Real-Time Streaming

Intro to Amazon Kinesis

Roy Ben-Alta
Architect and Business Development Manager
Amazon Web Services - Big Data, IoT Analytics Solutions

25th of October 2016

amazon web services | Pop-up Loft

Agenda

Streaming data on AWS and Customer scenarios

Amazon Kinesis platform overview

Demo

Q&A

AWS provides the broadest platform for big data analytics in the market today.

Big Data Storage

Data Warehousing

Real-time Streaming

Distributed Analytics (Hadoop, Spark, Presto)

NoSQL Databases

Business Intelligence

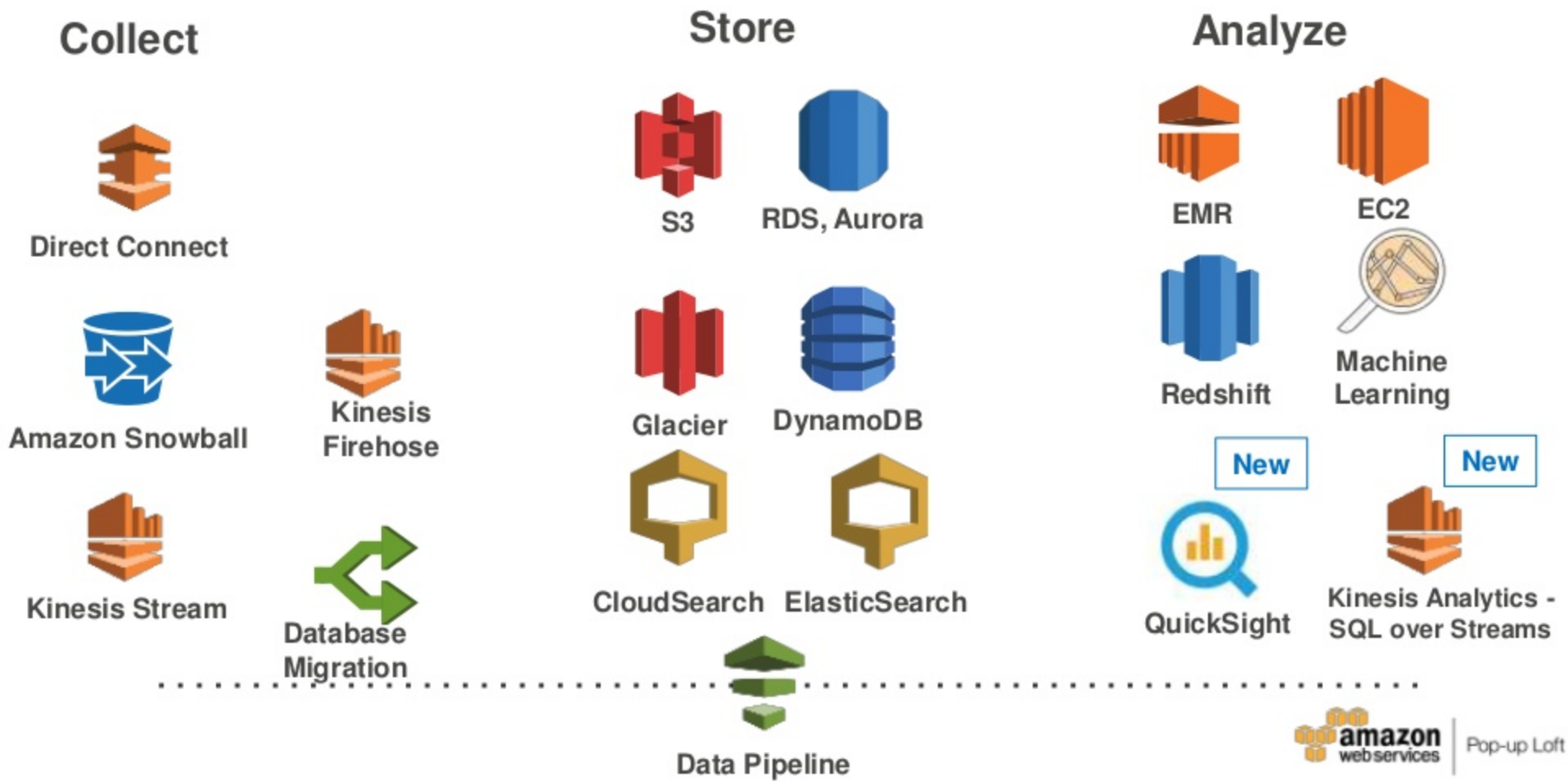
Relational Databases

Internet of Things (IoT)

Machine Learning

Server-less Compute

AWS Big Data Portfolio



Most Data is produced continuously

Mobile Apps

Web Clickstream

```
[Wed Oct 11 14:32:52
2000] [error] [client
127.0.0.1] client
denied by server
configuration:
/export/home/live/ap/h
tdocs/test
```

Application Logs

Metering Records

IoT Sensors

Smart Buildings

What is streaming data? Data that is…

- **Moving** – captured and processed with low latency (in ms to low single-digit sec)
- **Small** (typically < 5KB) with high frequency (from hundreds to millions of data records per sec)
- **Sequenced** – Data is produced, captured, and processed by sequence, by time or a derivative

