Processing Big Data in Motion

Streaming Data Ingestion and Processing

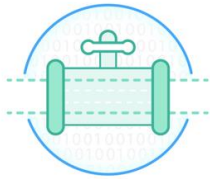
Roger Barga
General Manager
Kinesis Streaming Services, AWS

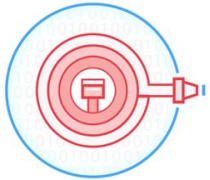
June 24th, 2016

amazon web services

Riding the Streaming Rapids

Microsoft Research





Streaming Map Reduce
& Machine Learning over Streams

Azure Stream Analytics

Relational Semantics
and Implementation

Complex Event Processing
over Streaming Data

DEBS Keynote 2013

2007 & 2008 2009 2010 2011 2012 2013 2015 2016

Interest in and *demand* for stream data processing is *rapidly increasing*[*] …

[*] *Understatement of the year (credit to Kostas Tzoumas)…*

Most data is *produced continuously*

127.0.0.1 user-identifier frank [10/Oct/2000:13:55:36 -0700] "GET /apache_pb.gif HTTP/1.0" 200 2326

Common Log Entry

Smart Textiles

Beacons

<R,AMZN,T,G,R1>

NASDAQ OMX Record

"SeattlePublicWater/Kinesis/123/Realtime" – 412309129140

MQTT Record

Health Monitors

Smart Buildings

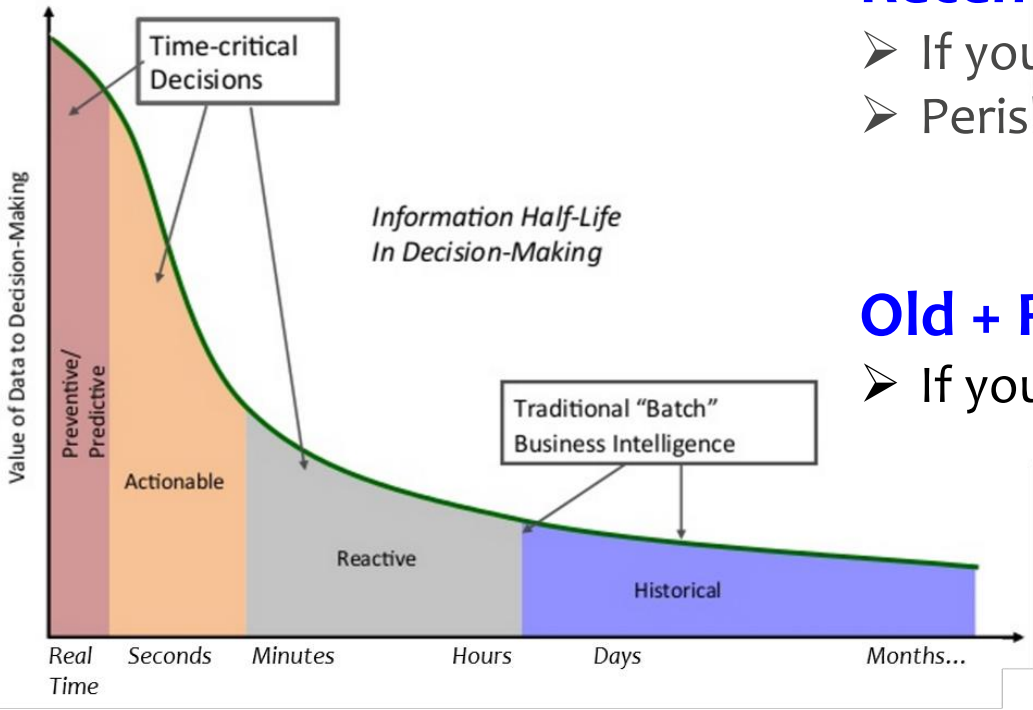
```
{
  "payerId": "Joe",
  "productCode": "AmazonS3",
  "clientProductCode": "AmazonS3",
  "usageType": "Bandwidth",
  "operation": "PUT",
  "value": "22490",
  "timestamp": "1216674828"
}
```

Metering Record

<165>1 2003-10-11T22:14:15.003Z mymachine.example.com evntslog - ID47 [exampleSDID@32473 iut="3" eventSource="Application" eventID="1011"][examplePriority@32473 class="high"]

Syslog Entry

Time is money…



Recent data is highly valuable

- If you act on it in time
- Perishable Insights (M. Gualtieri, Forrester)

Old + Recent data is more valuable

- If you have the means to combine them

Disruptive

Most 'big data' (Hadoop) jobs process data that was continuously generated

Foundational for business

Enable new class of applica

that process data continuo

Agenda

- Scalable & Durable Data Ingest
  - A quick word on our motivation
  - Kinesis Streams, through a simple example
- Continuous Stream Data Processing
  - Kinesis Client Library (KCL)
  - How customers are using Kinesis Streams today
- Building on Kinesis Streams
  - Kinesis Firehose
  - Kinesis Analytics

Our Motivation for Continuous Processing

AWS Metering service

- 100s of millions of billing records per second
- Terabytes^{++} per hour
- Hundreds of thousands of sources
- For each customer: gather all metering records & compute monthly bill
- Auditors guarantee 100% accuracy at months end

