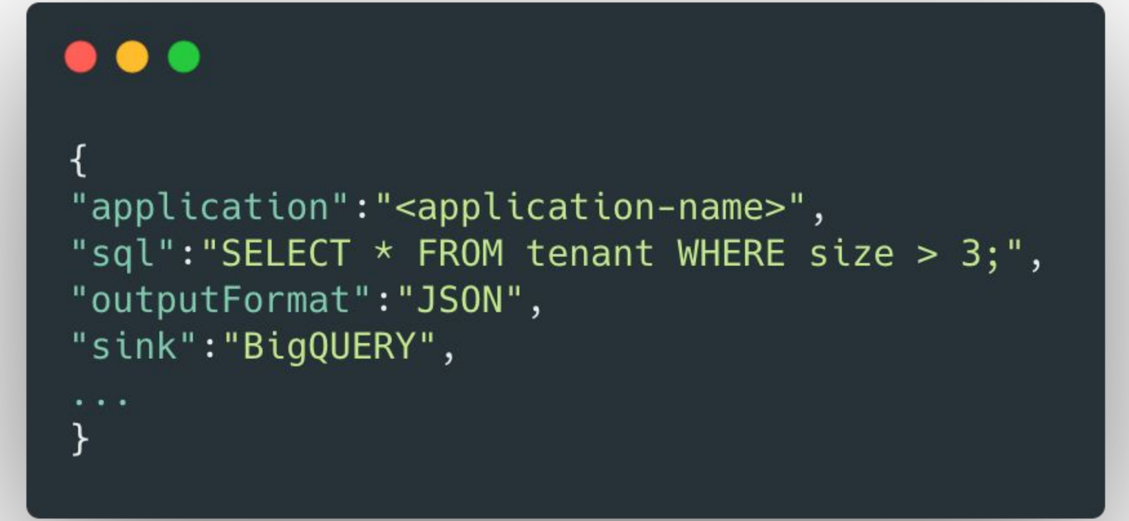


Data in Kafka

- We have multiple kafka clusters
- Topic per tenant
- Any Kafka operation such as topic creation, partition increase etc. has to go thru Metadata Service.
- Metadata decide cluster for topic creation
- Metadata Service source of truth fro Kafka



Subscription Model

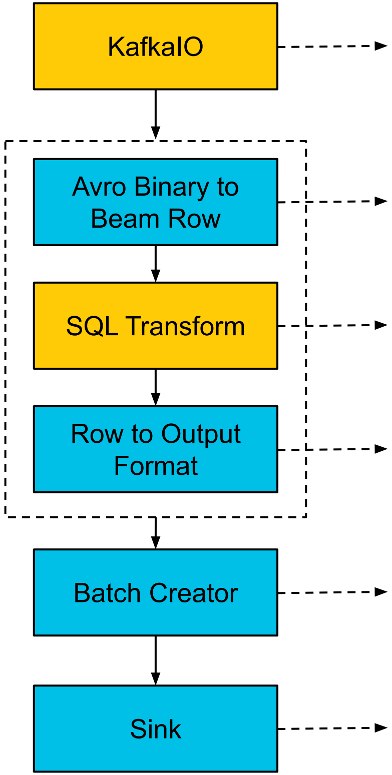


```
{
"application":"<application-name>",
"sql":"SELECT * FROM tenant WHERE size > 3;",
"outputFormat":"JSON",
"sink":"BigQUERY",
...
}
```

Streaming Service Features

- We use Beam with Dataflow Runner
- Streaming Generates DAG based on Application’s Rest Payload
- Several different output format such as Avro, Json CSV… etc
- Variety of Sink type such as Https, Grpc, Syslog, Storage, SQL... etc
- CRUD operation for any subscriptions
- Self Healing for Kafka Changes
- Cost Optimizer for Our Jobs aka Cold Starter

How Generated DAG Looks Like



Reading from Kafka. This is Beam SDk’s KafkaIO

So far all logs are bytearray of Avro. To apply application’s filter we need to create Beam Row and set Schema

Beam generates necessary Transformation for SQL query

Output of SQL query is ROW. We convert Application’s output format after we receive results.