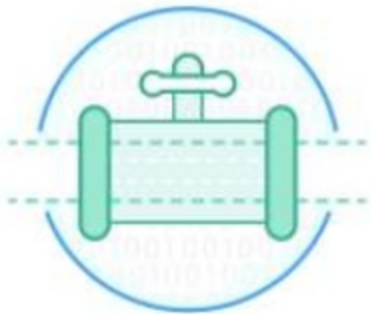
Streaming Data on AWS: Customer Scenarios

Scenarios	1 Accelerated Ingest-Transform-Load to final destination	2 Continual Metrics/ KPI Extraction	3 Responsive Data Analytics
Ad Tech/ Marketing Analytics	Advertising data aggregation	Advertising metrics like coverage, yield, conversion, scoring webpages	User activity engagement analytics, optimized bid/ buy engines
Consumer Online/ Gaming	Online customer engagement data aggregation	Consumer/ app engagement metrics like page views, CTR	Customer clickstream analytics, recommendation engines,
Financial Services	Market/ financial transaction order data collection	Financial market data metrics	Fraud monitoring, and value-at-risk assessment, auditing of market order data
IoT / Sensor Data	Fitness device , vehicle sensor, telemetry data ingestion	Wearable sensor operational metrics, and dashboards	Devices / sensor operational intelligence

Amazon Kinesis Platform

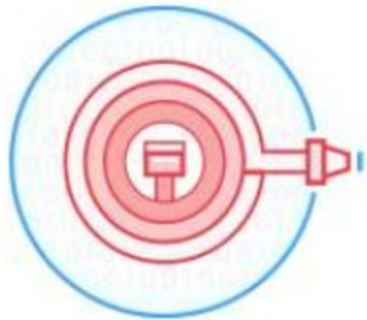
Amazon Kinesis: Streaming Data Made Easy

Services make it easy to capture, deliver, process streams on AWS



Amazon Kinesis Streams

- For Technical Developers
- Build your own custom applications that process or analyze streaming data



Amazon Kinesis Firehose

- For ETL, Data Engineer
- Easily load massive volumes of streaming data into S3, Amazon Redshift and Amazon Elasticsearch service

Amazon Kinesis Analytics

- For all developers, data scientists
- Easily analyze data streams using standard SQL queries

Amazon Kinesis Streams

Build your own data streaming applications

Send clickstream data to Kinesis Streams

Kinesis Streams stores and exposes clickstream data for processing

Custom application built on Kinesis Client Library makes real-time content recommendations

Readers see personalized content suggestions

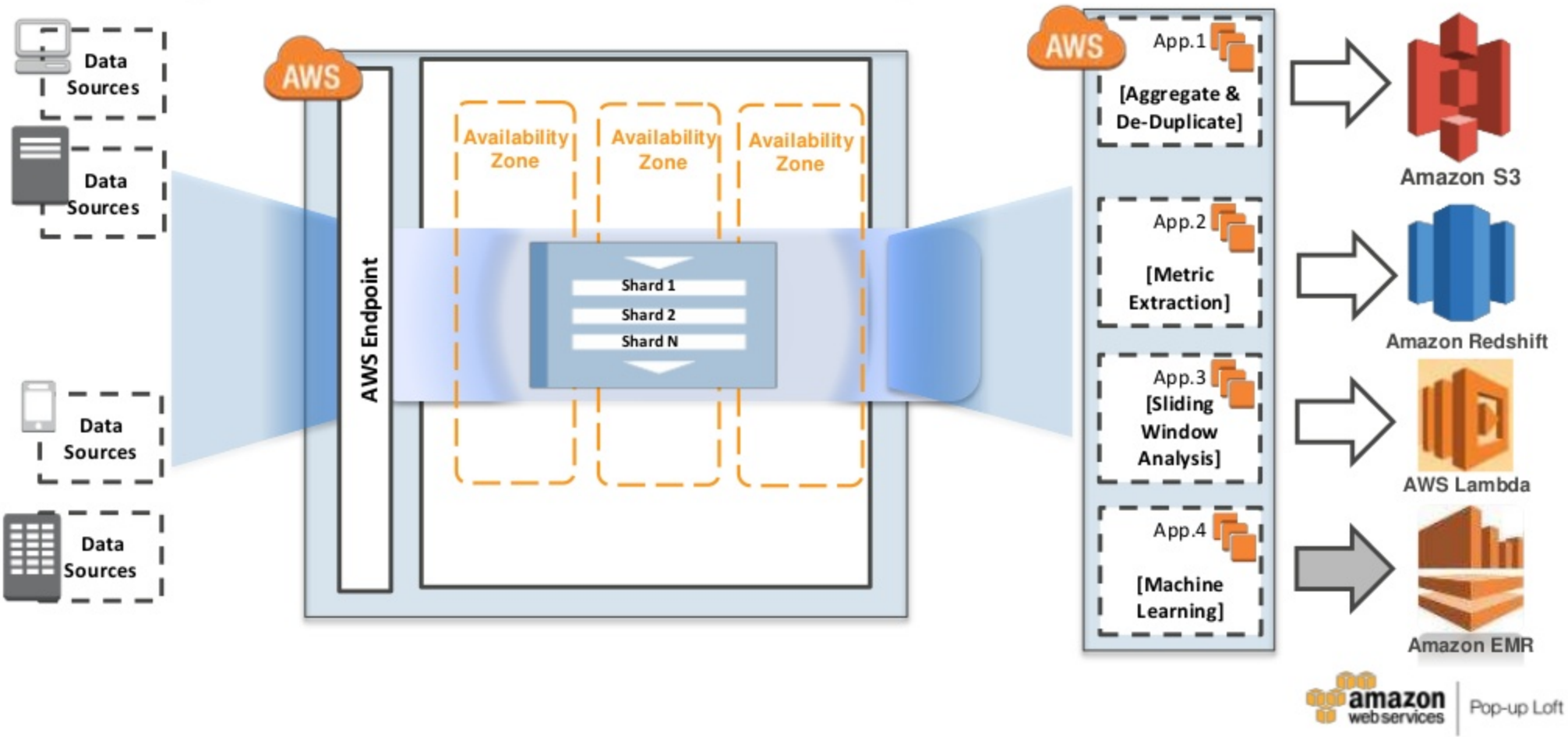
Easy administration: Simply create a new stream, and set the desired level of capacity with shards. Scale to match your data throughput rate and volume.