Seem perfectly reasonable to run as a batch, but relentless pressure for realtime…

With a Data Warehouse to load

- 1000s extract-transform-load (ETL) jobs every day
- Hundreds of thousands of files per load cycle
- Thousands of daily users, hundreds of queries per hour

Our Motivation for Continuous Processing

AWS Metering service

- 100s of millions of billing records per second
- Terabytes^{++} per hour
- Hundreds of thousands of sources
- For each customer: gather all metering records & compute monthly bill
- Auditors guarantee 100% accuracy at months end

Other Service Teams, Similar Requirements

- CloudWatch Logs and CloudWatch Metrics
- CloudFront API logging
- ‘Snitch’ internal datacenter hardware metrics

Right Tool for the Job

Enable Streaming Data Ingestion and Processing

Real-time Ingest

- Highly Scalable
- Durable
- Replayable Reads

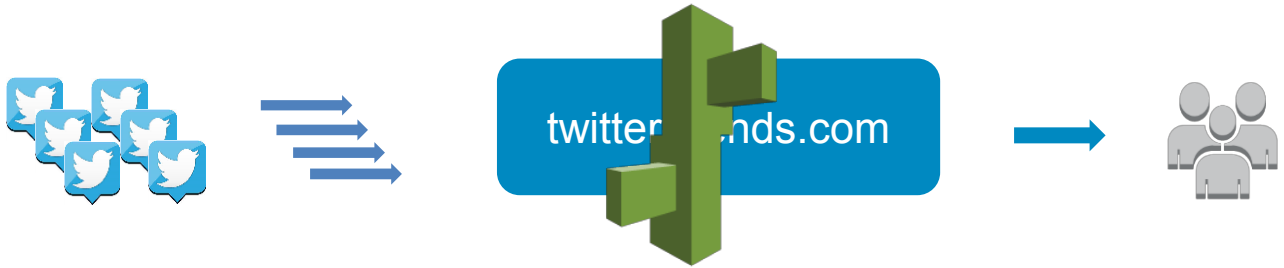
Continuous Processing

- Support multiple simultaneous data processing applications
- Load-balancing incoming streams, scale out processing
- Fault-tolerance, Checkpoint / Replay

