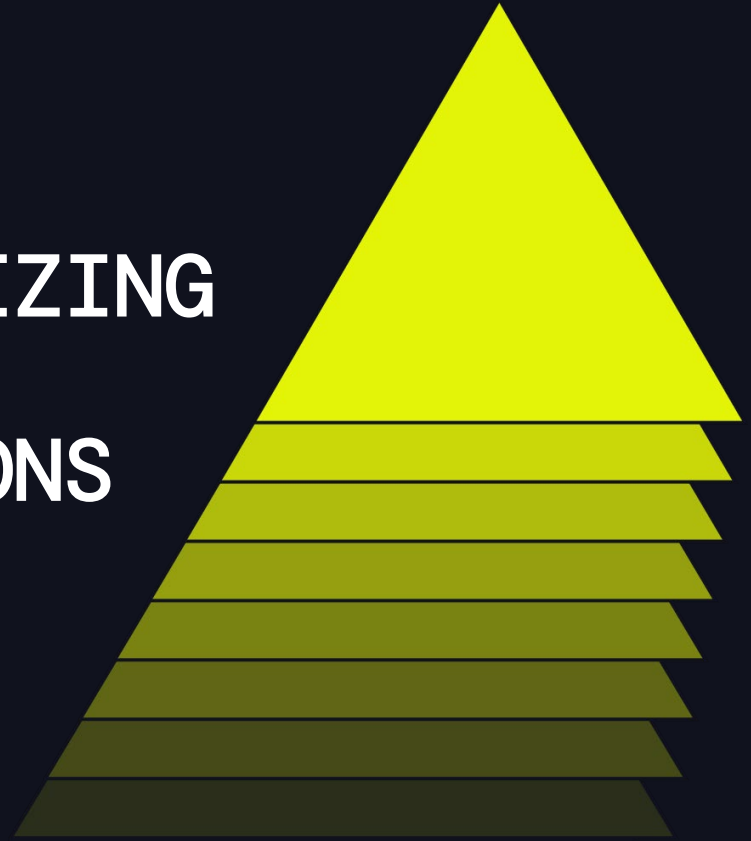


MIGRATING AND OPTIMIZING LARGE-SCALE STREAMING APPLICATIONS WITH DATABRICKS



Sharif Doghmi, Donghui Li
June 2024

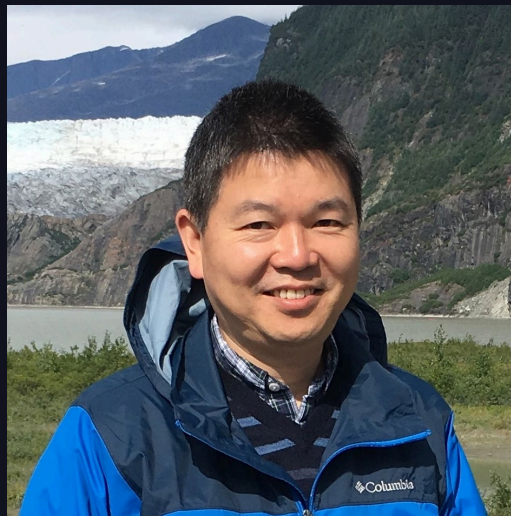
WHO WE ARE



Sharif Doghmi

Lead Software Engineer, FreeWheel

[linkedin.com/in/sharifdoghmi](https://www.linkedin.com/in/sharifdoghmi)



Donghui Li

Lead Software Engineer, FreeWheel

[linkedin.com/in/donghui](https://www.linkedin.com/in/donghui)

SESSION OVERVIEW

- FreeWheel, Beeswax, and Programmatic Advertising
- Beeswax Data Platform
- Redesign and Migration of Streaming Application
- Monitoring and Observability
- Performance Optimization
- Databricks Development Experience
- Future Databricks Work

FREEWHEEL, BEESWAX, AND PROGRAMMATIC ADVERTISING

FREOWHEEL

A COMCAST COMPANY

FREEWHEEL

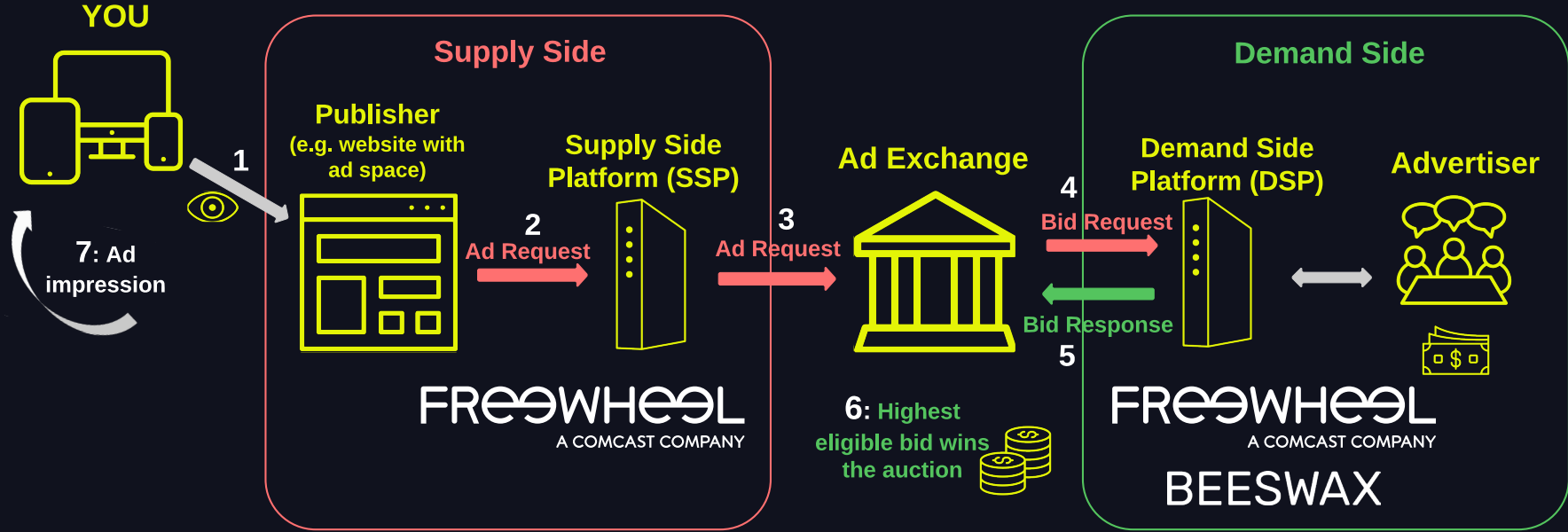
A COMCAST COMPANY

- Comprehensive ad platforms for publishers, advertisers, and media buyers
- We make it easier for buyers and sellers to transact
- Products for demand and supply sides of advertising
- We serve many of the largest media companies in the U.S. and Europe
- We power advertising for many of the biggest sporting events