Build real-time applications: Perform continual processing on streaming big data using Amazon Kinesis Client Library (KCL), Apache Spark/Storm, AWS Lambda, and more.

Low cost: Cost-efficient for workloads of any scale.

Amazon Kinesis Streams

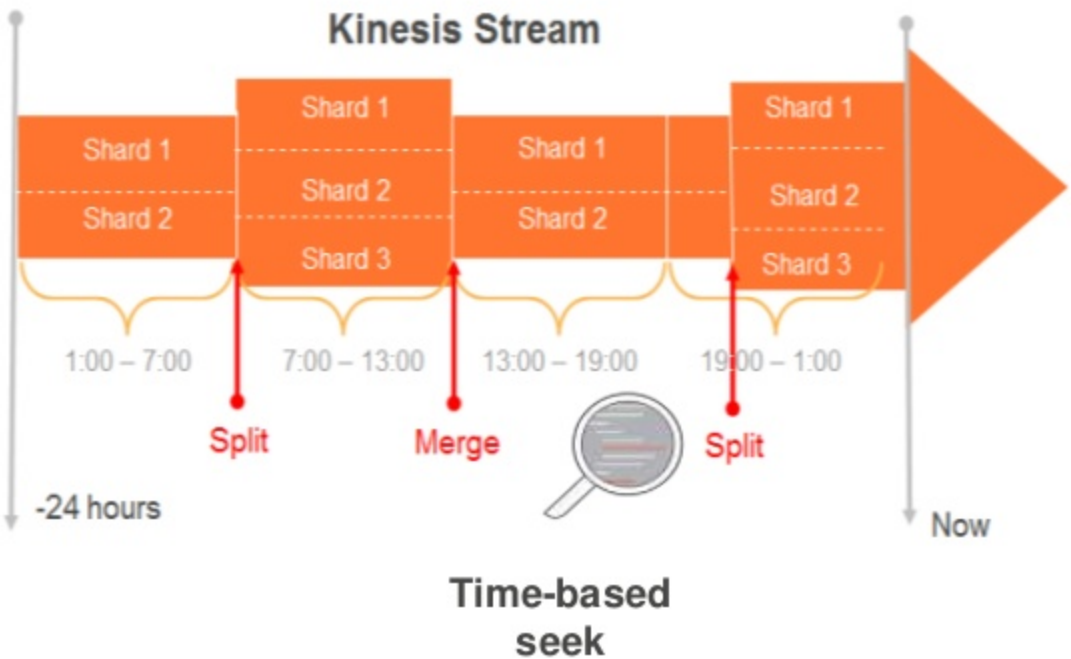
Managed service for real-time streaming



amazon web services | Pop-up Loft

Amazon Kinesis Streams

Managed Ability to Capture and store Data



Time-based seek

- Streams are made of **shards**
- Each shard ingests up to 1 MB/sec, and 1000 records/sec
- Each shard emits up to 2 MB/sec
- All data is **stored for 24 hours by default**; storage can **be extended for up to 7 days**
- **Scale** Kinesis streams using scaling util
- **Replay** data inside of 24-hour window