Amazon Kinesis

Example application

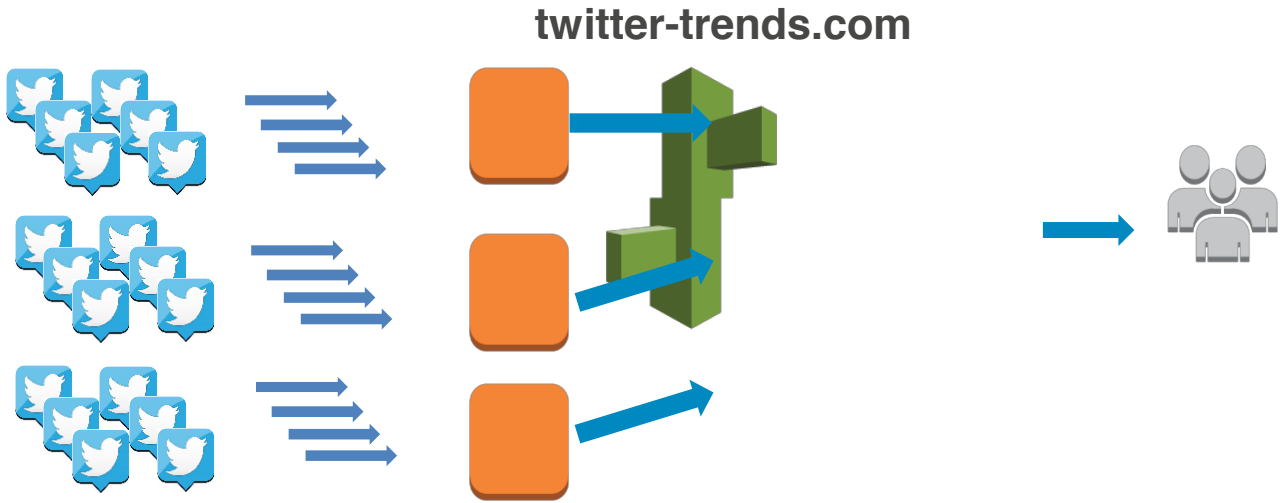
twitter-trends.com website



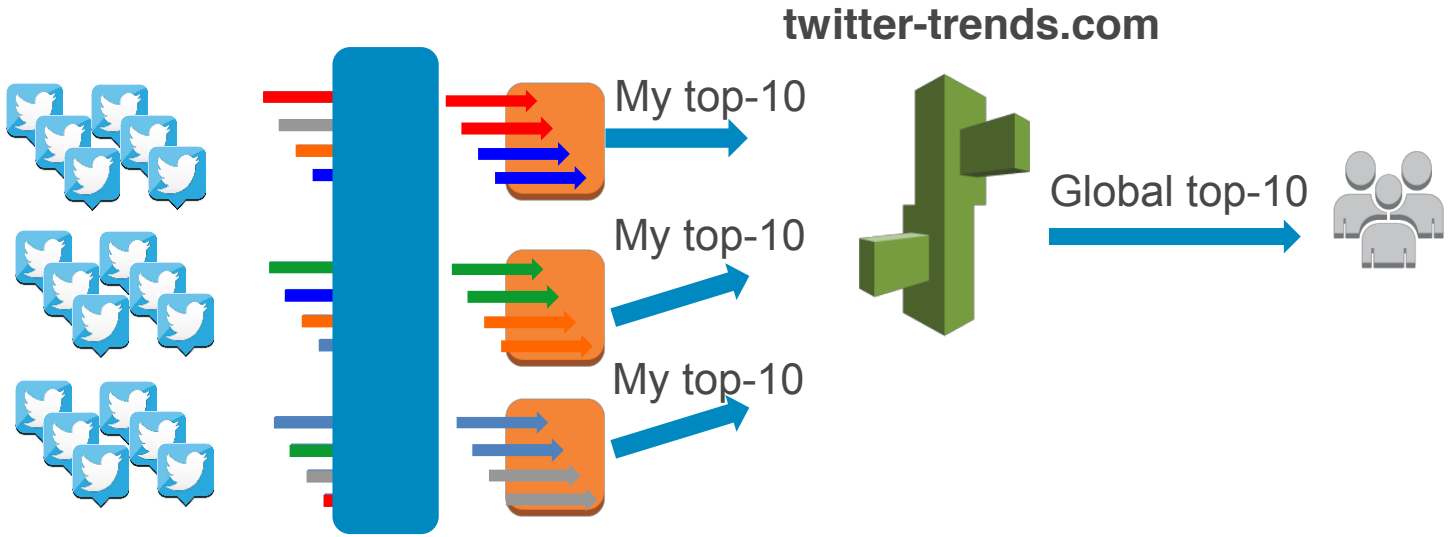
Elastic Beanstalk

twitter-trends.com

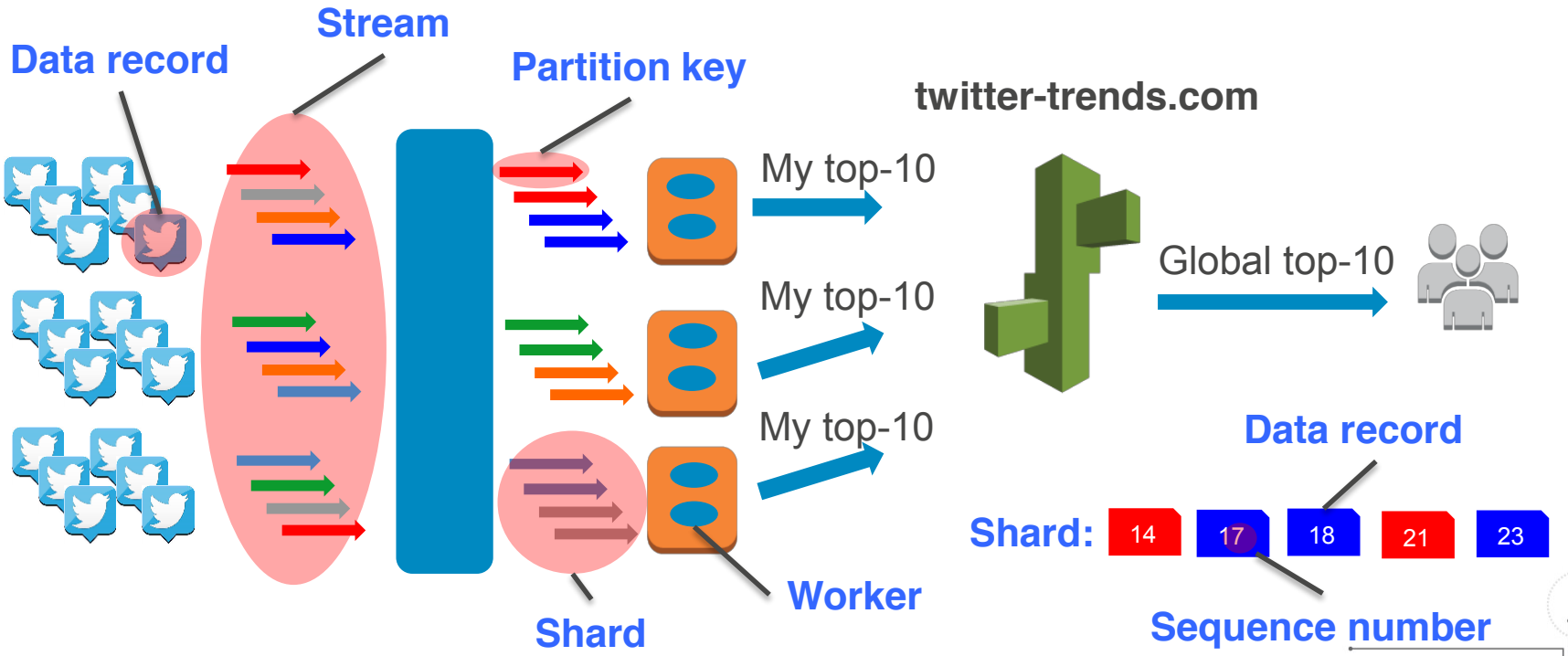
Too big to handle on one box



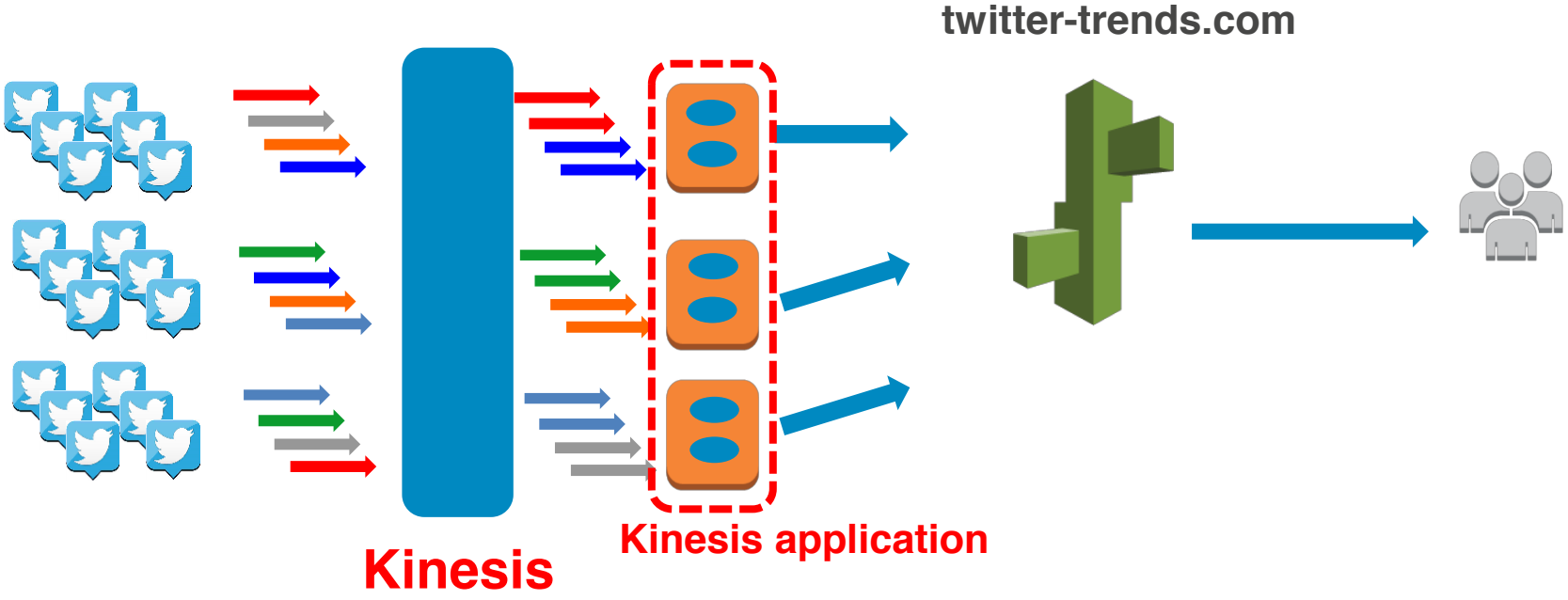