BEE SWAX

PROGRAMMATIC ADVERTISING 101

Real Time Bidding



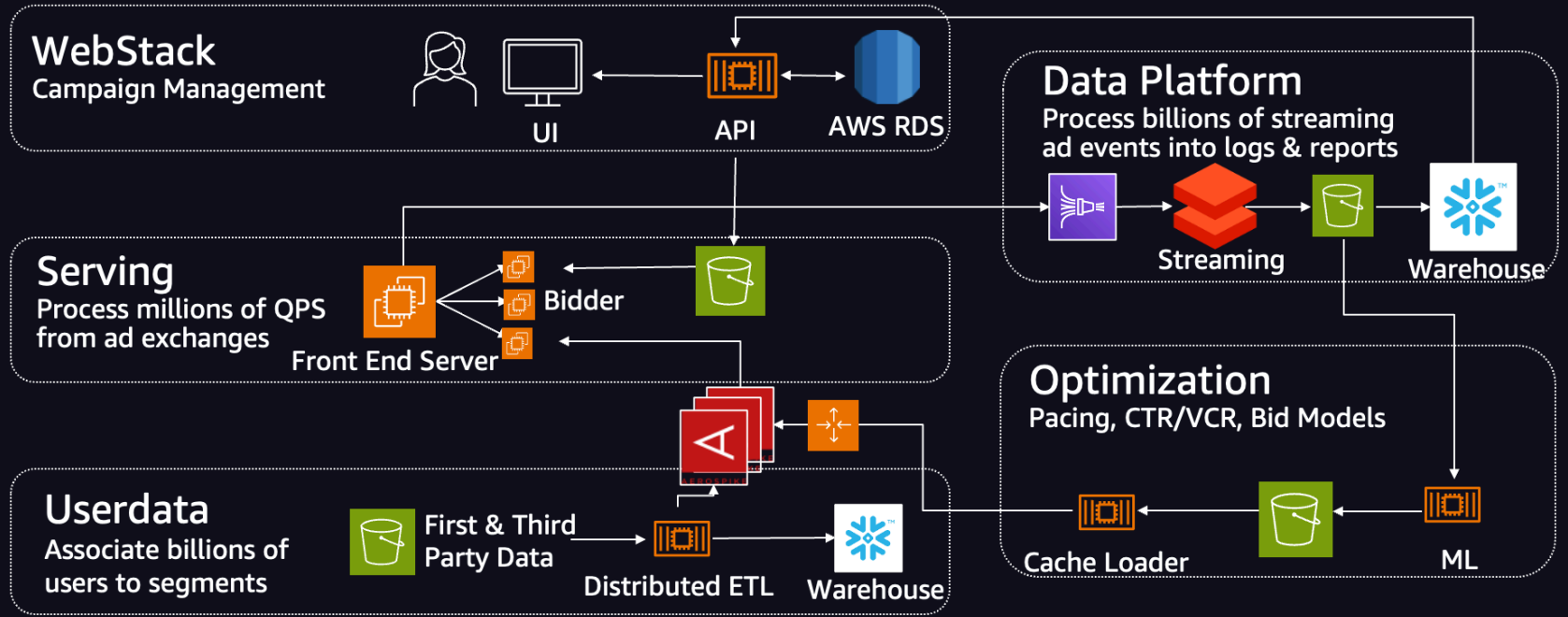
BEESWAX

FreeWheel's Programmatic Buying Platform

- Enables demand-side customers to:
 - Plan and execute ad campaigns
 - Bid on ad inventory using RTB
 - Monitor, analyze, and optimize their ad campaigns
- Built on the principles of transparency, customizability, and control
- Key offerings:
 - DSP: Highly customizable - out-of-the-box
 - BaaS™: Fully customizable – own bidder - single-tenant architecture

BEESWAX DATA PLATFORM

BEESWAX HIGH-LEVEL ARCHITECTURE



ABOUT OUR DATA

Main ad event types received by the data platform

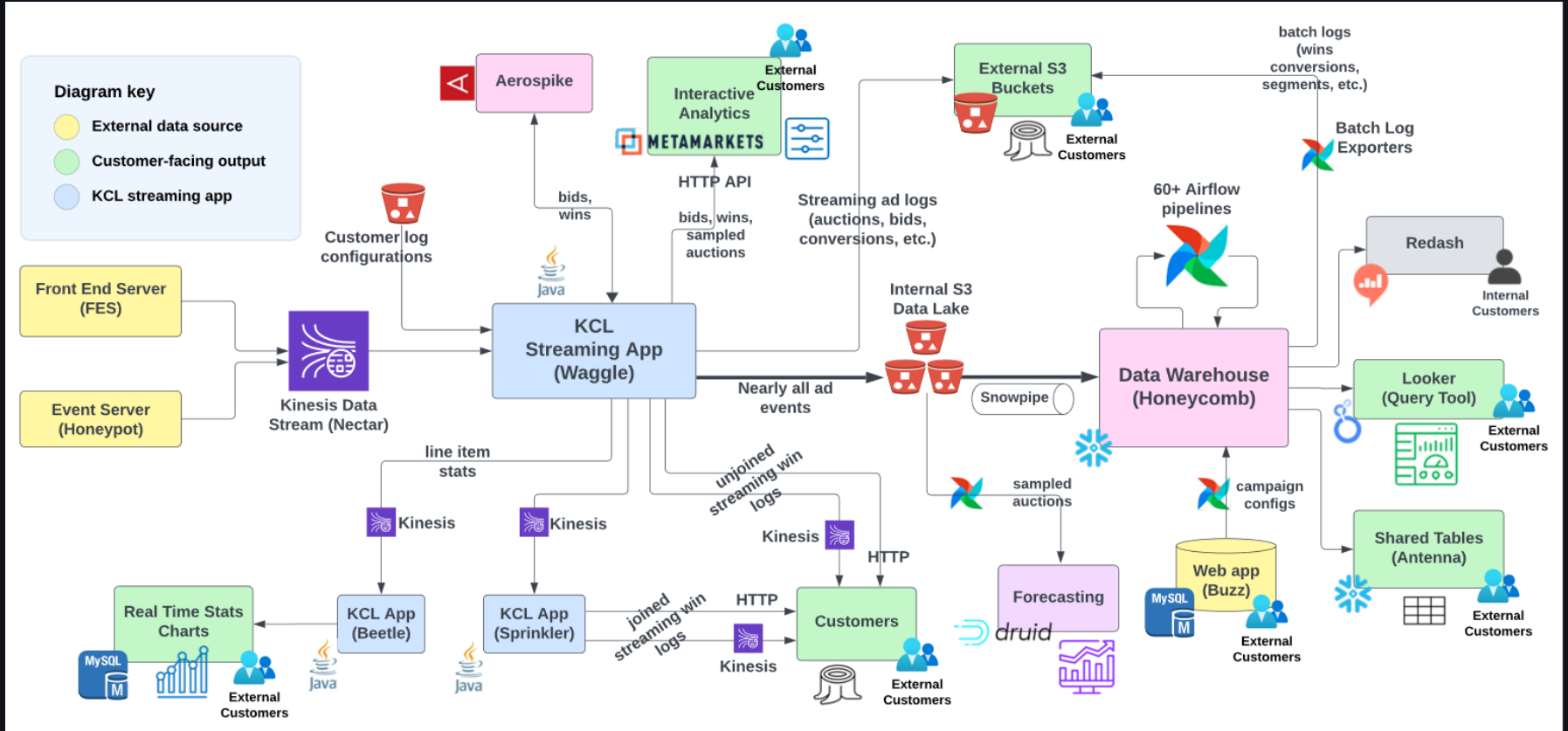
- Bid Request
- Bid Response
- Impression
- Click
- Activity
- Conversion
- Much less frequent: Bot Clicks, VAST Error, SKAd, and more