Sending & Reading from Amazon Kinesis

Amazon Kinesis Streams 3rd Party Connectors

Amazon Kinesis Customer Base Diversity

Amazon Kinesis as Databus

1 billion events/wk from connected devices | IoT

17 PB of game data per season | Entertainment

80 billion ad impressions/day, 30 ms response time | Ad Tech

300 GB/day click streams from 300+ sites | Enterprise

50 billion ad impressions/day sub-50 ms responses | Ad Tech

Sleep tracker sensor analysis | IoT

Funnel all production events through Amazon Kinesis

amazon web services | Pop-up Loft

Streaming Data on AWS: Customer Scenarios

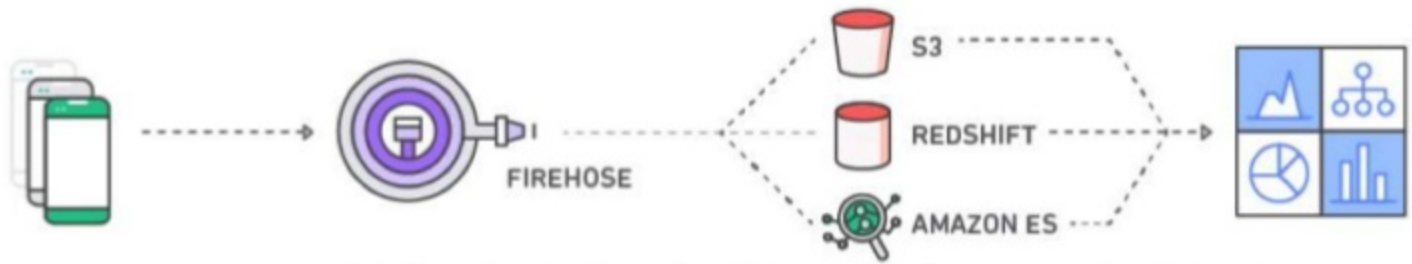
Scenarios	1 Accelerated Ingest-Transform-Load to final destination	2 Continual Metrics/ KPI Extraction	3 Responsive Data Analytics
Ad Tech/ Marketing Analytics	Advertising data aggregation	Advertising metrics like coverage, yield, conversion, scoring webpages	User activity engagement analytics, optimized bid/ buy engines
Consumer Online/ Gaming	Online customer engagement data aggregation	Consumer/ app engagement metrics like page views, CTR	Customer clickstream analytics, recommendation engines,
Financial Services	Market/ financial transaction order data collection	Financial market data metrics	Fraud monitoring, and value-at-risk assessment, auditing of market order data
IoT / Sensor Data	Fitness device , vehicle sensor, telemetry data ingestion	Wearable sensor operational metrics, and dashboards	Devices / sensor operational intelligence

Fault Tolerant ingestion and delivery at scale is difficult

A. All of the challenges associated with capturing the data
B. Anti-pattern to send data to many destinations with large amounts of small records
A. Issues downstream create backpressure on producers
B. Do not want to stop processing if one portion of pipeline is down
C. Need to perform data prep before persistence (compression, encryption, etc.)

Amazon Kinesis Firehose

Load massive volumes of streaming data into Amazon S3, Amazon Redshift, Amazon Elasticsearch service



Capture and submit streaming data to Firehose

Firehose loads streaming data continuously into S3, Amazon Redshift and Amazon Elasticsearch

Analyze streaming data using your favorite BI tools

Zero administration: Capture and deliver streaming data into Amazon S3, Amazon Redshift and Amazon Elasticsearch **without writing an application or managing infrastructure.**

Direct-to-data store integration: Batch, compress, and **encrypt** streaming data for delivery into data destinations **in as little as 60 secs** using simple configurations.

Seamless elasticity: Seamlessly scales to match data throughput w/o intervention

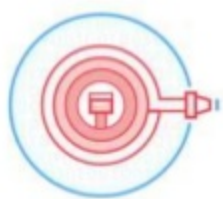
How does Amazon Kinesis Firehose help?

A. Provides scalable and durable ingest like Kinesis Streams (but not ordering)
B. Configurable aggregation of data before writing to final destination
C. Provides fully managed buffer with zero administration if destinations have issues
D. Independently delivers data to different destinations; separates out processing and delivery in the case of Kinesis Analytics integration
E. Configurable data-prep options like compression and encryption

Amazon Kinesis Firehose vs. Amazon Kinesis Streams

Amazon Kinesis Streams

Amazon Kinesis Streams is for use cases that require **custom processing**, per incoming record, with sub-1 second processing latency, and a choice of stream processing frameworks.



Amazon Kinesis Firehose

Amazon Kinesis Firehose is for use cases that require zero administration, ability to **use existing analytics tools based on Amazon S3, Amazon Redshift and Amazon Elasticsearch Service** and a data latency of 60 seconds or higher.

Streaming Data on AWS: Customer Scenarios

Scenarios	1 Accelerated Ingest-Transform-Load to final destination	2 Continual Metrics/ KPI Extraction	3 Responsive Data Analytics
Ad Tech/ Marketing Analytics	Advertising data aggregation	Advertising metrics like coverage, yield, conversion, scoring webpages	User activity engagement analytics, optimized bid/ buy engines
Consumer Online/ Gaming	Online customer engagement data aggregation	Consumer/ app engagement metrics like page views, CTR	Customer clickstream analytics, recommendation engines,
Financial Services	Market/ financial transaction order data collection	Financial market data metrics	Fraud monitoring, and value-at-risk assessment, auditing of market order data
IoT / Sensor Data	Fitness device , vehicle sensor, telemetry data ingestion	Wearable sensor operational metrics, and dashboards	Devices / sensor operational intelligence

Amazon Kinesis Analytics (New)

Connect to Amazon Kinesis streams, Firehose delivery streams

Run standard SQL queries against data streams

Amazon Kinesis Analytics can send processed data to analytics tools so you can create alerts and respond in real-time

Apply SQL on streams: Easily connect to an Amazon Kinesis stream or Firehose delivery Stream and apply SQL skills.

Build real-time applications: Perform continual processing on streaming big data with sub-second processing latencies.

Easy Scalability: Elastically scales to match data throughput.