This step is responsible for create chuck of logs based on Application’s definition.

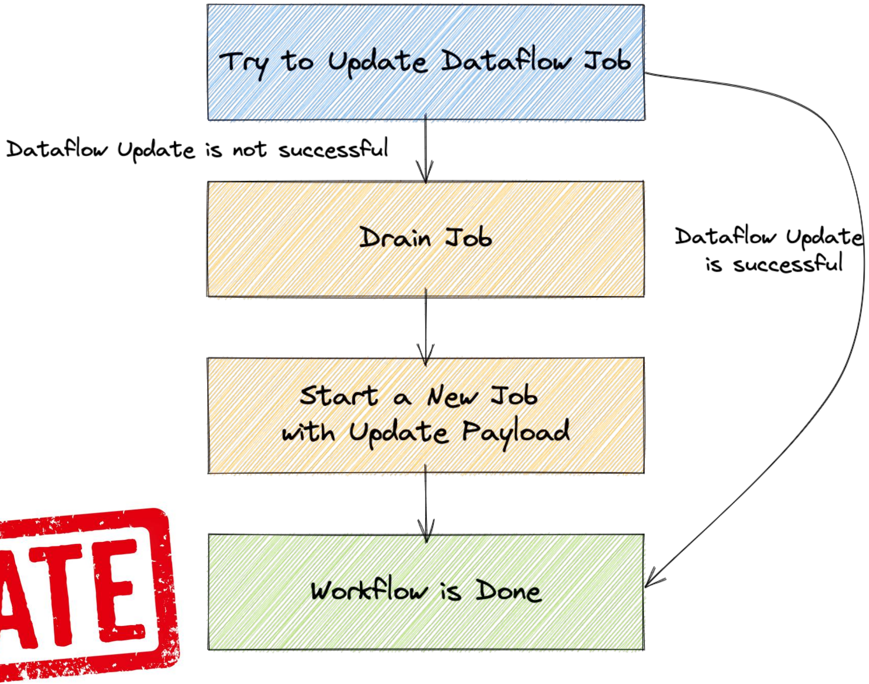
Sink responsable to write Application Endpoint.

Operational Challenges

Reliable Update

- Dataflow Update does not allow a replacement job if DAG or Input Split count changes
- Example for **Input Split count change**: Kafka partition count is changed, We can not updated
- Example for **DAG change**: User may change their SQL query from filtering to aggregation.

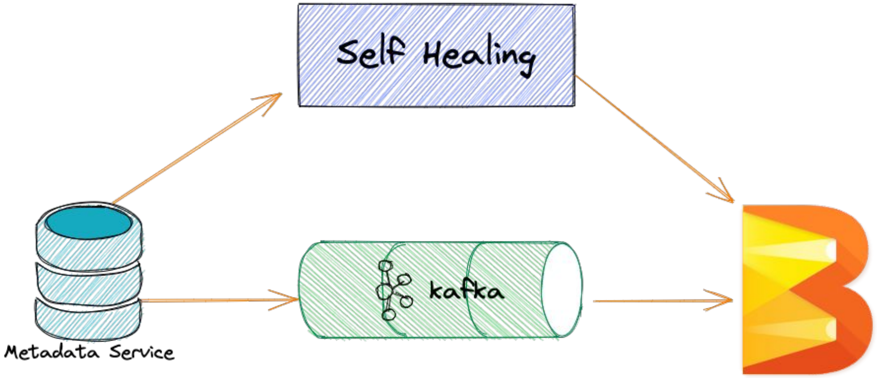
UPDATE


Try to Update Dataflow Job
Dataflow Update is not successful
Drain Job
Start a New Job with Update Payload
Workflow is Done
Dataflow Update is successful