DATA PLATFORM

Overview

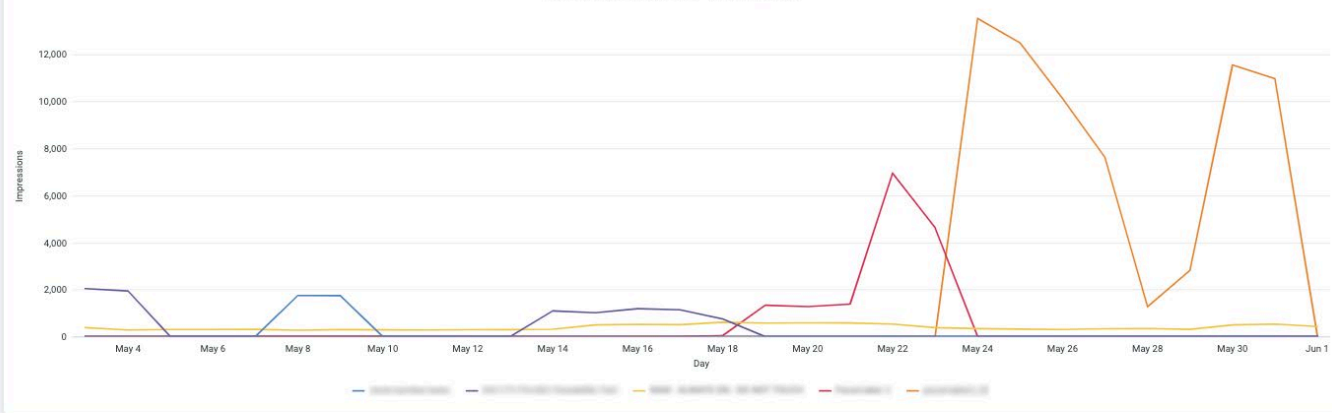
- Consumes ad events and ad campaign data
- Filters, deduplicates, anonymizes, joins, enriches, and aggregates data
- Acts as the single source of truth for data.
- Enables customers to monitor, analyze, and forecast campaign performance
- Provides:
 - Real-time visual analytics: bid/win/spend stats and charts
 - High performance interactive analytics: auction, bid, and win multi-dimensional analysis
 - Raw event-level data: streaming and batch delivery, Snowflake shared tables
 - Aggregate reports and dashboards: Looker-based UI and API, batch file delivery

DATA PLATFORM ARCHITECTURE



110,274 Total Impressions	232 Total Clicks	0.21% Average CTR	\$247.15 Total Spend	5 Active Campaigns	61 Active Line Items
-------------------------------------	----------------------------	-----------------------------	--------------------------------	------------------------------	--------------------------------

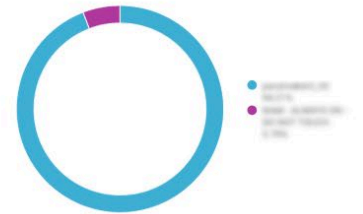
Performance Dashboard - Trending KPIs



Yesterday's Highlights

\$13.63 Yesterday's Spend	11,493 Yesterday's Impressions
-------------------------------------	--

Yesterday's Impression Breakdown



Top Performers

Your top performers, based on the view and KPI selected above.

Top 10 by Spend		Top 10 by Impressions		Top 10 by CPM	
Campaigns	Spend	Campaigns	Impressions	Campaigns	CPM
1	\$91.31	1	70,344	1	9.99
2	\$75.97	2	15,600	2	4.81
3	\$56.32	3	11,707	3	2.87
4	\$13.54	4	9,138	4	1.08
5	\$10.00	5	3,485	5	0.87

Analytics

Real Time Stats

Forecasting

Bid Decision Flow

Lookback Range 🕒 Last 24 hours ▾

📄 Data as of 06/01/2024 at 04:44:47 PM 🔄 Refresh

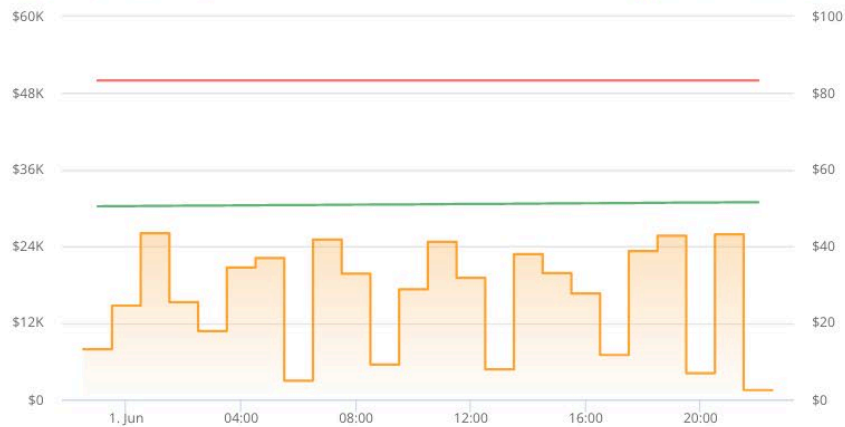
Bids and Wins

● Bids ● Wins



Spend

— Budget — Total Spend ● Incremental Spend



Auctions +

Day
May 31 - Jun 1, 2024 (1-1PM) EDT Compare to: Previous day
May 30-31, 2024

+ Metrics

Metric Trends by Hour ▾

+ Dimensions

Dimension Leaders by Requests ▾



App/Site Category	Requests
Arts & Entertainment	54.2M
Not Available	45.9M
Hobbies & Interests	9.2M
News	5.9M
Technology & Computing	4.0M
Sports	3.6M
Uncategorized	2.6M
Business	2.5M
Food & Drink	2.3M
Society	2.0M
Health & Fitness	1.9M
Education	1.9M

Device Type	Requests
Set Top Box[NR1]	54.46M
Mobile/Tablet	41.37M
Connected TV	38.63M
Personal Computer	15.92M
Tablet	3.17M
Games Console	1.08M
Phone	0.20M
Connected Device	0.09M
Not Available	0.01M

Region/State	Requests
USA/NJ	25.0M
USA/CA	23.9M
USA/NY	14.6M
USA/NV	13.9M
USA/TX	9.8M
ENG	9.4M
USA/MA	6.3M
000/00	3.3M
USA/CO	1.7M
GBR/MAN	1.5M
GBR/BIR	1.2M
GBR/GLG	1.2M