The solution: streaming map/reduce



Core concepts

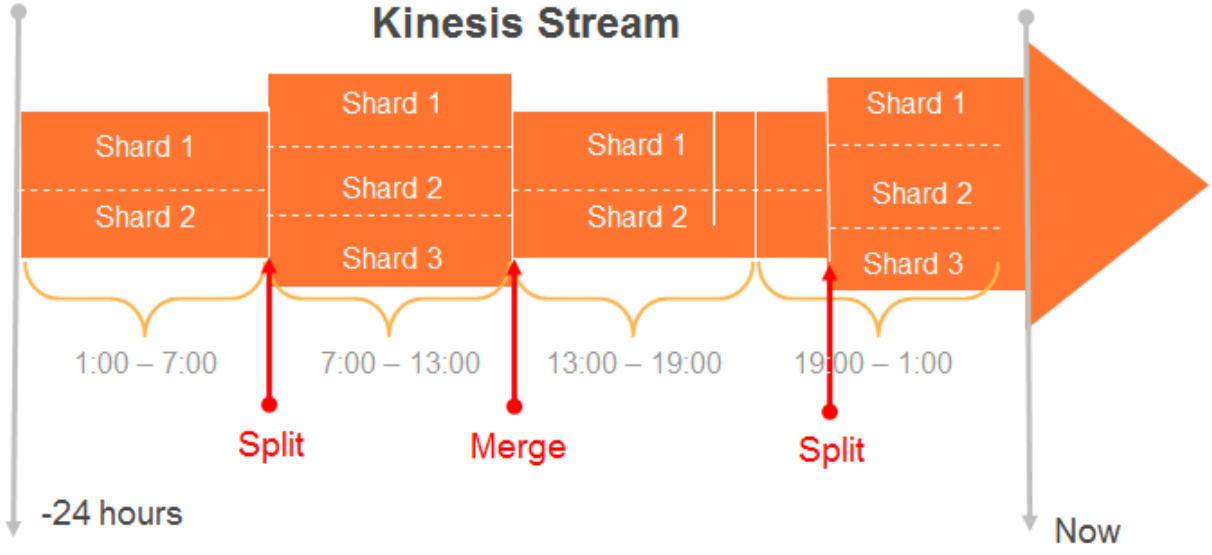


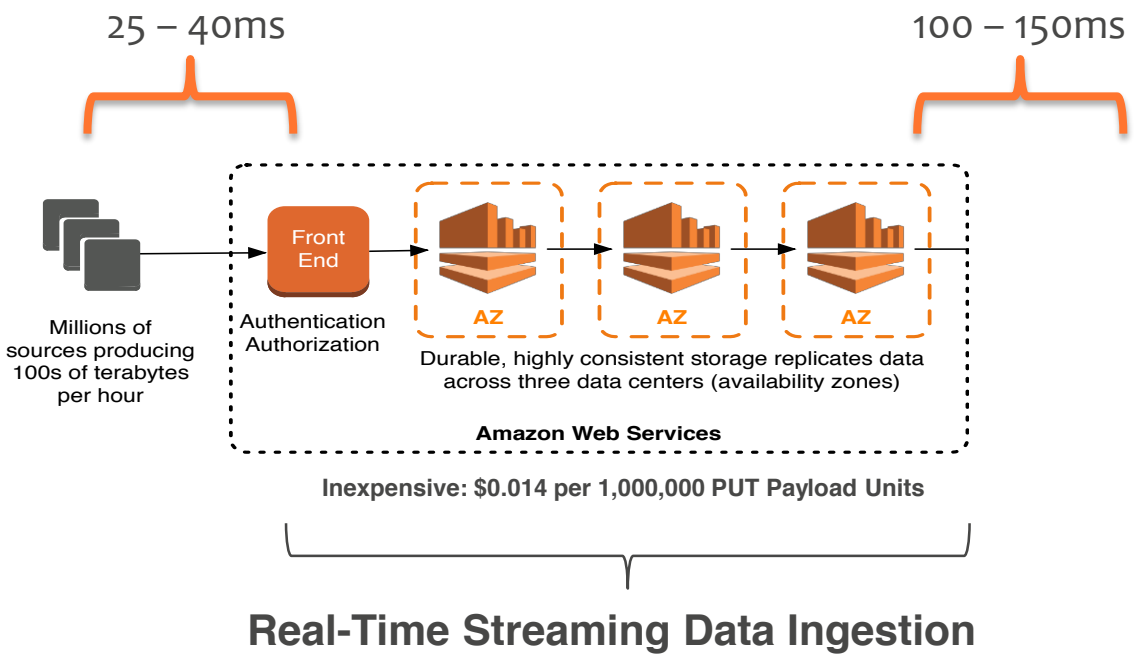
How this relates to Kinesis



Kinesis Streaming Data Ingestion

- Streams are made of **Shards**
- Each Shard ingests data up to 1MB/sec, and up to 1000 TPS
- Producers use a PUT call to store data in a Stream: PutRecord {Data, PartitionKey, StreamName}
- Each Shard emits up to 2 MB/sec
- All data is stored for **24 hours**, **7 days** if extended retention is 'ON'
- **Scale** Kinesis streams by adding or removing Shards
- **Replay** data from retention period