Use SQL To Build Real-Time Applications

Connect to streaming source

Easily write SQL code to process streaming data

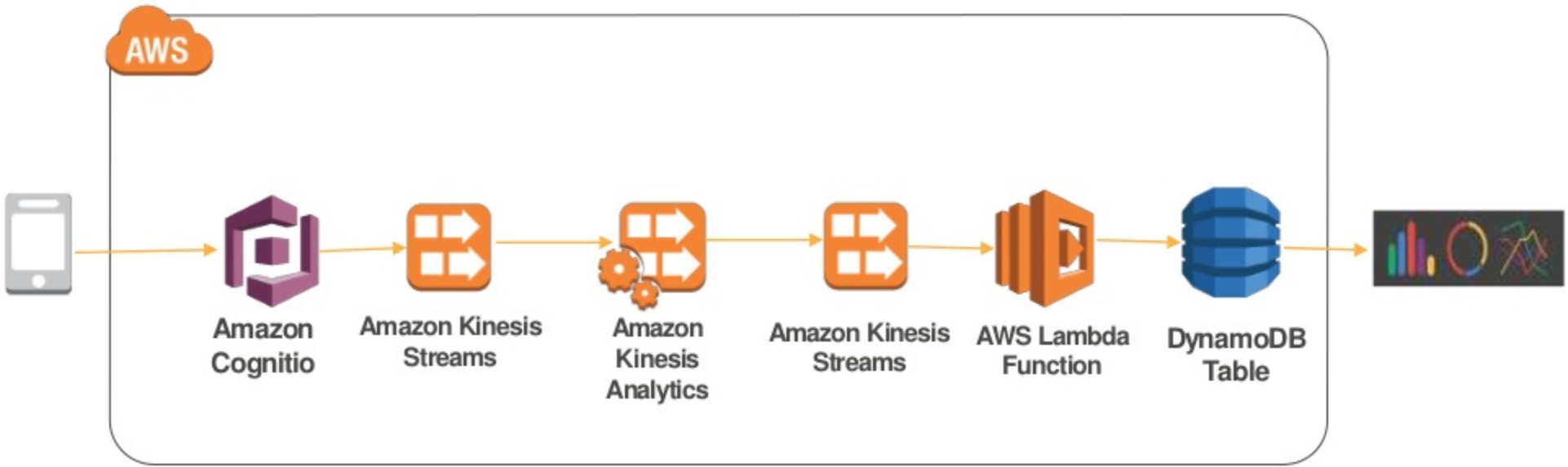
Continuously deliver SQL results

Real-time analytical patterns

- Pre-processing: filtering, transformations
- Basic Analytics: Simple counts, aggregates over windows
- Advanced Analytics: Detecting anomalies, event correlation
- Post-processing: Alerting, triggering, final filters

Demo

Building Serverless IoT Analytics stack



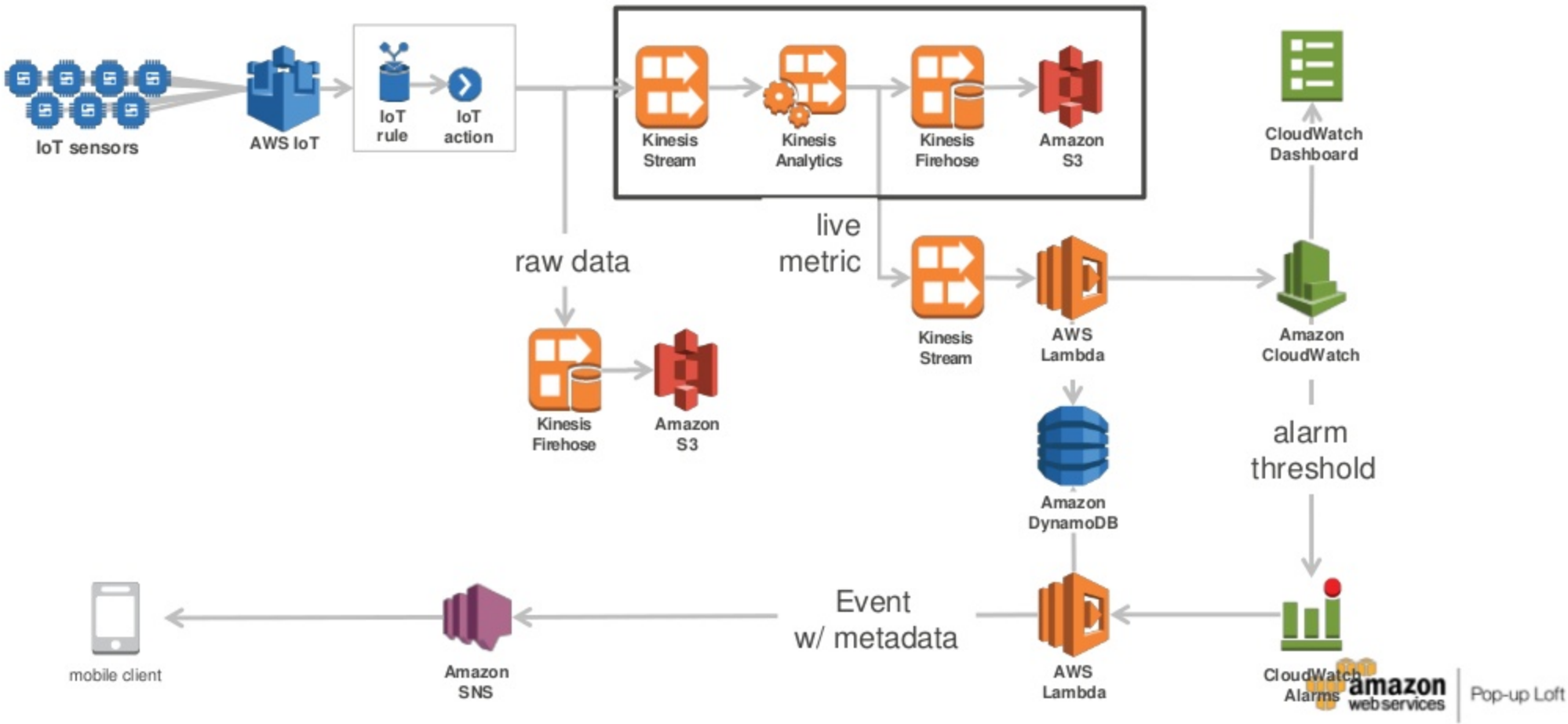
Streaming Data Platform for IoT Sensor Data