DATA SCALE

Waggle Streaming Application



EVERY SECOND

- ~3 Million Input Ad Events
- >5 GB of Input Data



EVERY DAY

- > 250 Billion Input Ad Events
- > 4 TB of Input Data

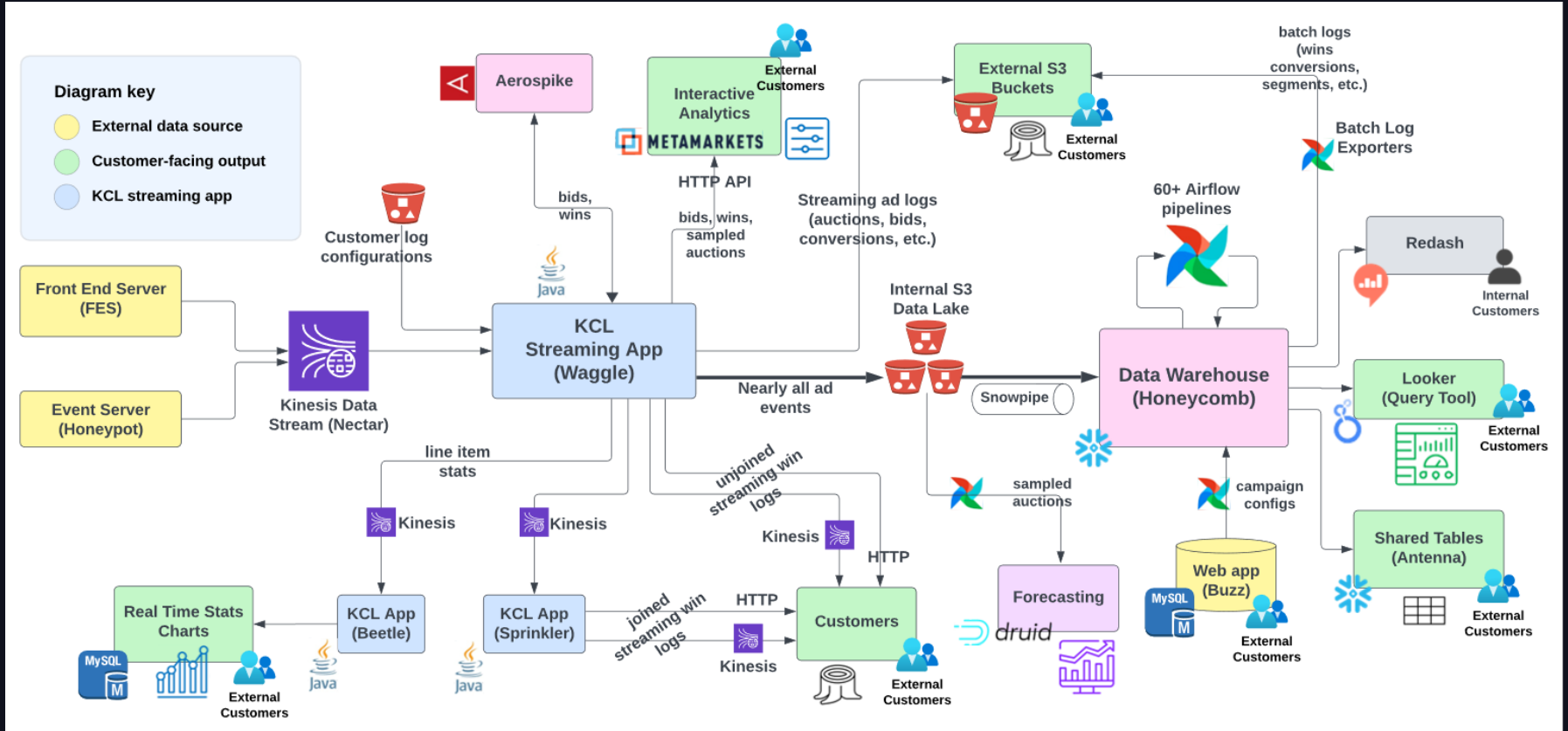
REDESIGN AND MIGRATION OF STREAMING APPLICATION

WAGGLE

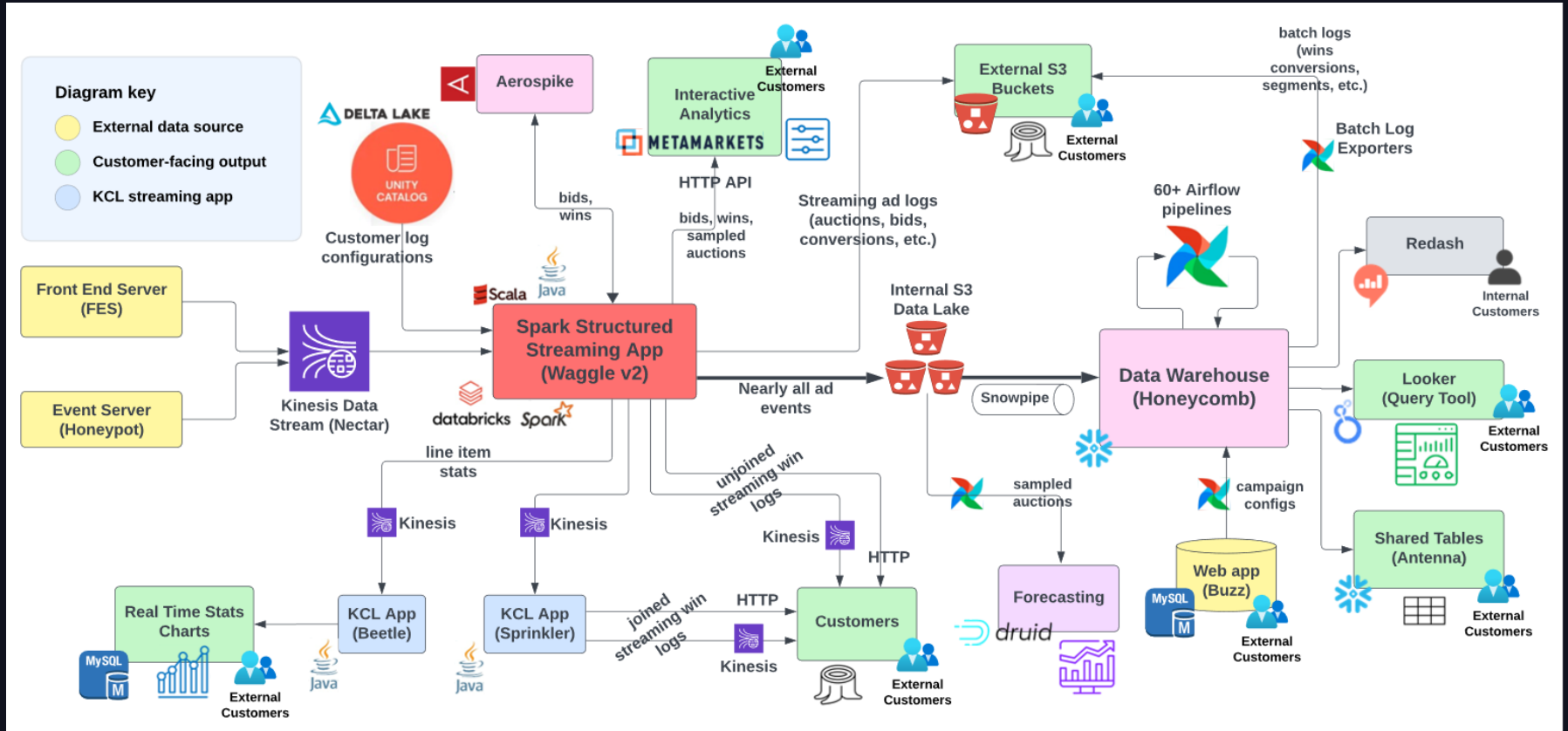
Streaming application at the frontline of the Data Platform

- Waggle: name of dance used by honeybees to communicate with each other
- Reads data in real-time from Kinesis Stream written to by upstream data sources
- Massive application with over 30K lines of code
- Java KCL application migrated to Scala/Java Spark Structured Streaming
- Transforms, filters, anonymizes, and joins ad events
- Joins using Aerospike database – Waggle itself remains stateless
- Routes transformed ad events to S3, Kinesis, and HTTP endpoints
- Delivers streaming ad logs and writes data to our warehouse

DATA PLATFORM ARCHITECTURE



DATA PLATFORM ARCHITECTURE



REDESIGN AND MIGRATION OBJECTIVES

Target KPIs

- System stability
- Cost inefficiencies
- Avoidable maintenance time
- Data quality dimensions:
 - Accuracy
 - Completeness
 - Timeliness
 - Delivery semantics