Kafka Topic Evolution

Kafka Topic Evolution

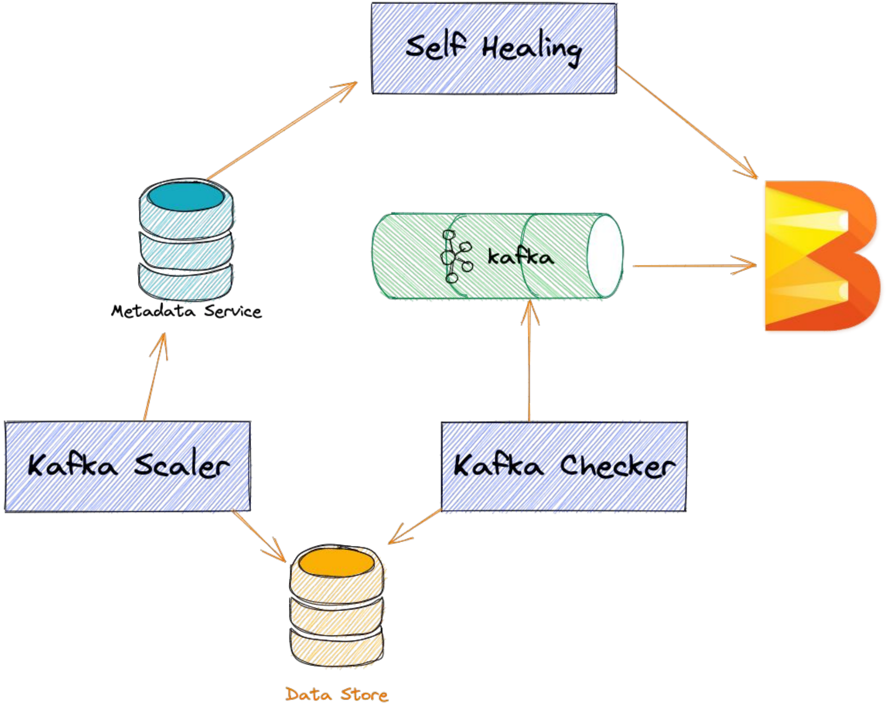
Self Healing



- Application submit their job by using our endpoint
- They don't have any invisibility to modify jobs for any related infrastructure changes such as Kafka cluster migration, partition count increase
- If there is issue that affects our jobs we need to heal them