You have hundreds of IoT sensors that are producing data continuously that you need to ingest, analyze, and store.

You will:

1. Produce data continuously from hundreds of (simulated) IoT sensors
2. Durably ingest the incoming data using Kinesis Streams
3. Filter and aggregate the data using Kinesis Analytics
4. Deliver processed data to S3 using Kinesis Firehose

Mature Streaming IoT Platform Architecture



Summary

1. Leverage AWS managed services;
   Scalable/elastic, available, reliable, secure, no/low admin
2. Amazon kinesis is platform for streaming data ingestion, processing, and Analytics.
3. Create your first Amazon Kinesis stream. Configure hundreds of thousands of data producers to put data into an Amazon Kinesis stream.
4. Choose the Processing framework for your use case:
   Kinesis Analytics, Amazon EMR with Spark, Lambda, and more.
5. Check out our AWS Big Data Blog and find useful code sample for your use case:
   - Getting started with Amazon Kinesis Analytics
   - Spark SQL and Amazon Kinesis Streams
   - Building Near Real-Time Discover platform using Kinesis Firehose and Lambda

Thank you!

aws.amazon.com/kinesis

Reference

We have many AWS Big Data Blogs which cover more examples. Full list here. Some good ones:

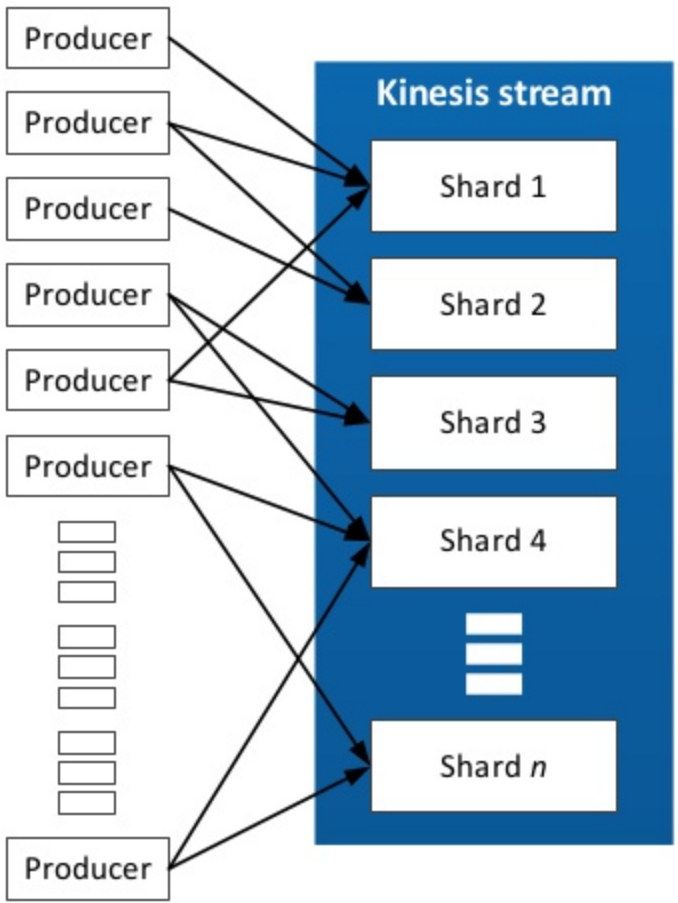
1. Kinesis Streams
    1. Implement Efficient and Reliable Producers with the Amazon Kinesis Producer Library
    2. Presto and Amazon Kinesis
    3. Querying Amazon Kinesis Streams Directly with SQL and Sparking Streaming
    4. Optimize Spark-Streaming to Efficiently Process Amazon Kinesis Streams
2. Kinesis Firehose
    1. Persist Streaming Data to Amazon S3 using Amazon Kinesis Firehose and AWS Lambda
    2. Building a Near Real-Time Discovery Platform with AWS
3. Kinesis Analytics
    1. Writing SQL on Streaming Data With Amazon Kinesis Analytics Part 1 | Part 2
    2. Real-time Clickstream Anomaly Detection with Amazon Kinesis Analytics