REDESIGN AND MIGRATION STRATEGY

To improve target KPIs

- Rearchitect system using top-down design
- Migrate from Kinesis Client Library (KCL) to Spark Structured Streaming on Databricks:
 - Eliminate memory leaks
 - Significantly reduce instance failures
 - Eliminate concurrency issues
 - Managed infrastructure
 - Built-in micro-batching
- Use Scala for new code and convert some of the Java code to Scala

RESULTS

Target KPIs

- System stability
- Cost inefficiencies
- Avoidable maintenance
- Data quality dimensions:
 - Accuracy
 - Completeness
 - Timeliness
 - Delivery semantics



- Improved
- Mostly eliminated
- Greatly reduced
- Data quality dimensions:
 - Improved
 - Improved
 - Improved
 - Closer to exactly once

APPLICATION STREAMING LOGIC

High Level Outline

SCALA

```
1  val streamingDataFrame = kinesisReader.getStreamingData(appConfig)
2  val dataStreamWriter = streamingDataFrame.writeStream.trigger(Trigger.ProcessingTime("60 seconds"))
3
4  dataStreamWriter.foreachBatch { (df, batchId) =>
5    df.persist()
6    val waggleMessageDS: Dataset[WaggleMessage] = df.as[KinesisRecord].map(decodeAndTransform)
7    val s3MessageDS: Dataset[S3Message] = waggleMessageDS.flatMap(_.s3Message)
8
9    s3MessageDS.foreachPartition { msgs: Iterator[S3Message] =>
10      msgs.toVector.groupBy(_.destination).foreach { case (destination, records) =>
11        writeRecordsToS3(location, records)
12      }
13    }
14    //output to other endpoints
15    df.unpersist()
16 }
```

MONITORING AND OBSERVABILITY

MONITORING AND OBSERVABILITY

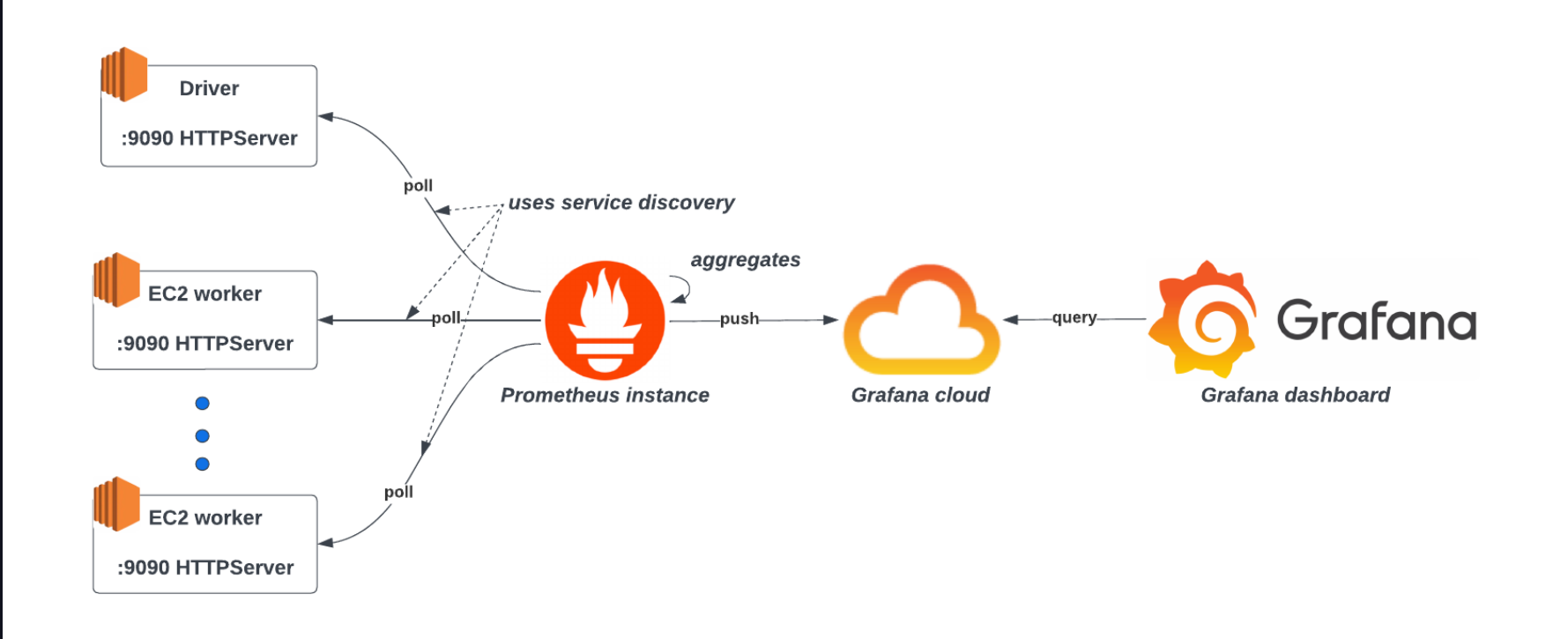
- Monitoring is crucial for ensuring system reliability and performance
- Challenges
 - Scalability
 - Flexibility
 - Reliability
 - Compatibility
 - ...
- Traditional monitoring solutions often struggle to address these challenges effectively

MONITORING AND OBSERVABILITY

- Prometheus - monitoring
 - A widely used open-source monitoring and alerting tool
 - Designed for reliability, scalability, and easy integration with other tools
 - Key features:
 - ✓ Multi-dimensional data model
 - ✓ PromQL
 - ✓ Pull-based metrics collection
 - ✓ Service discovery
 - ✓ ...
- Grafana – visualization and alerting management

MONITORING AND OBSERVABILITY

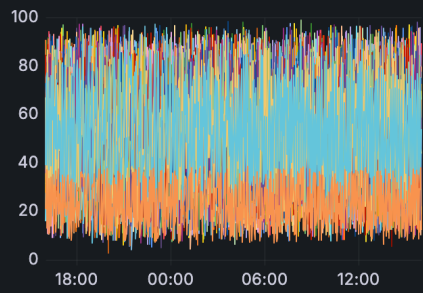
Architecture Diagram



MONITORING AND OBSERVABILITY

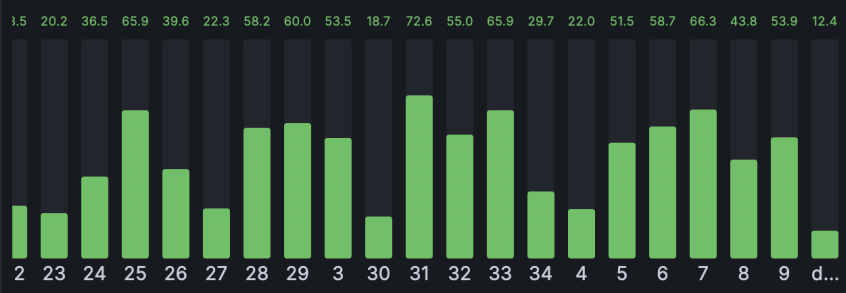
Cluster JVM

Memory utilization history (%)