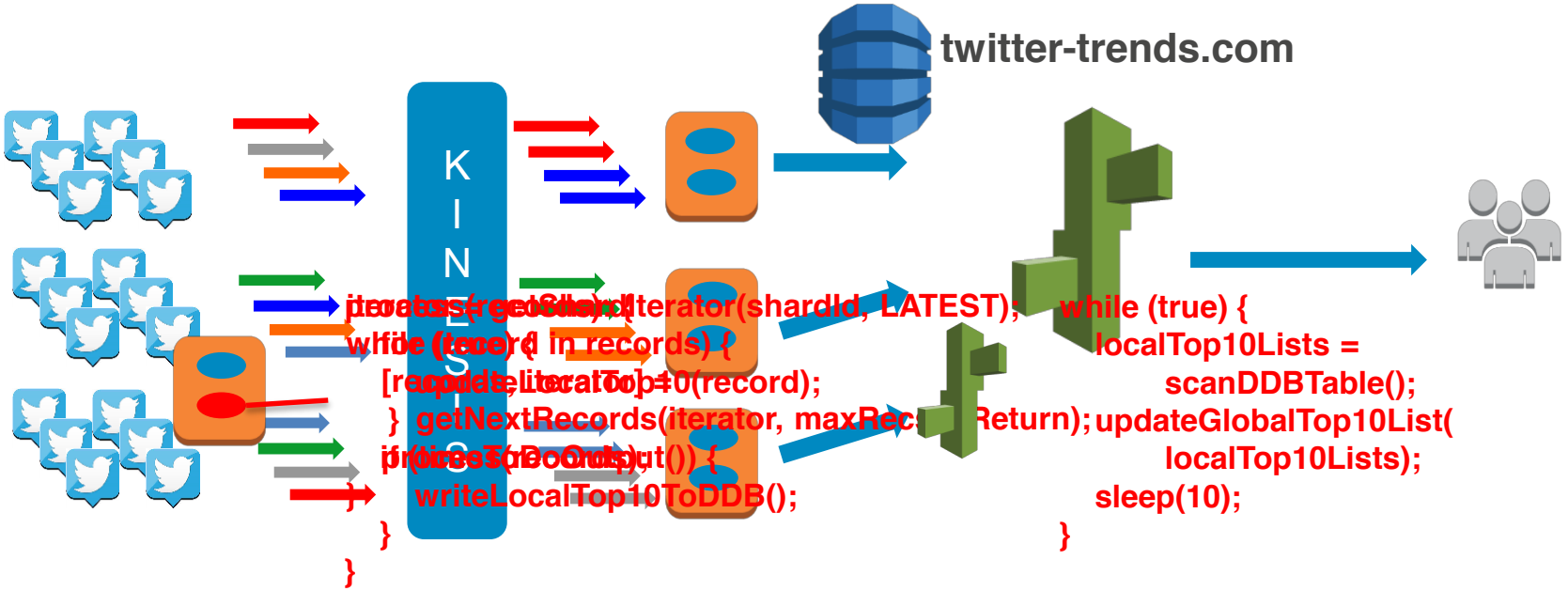




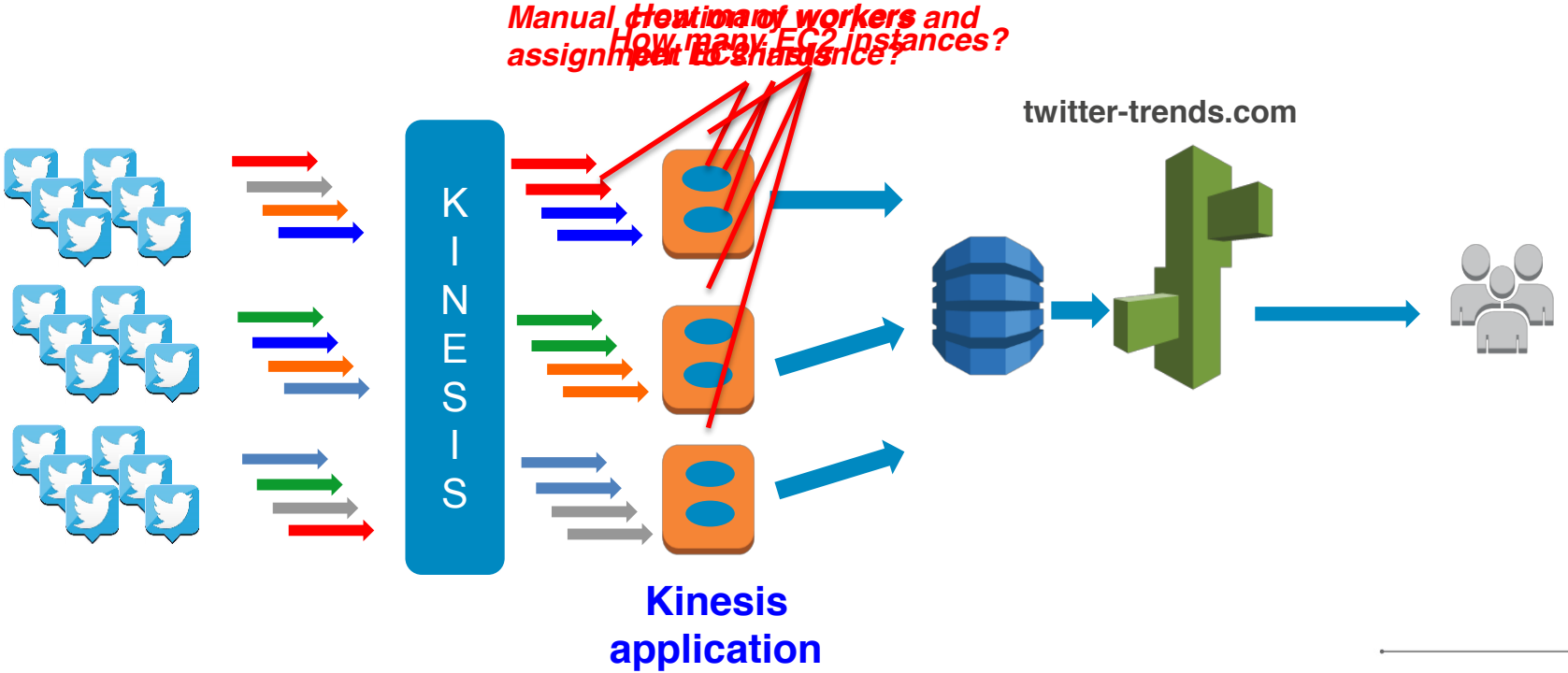
Real-Time Streaming Data Ingestion

Kinesis Client Library