Reference

- **Technical documentations**
  - Amazon Kinesis Agent
  - Amazon Kinesis Streams and Spark Streaming
  - Amazon Kinesis Producer Library Best Practice
  - Amazon Kinesis Firehose and AWS Lambda
  - Building Near Real-Time Discovery Platform with Amazon Kinesis
- **Public case studies**
  - Glu mobile – Real-Time Analytics
  - Hearst Publishing – Clickstream Analytics
  - How Sonos Leverages Amazon Kinesis
  - Nordstorm Online Stylist

APPENDIX

Putting Data into a Kinesis stream



- Data producers call **PutRecord(s)** to send data to a Kinesis stream
- **PutRecord** {Data,StreamName,PartitionKey}
- **PutRecords** {Records{Data,PartitionKey}, StreamName}
- A Partition Key is supplied by producer and used to distribute (MD5 hash) the PUTs across (hash key range) of Shards
- A unique Sequence # is returned to the Producer upon a successful PUT call
- Options: AWS SDKs, Kinesis Producer Library (KPL), Kinesis Agent, FluentD, Flume, and more…

Most data producers are not reliable

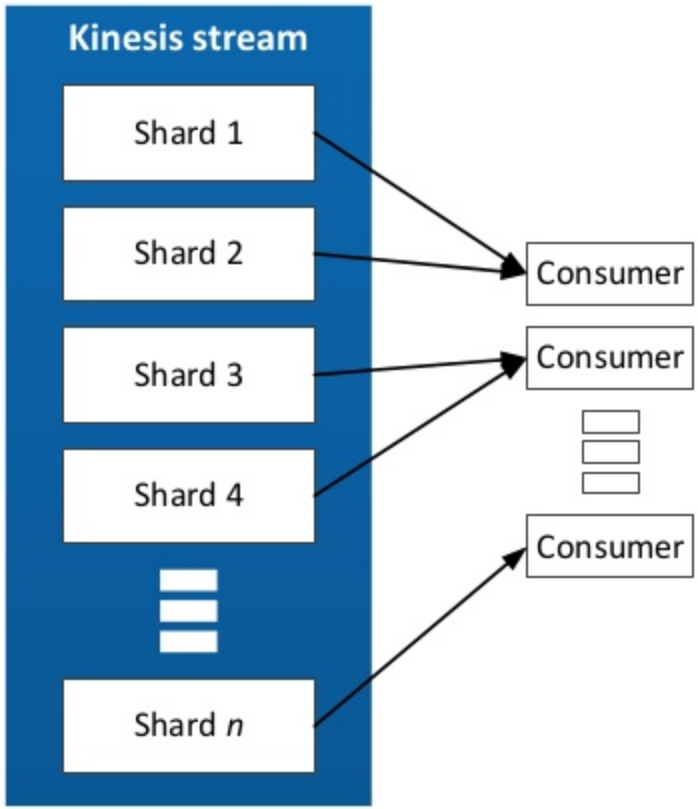
Connectivity – producers are not always connected, need to send data with low latency

Durability – producers have limited or no local storage, need to get data elsewhere quickly or its lost

Efficiency – producers primary job is not data collection, need low overhead for sending data

Distributed – large number of producers, need to receive and maintain order across different keys

Getting Data from a Kinesis stream



- Each shard is polled continuously using **GetRecords**, determine where to start using **GetSharIterator**
- GetRecords {Limit, ShardIterator}
- GetShardIterator {StreamName, ShardId, ShardIteratorType, StartingSequenceNumber, Timestamp}
- Options: Kinesis Client Library (KCL) on EC2, AWS Lambda, Spark Streaming (EMR), Storm on EC2
- (Almost) All solutions use the KCL under the hood

Why Amazon Kinesis Client Library?

- Open source client library available for Java, Ruby, Python, Node.JS dev
- Deploy on your EC2 instances, scales easily with Elastic Beanstalk
- Manages consumer to shard mapping and checkpoints
- KCL Application includes three components:
  1. **Record Processor Factory** – Creates the record processor
  2. **Record Processor** – Processor unit that processes data from a shard in Amazon Kinesis Streams
  3. **Worker** – Processing unit that maps to each application instance