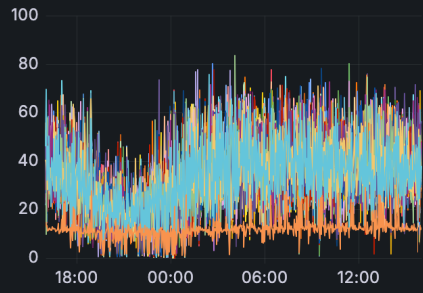


Executor 6 - 10.58.216.30	53.3
Executor 13 - 10.58.221.215	53.3
Executor 11 - 10.58.213.174	53.3
Executor 2 - 10.58.215.135	53.2
Executor 3 - 10.58.220.227	53.2
Executor 31 - 10.58.217.253	53.1
Executor 23 - 10.58.218.54	53.1
Executor 19 - 10.58.222.37	53.1
Executor 4 - 10.58.220.105	53.1

Latest memory utilization (%)

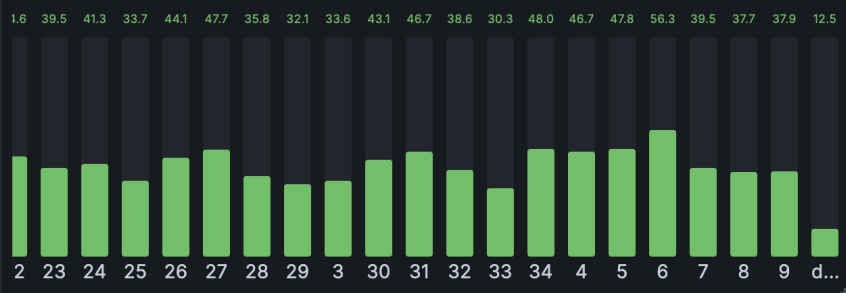


CPU utilization history (%)



Name	Mean
Executor 13 - 10.58.221.215	34.6
Executor 25 - 10.58.212.57	34.5
Executor 2 - 10.58.215.135	34.2
Executor 22 - 10.58.215.53	34.1
Executor 20 - 10.58.214.71	33.8
Executor 26 - 10.58.220.60	33.8
Executor 15 - 10.58.216.192	33.7
Executor 11 - 10.58.213.174	33.6

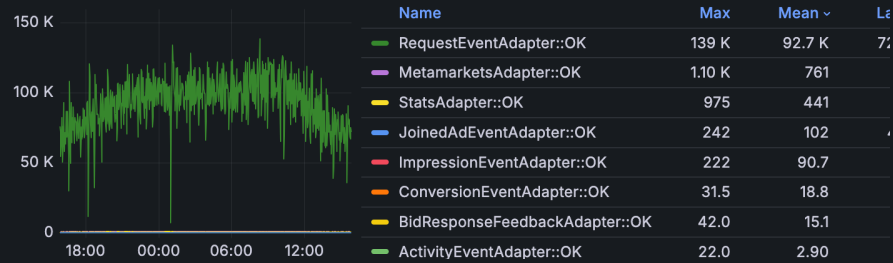
Latest CPU utilization (%)



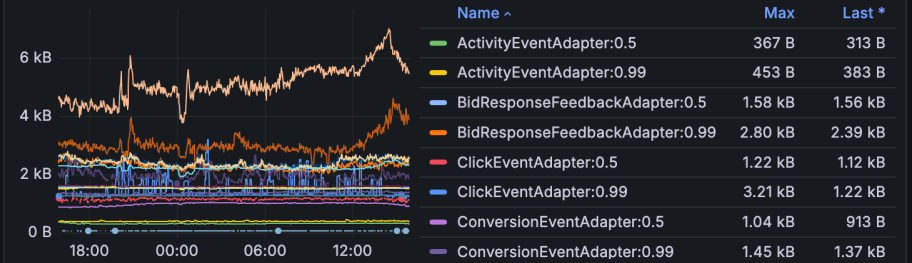
MONITORING AND OBSERVABILITY

~ Spark Application

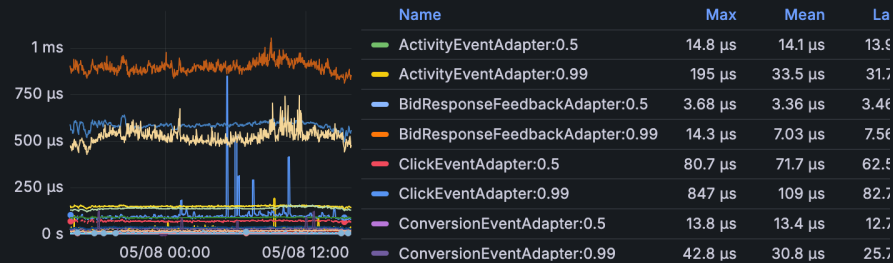
Number of events processed by adapter (per second)



Event data size histogram by adapter



Adapter processing time



Kinesis record size



MONITORING AND OBSERVABILITY

Implementation

- Use the secondary IP address to scrape for Prometheus metrics
 - Prometheus EC2 service discovery only gets the primary IP address
 - Our workaround
 - Have an EC2 tag for the secondary IP address
 - Relabel IP address to replace the primary with the secondary IP address

```
- job_name: "spark-app"
  scrape_interval: 60s
  ec2_sd_configs:
    ...
  relabel_configs:
    - source_labels: [__meta_ec2_tag_SecondaryPrivateIP]
      replacement: ${1}:9090
      target_label: __address__
```

MONITORING AND OBSERVABILITY

Implementation

- Incorporated Spark Dropwizard metrics to monitor the underlying Spark platform
- Created another Grafana dashboard as a central place for Databricks logs
 - Configured Databricks to send logs to AWS S3
 - Ingested logs from S3 into Grafana Cloud
 - Grafana dashboard provides searching and filtering capabilities

MONITORING AND OBSERVABILITY

Home > Dashboards > DPE > Waggle V2 Databricks Logs ☆

env prod region eu-west-1 cluster_id Enter variable value node_ip Enter variable value app_id Enter variable value executor_id Enter variable value filter_string Enter variable value

GitHub Source

Driver - [run_id, filter_string]

Driver [log4j] [ERROR][FATAL]

No data

Driver [stdout]

```
" s3_bucket=beeswax-logs-eu-west-1 s3_key=prod/wagglev2/databricks-logs/0529-123246-20b291sk/driver/stdout
> message="2024-05-29T15:35:07.152Z batch 80763 - ReportingLog.Conversion.-1026896932 9993 rows 2.3 MB - rep
" s3_bucket=beeswax-logs-eu-west-1 s3_key=prod/wagglev2/databricks-logs/0529-123246-20b291sk/driver/stdout
> message="2024-05-29T15:35:06.749Z batch 80763 - ReportingLog.BidResponseFeedback.-874684763 2732 rows 155.4
" s3_bucket=beeswax-logs-eu-west-1 s3_key=prod/wagglev2/databricks-logs/0529-123246-20b291sk/driver/stdout
> message="2024-05-29T15:35:06.549Z batch 80763 - ReportingLog.MergedVendorFee.-1056228554 32917 rows 9.8 MB
" s3_bucket=beeswax-logs-eu-west-1 s3_key=prod/wagglev2/databricks-logs/0529-123246-20b291sk/driver/stdout
> message="2024-05-29T15:35:06.318Z batch 80763 - ReportingLog.BidResponse.2081618794 893091 rows 951.4 MB -
" s3_bucket=beeswax-logs-eu-west-1 s3_key=prod/wagglev2/databricks-logs/0529-123246-20b291sk/driver/stdout
> message="2024-05-29T15:35:06.025Z batch 80763 - ReportingLog.BidRequest.1830871200 642054 rows 894.8 MB - r
" s3_bucket=beeswax-logs-eu-west-1 s3_key=prod/wagglev2/databricks-logs/0529-123246-20b291sk/driver/stdout
```