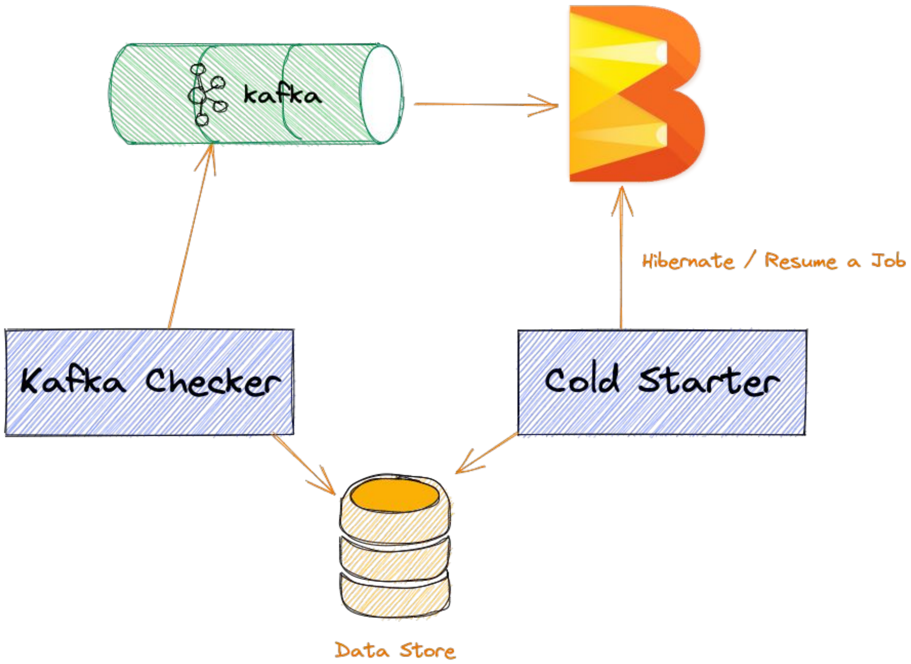
Kafka Topic Evolution

Partition Scaler



- Kafka Checker checks all kafka cluster every minute interval
- Based on calculated traffic per topic Kafka Partition Increaser increase partitions
- Whenever self healing see changes on Kafka topic it updates all related streaming jobs by reliable update
- Decrease is actually migrate topic. We create new topic with less partition and consume old and new topic for a while

Cold Starter



- Dataflow has auto scaling feature which helps us a lot for idle resources
- However auto scaling is only scale down until 1 Worker
- Sometimes we need less than one worker for jobs
- Based on Kafka Topic traffic it decide hibernation or resume for consuming jobs.

Questions?

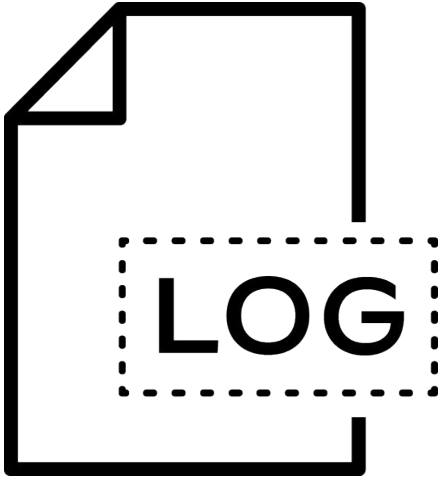
Send me an Email
talat@apache.org

Find me on Twitter
@talatuyarer

BEAM SUMMIT Austin, 2022

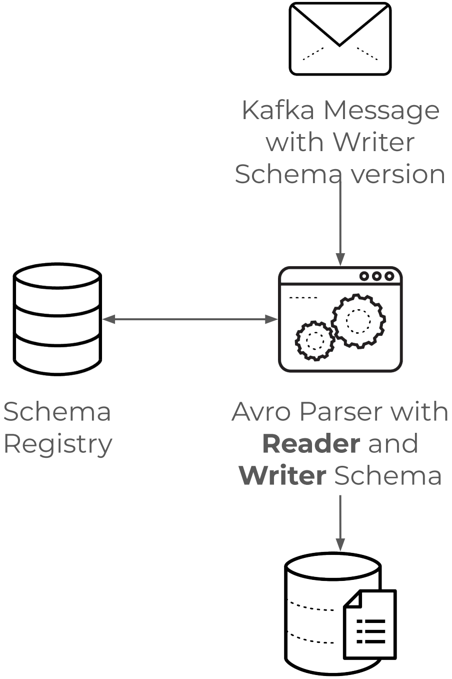
Extras

Schema Evolution



- Beam SQL does not support schema changes
- This is painful if you have Select * style jobs.
- Currently only way is re-submitting stream jobs to re-generate their Beam SQL Java code with new schemas
- Luckily all events are written as Avro binary. Avro support some kind of schema evolution.

How we handle Schema Evolution



Kafka Message with Writer Schema version

Schema Registry

Avro Parser with **Reader** and **Writer** Schema

Avro Generic Record

- Each job has their submitted version of Avro schema. We call Reader schema.
- Each Kafka message has Writer schema version as metadata on Kafka header.
- We convert all version of Avro events to Job's version of Avro Generic Record and convert it to Row.
- Our schema updater check all jobs' SQL queries if their sql has relevant fields with changes we update Job to update Beam SQL's Java generated Code otherwise We don't restart the job

Few Other Auxiliary Services

- Rolling Update for code changes
- Audit Services