Streaming Architecture Workflow: Lambda + Kinesis

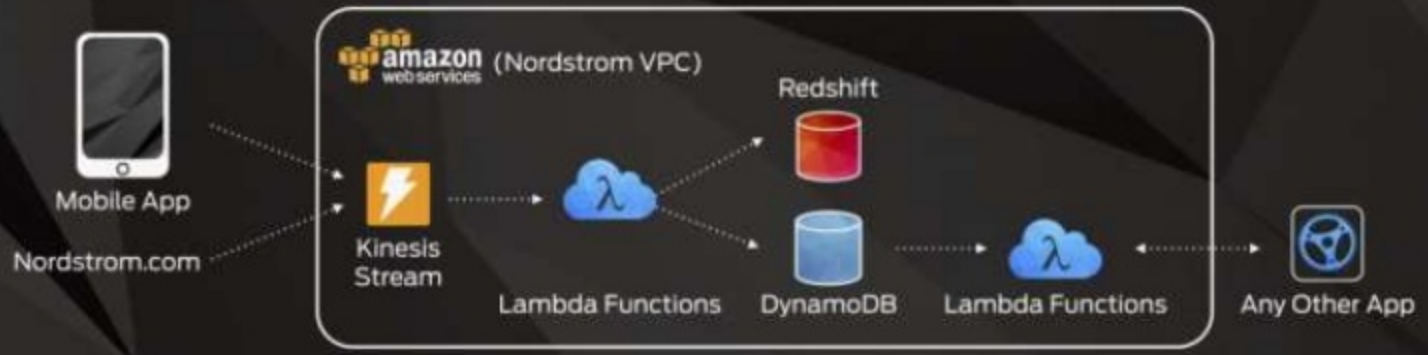
Data Input	Amazon Kinesis	Action	Lambda	Data Output
IT application activity	Capture the stream	Audit	Process the stream	SNS
Metering records		Condense		Amazon Redshift
Change logs		Backup		S3
Financial data		Store		RDS
Transaction orders		Process		SQS
Server health metrics		Monitor		EC2
User clickstream		Analyze		EMR
IoT device data		Respond		Backend endpoint
Custom data		Custom action		Custom application

Nordstrom Online Stylist

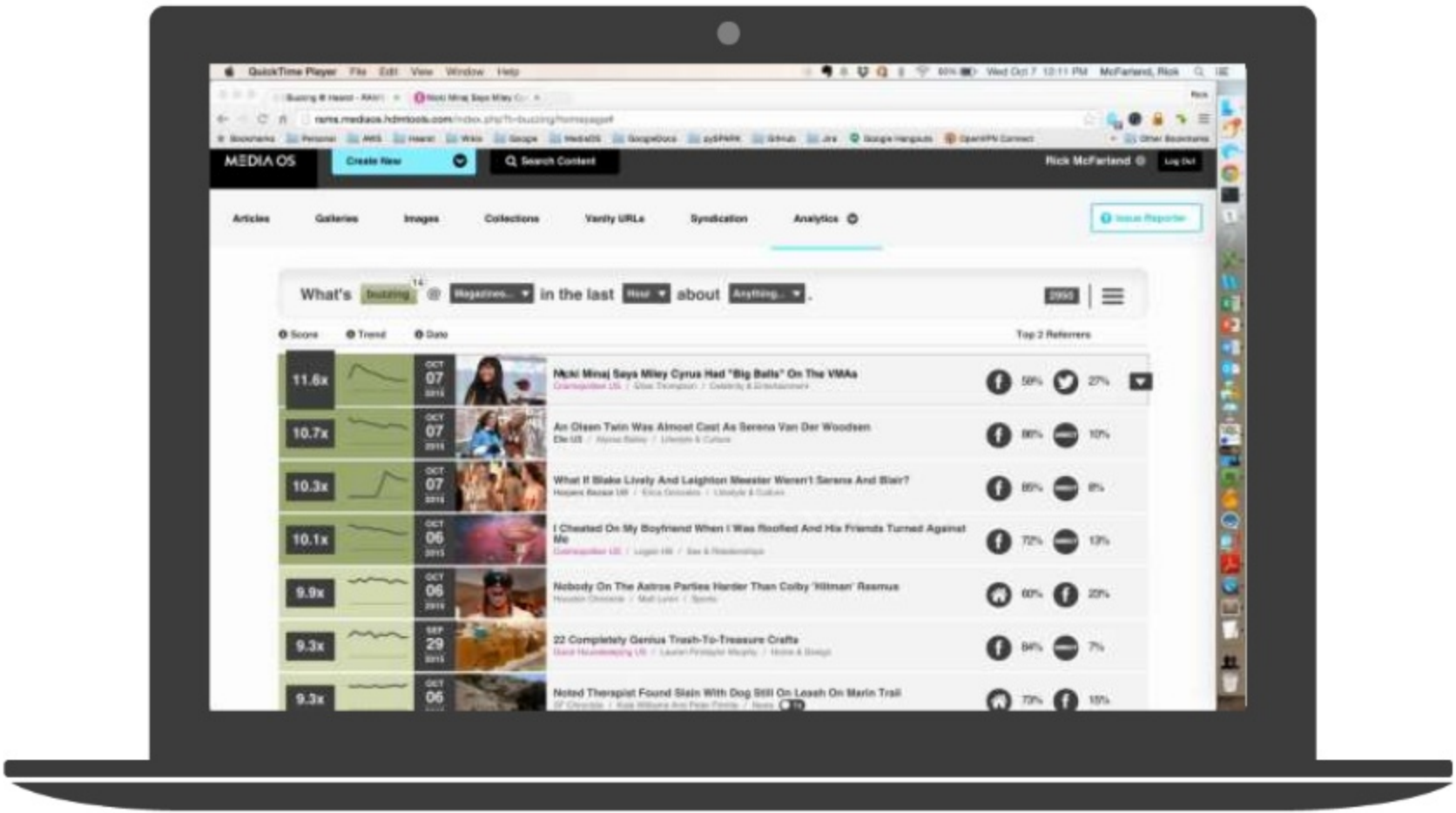
Nordstrom Recommendation

15-20 minute of processing into seconds

2x order of magnitude for cost savings



Let’s start with a product that empowers our Data Creators – Buzzing@Hearst



HEARST technology

Hearst's Serverless Data Pipeline