Driver [stderr]

No data

Executor - [run_id, app_id, executor_id, filter_string]

Executor [stdout]

```
> message="2024-05-29T12:29:34.983Z S3 batch 80599/96131 [REDACTED] - took 0.034 seconds, fil
" s3_bucket=beeswax-logs-eu-west-1 s3_key=prod/wagglev2/databricks-logs/0523-142422-70urwscz/executor/app-2
> message="2024-05-29T12:29:34.977Z S3 batch 80599/96131 [REDACTED] - took 0.038 seconds, f
" s3_bucket=beeswax-logs-eu-west-1 s3_key=prod/wagglev2/databricks-logs/0523-142422-70urwscz/executor/app-2
> message="2024-05-29T12:29:34.976Z S3 batch 80599/96131 [REDACTED] - took 0.041 seconds, file
" s3_bucket=beeswax-logs-eu-west-1 s3_key=prod/wagglev2/databricks-logs/0523-142422-70urwscz/executor/app-2
> message="2024-05-29T12:29:34.967Z S3 batch 80599/96131 [REDACTED] - took 0.073 seconds, f
" s3_bucket=beeswax-logs-eu-west-1 s3_key=prod/wagglev2/databricks-logs/0523-142422-70urwscz/executor/app-2
> message="2024-05-29T12:29:34.964Z S3 batch 80599/96131 [REDACTED] - took 0.025 seconds
```

Executor [stderr][ERROR][FATAL]

No data

Events - [run_id, filter_string] (1 panel)

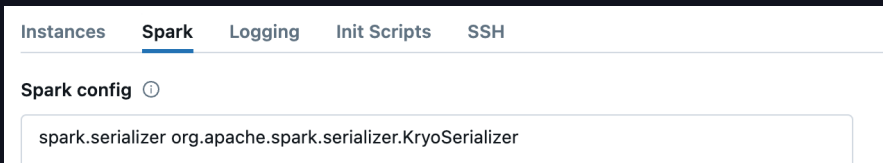
Init Scripts - [run_id, node_ip, filter_string] (2 panels)



PERFORMANCE OPTIMIZATION

PERFORMANCE OPTIMIZATION

- Used Kryo serialization
 - Kryo is faster than Java serialization
 - Spark uses Java serialization by default



The screenshot shows a Databricks configuration interface with tabs for 'Instances', 'Spark', 'Logging', 'Init Scripts', and 'SSH'. The 'Spark' tab is active, and the 'Spark config' section is expanded to show the configuration for 'spark.serializer', which is set to 'org.apache.spark.serializer.KryoSerializer'.

```
spark.serializer org.apache.spark.serializer.KryoSerializer
```

- Memory optimization
 - Used lookup table for reusable Java objects
 - Kept only necessary information for output in DataFrame/Dataset

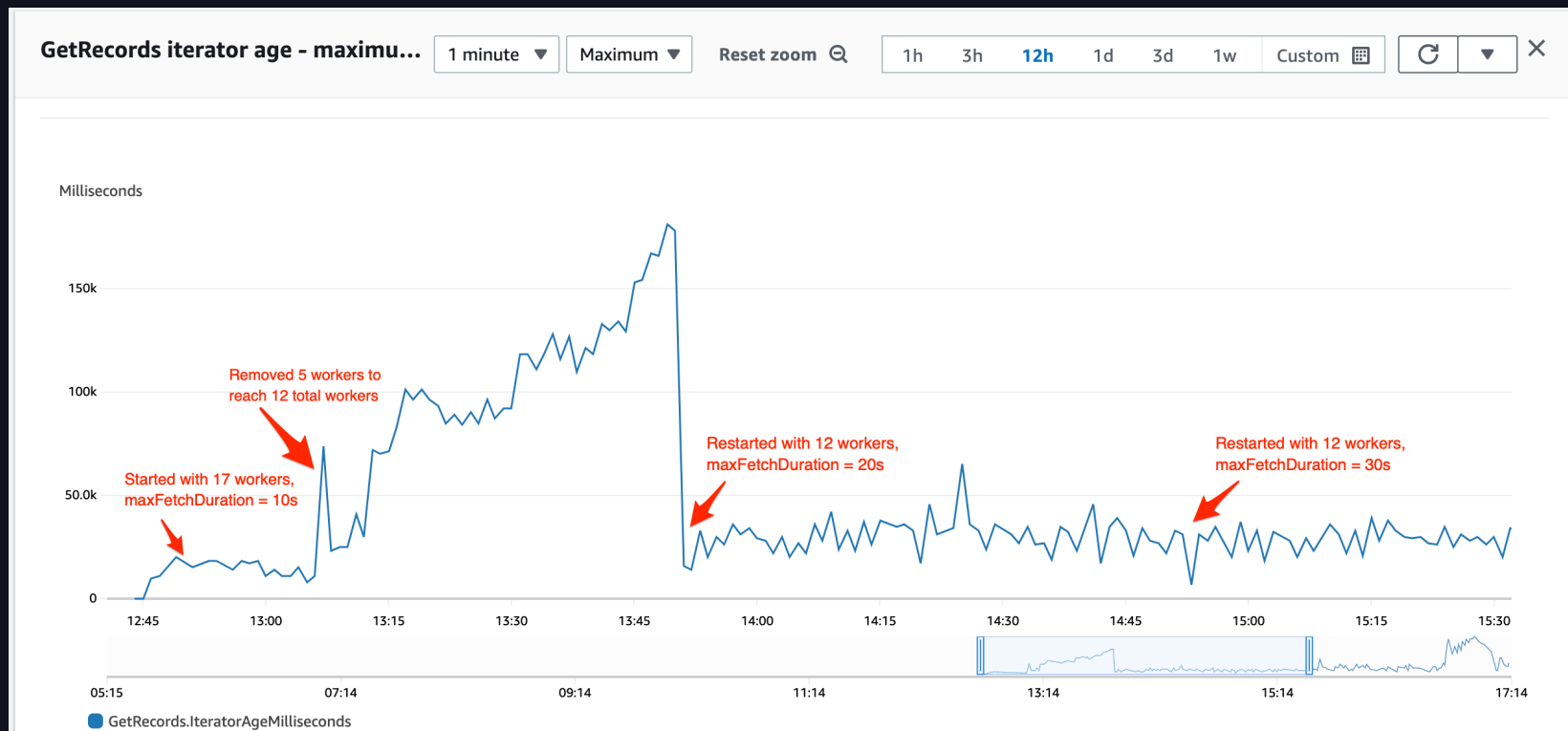
PERFORMANCE OPTIMIZATION

- Kinesis parameters

For micro-batch interval = 60 seconds

Parameter	Meaning	Default	What we use
maxFetchRate	Maximum prefetch data rate	1.0	2.0
maxFetchDuration	Buffer time of prefetched data before data is made available for processing	10s	30s
shardsPerTask	Number of Kinesis shards each Spark task handles.	5	15
coalesceThresholdBlock Size	The threshold at which automatic coalesce occurs.	10MB	50MB
coalesceBinSize	Approximate block size after coalescing.	128MB	128MB