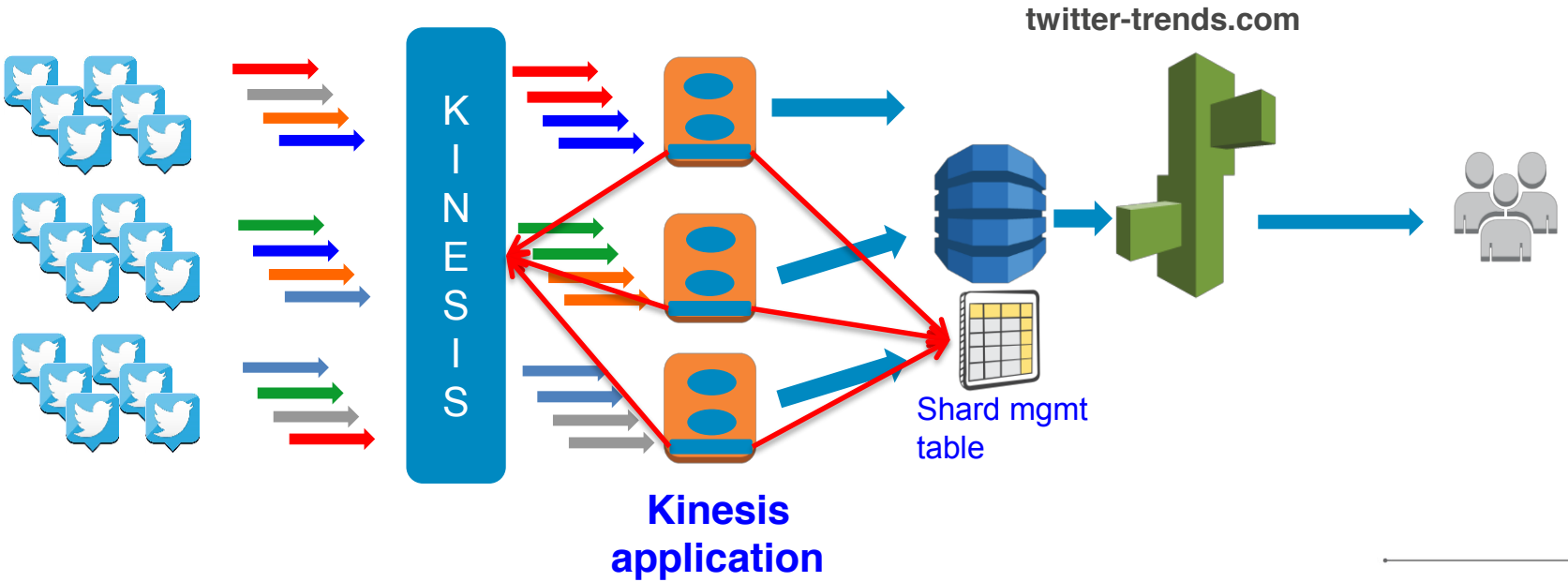
Using the Kinesis API directly



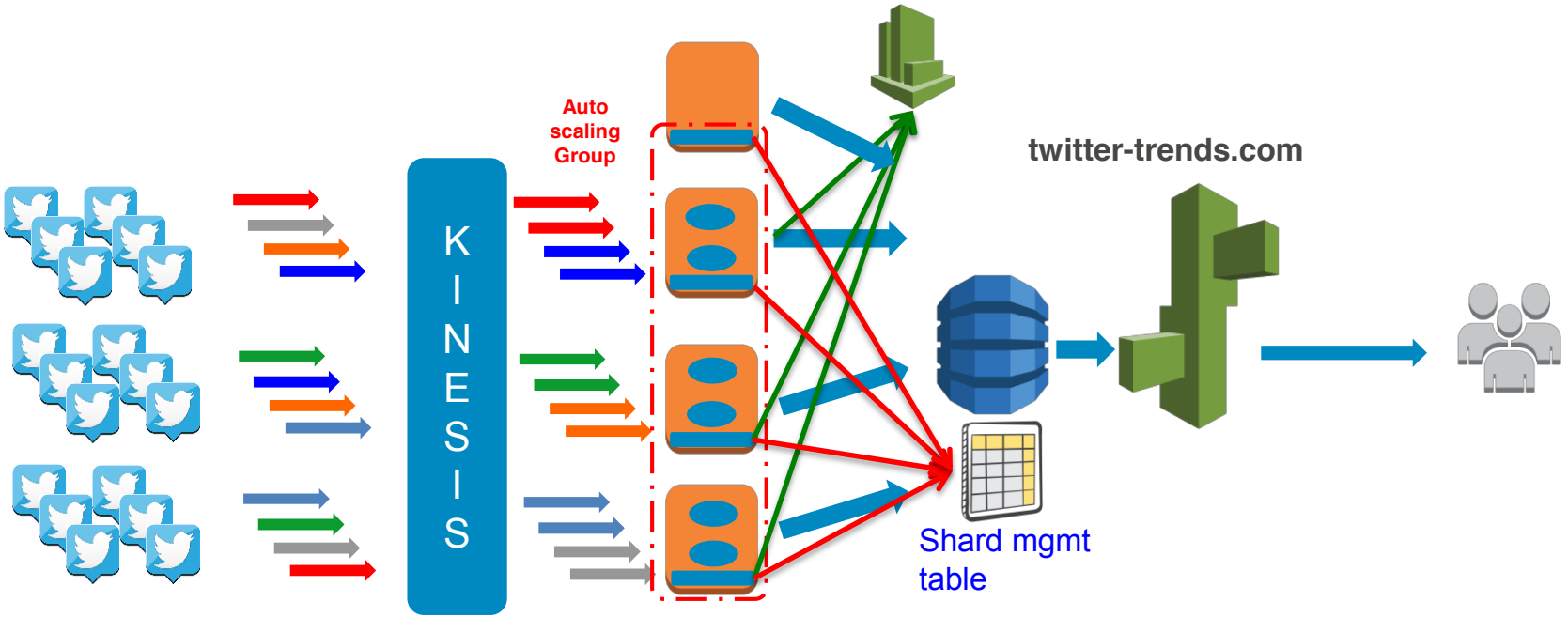
Challenges with using the Kinesis API directly