PERFORMANCE OPTIMIZATION



PERFORMANCE OPTIMIZATION

- Parallel processing
 - Driver – simultaneous DataFrame processing
 - Executor – simultaneous S3 writes within custom Java code
- Caching
 - DataFrame
 - Reusable AWS resources (S3 clients, Kinesis producers, etc.)
- Future: Stream pipelining – serverless

PERFORMANCE OPTIMIZATION

```
2024-05-23T15:42:30.641Z S3 batch 73720/78706 - BEGIN
2024-05-23T15:42:30.849Z S3 batch 73720/78706 - 51 log config identifiers
...
...
2024-05-23T15:42:32.212Z S3 batch 73720/78706 - took 0.310 seconds, fileSize=249.4 kB
2024-05-23T15:42:32.239Z S3 batch 73720/78706 - took 0.293 seconds, fileSize=217.6 kB
2024-05-23T15:42:32.266Z S3 batch 73720/78706 - took 0.392 seconds, fileSize=414.7 kB
2024-05-23T15:42:32.324Z S3 batch 73720/78706 - took 0.261 seconds, fileSize=231.0 kB
2024-05-23T15:42:32.336Z S3 batch 73720/78706 - took 0.298 seconds, fileSize=95.6 kB
2024-05-23T15:42:32.372Z S3 batch 73720/78706 - took 0.354 seconds, fileSize=399.2 kB
2024-05-23T15:42:32.408Z S3 batch 73720/78706 - took 0.541 seconds, fileSize=706.9 kB
2024-05-23T15:42:32.447Z S3 batch 73720/78706 - took 0.502 seconds, fileSize=592.1 kB
2024-05-23T15:42:32.457Z S3 batch 73720/78706 - took 0.590 seconds, fileSize=646.8 kB
2024-05-23T15:42:32.504Z S3 batch 73720/78706 - took 0.637 seconds, fileSize=636.1 kB
2024-05-23T15:42:32.623Z S3 batch 73720/78706 - took 0.640 seconds, fileSize=692.5 kB
2024-05-23T15:42:32.758Z S3 batch 73720/78706 - took 0.663 seconds, fileSize=908.6 kB
2024-05-23T15:42:32.964Z S3 batch 73720/78706 - took 0.939 seconds, fileSize=1.5 MB
2024-05-23T15:42:32.971Z S3 batch 73720/78706 - took 1.146 seconds, fileSize=1.9 MB
2024-05-23T15:42:33.005Z S3 batch 73720/78706 - took 1.051 seconds, fileSize=1.6 MB
2024-05-23T15:42:33.123Z S3 batch 73720/78706 - took 1.203 seconds, fileSize=3.1 MB
2024-05-23T15:42:33.300Z S3 batch 73720/78706 - took 1.385 seconds, fileSize=3.7 MB
2024-05-23T15:42:33.318Z S3 batch 73720/78706 - took 1.433 seconds, fileSize=3.5 MB
2024-05-23T15:42:33.318Z S3 batch 73720/78706 - END 2.677 seconds
```

DATABRICKS DEVELOPMENT EXPERIENCE

DATABRICKS DEVELOPMENT EXPERIENCE

Useful Features

- Intuitive APIs (for application code)
- Databricks Terraform Provider (for IAC)
- Databricks Asset Bundles (for CI/CD)
- DatabricksIQ-powered features:
 - Databricks Assistant
 - AI-Generated Comments

DATABRICKS ASSISTANT

- Helps quickly diagnose problems
- Provides useful code examples

Assistant

The error message indicates that the application is unable to connect to an Aerospike database server.

The error code "-8" indicates that a network error occurred either due to timeout or DNS resolution issues. This could happen due to a variety of reasons such as firewall issues, incorrect hostname or IP address, or Aerospike server issues.

To resolve this issue, check that you have properly specified the appropriate Aerospike server details (such as the IP address and port) and that security configurations are applied correctly. You can also check the status and logs of the Aerospike cluster to ensure that there are no issues on that side. Additionally, check for any network issues such as firewalls, proxies, or SSL verification that might be preventing the connection.

Once the possible issues are ruled out, try connecting to the Aerospike server again.

Enter your message

```
Stream stopped...
org.apache.spark.SparkException: Job aborted due to stage failure: Task 1 in stage 13.0 failed 4 times, most recent failure: Lost task 1.3 in stage 13.0 (TID 68) (10.43.105.213 executor 0): com.aerospike.client.AerospikeException$Connection: Error -8: Failed to connect to [1] host(s):
10.54.245.4 3000 Error -8: java.net.SocketTimeoutException: connect timed out

at com.aerospike.client.cluster.Cluster.seedNode(Cluster.java:707)
at com.aerospike.client.cluster.Cluster.tend(Cluster.java:544)
at com.aerospike.client.cluster.Cluster.waitTillStabilized(Cluster.java:497)
at com.aerospike.client.cluster.Cluster.initTendThread(Cluster.java:429)
at com.aerospike.client.cluster.Cluster.<init>(Cluster.java:383)
at com.aerospike.client.AerospikeClient.<init>(AerospikeClient.java:294)
at com.beeswax.wagglev2.waggle.joiner.connectors.AerospikeConnector.<init>(AerospikeConnector.java:61)
at com.beeswax.wagglev2.waggle.joiner.JoinerProcessor.<init>(JoinerProcessor.java:44)
at com.beeswax.wagglev2.waggle.joiner.JoinerProcessor.getInstance(JoinerProcessor.java:56)
at com.beeswax.wagglev2.KinesisRecordParser$.anonfun$parse$1(KinesisRecordParser.scala:117)
at scala.collection.immutable.List.flatMap(List.scala:366)
at com.beeswax.wagglev2.KinesisRecordParser$.parse(KinesisRecordParser.scala:114)
at com.beeswax.wagglev2.streaming.WaggleStreamer$.anonfun$transform$1(WaggleStreamer.scala:166)
at scala.collection.Iterator$$anon$11.nextCur(Iterator.scala:486)
at scala.collection.Iterator$$anon$11.hasNext(Iterator.scala:492)
at scala.collection.Iterator$$anon$10.hasNext(Iterator.scala:460)
at org.apache.spark.sql.catalyst.expressions.GeneratedClass$GeneratedIteratorForCodegenStage1.processNext

(Unknown Source)
```

AI-GENERATED COMMENTS

- Accurate
- Saves time
- Encourages good habits

Tags:

AI Suggested Comment Preview ✕

The 'kinesis_log_config' table stores configuration information related to the logging of data in the Kinesis platform. It includes details such as the unique buzz key used for logging, the stream name where the data will be stored, and the credentials information required for accessing the stream. Additionally, it keeps track of the UTC timestamp indicating when the configuration was last updated. This table can be used to manage and configure logging for different streams in the Kinesis platform, ensuring consistent and appropriate logging practices across various systems.

Columns | [Sample Data](#) | [Details](#) | [Permissions](#) | [History](#) | [Lineage](#) | [Insights](#)

Column	Type	Comment
buzz_key	string	The unique identifier for each log entry in the waggle log. ✓
stream_name	string	The name of the stream where the log entries are stored. ✓
credentials_info	string	Additional information about the credentials required to access the stream. ✓
update_time_utc	timestamp	The timestamp indicating when the configuration was last updated. ✓

FUTURE DATABRICKS WORK

FUTURE DATABRICKS WORK

- Trying out these upcoming Databricks features:
 - Serverless compute for streaming workflows
 - Autoscaling for serverless streaming workflows
- More Databricks Compute and Unity Catalog adoption

THANK YOU!