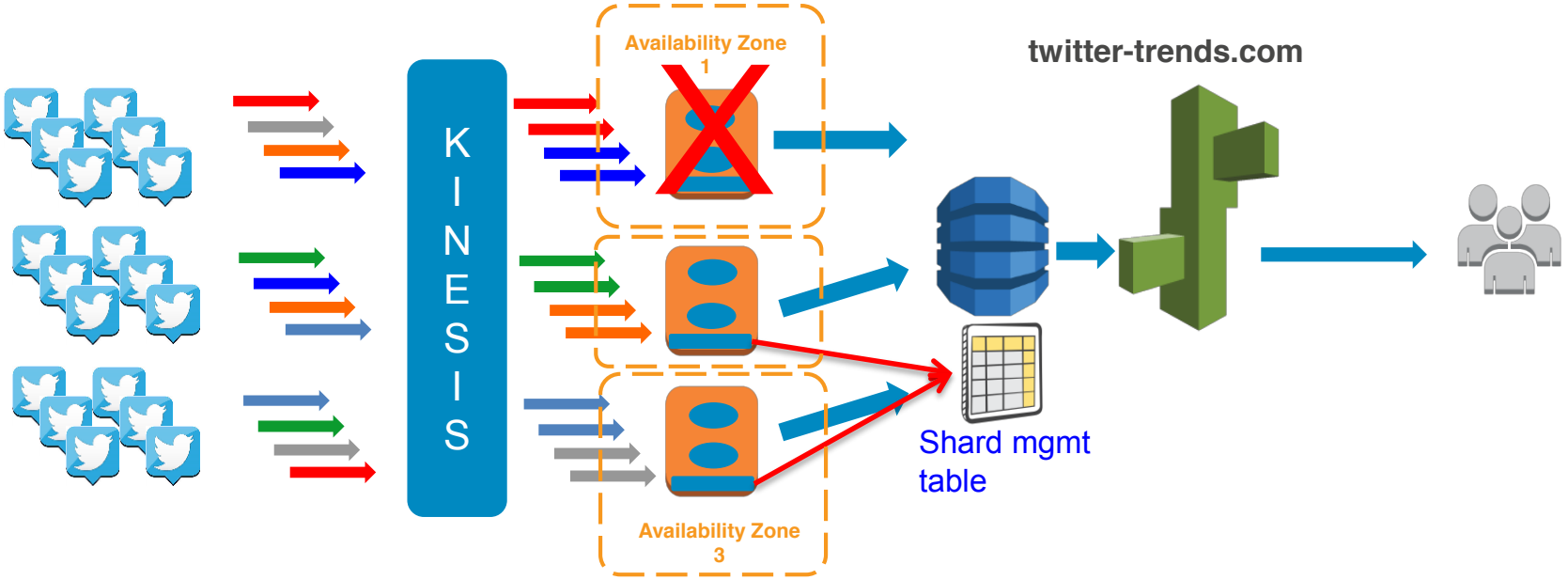
Using the Kinesis Client Library



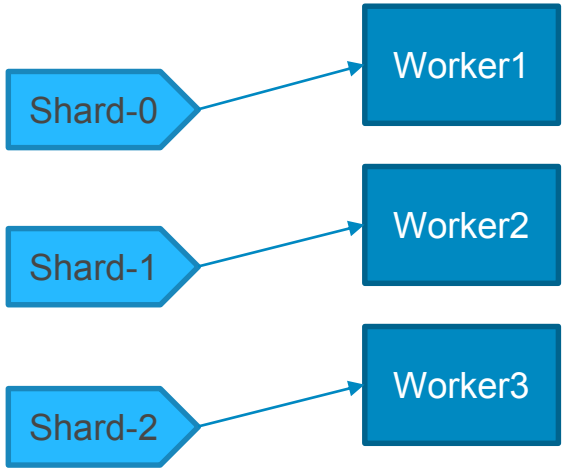
Elasticity and Load Balancing



Fault Tolerance Support

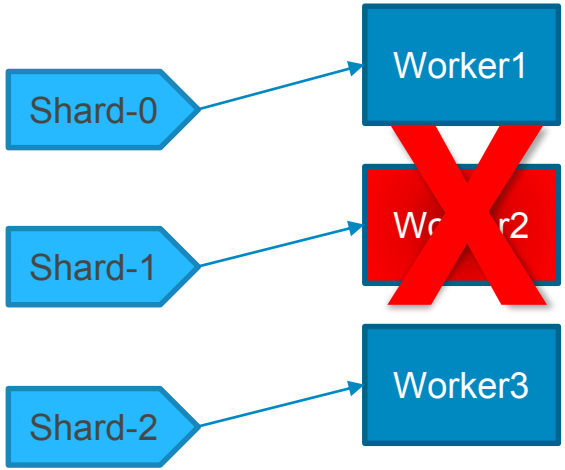


Worker Fail Over



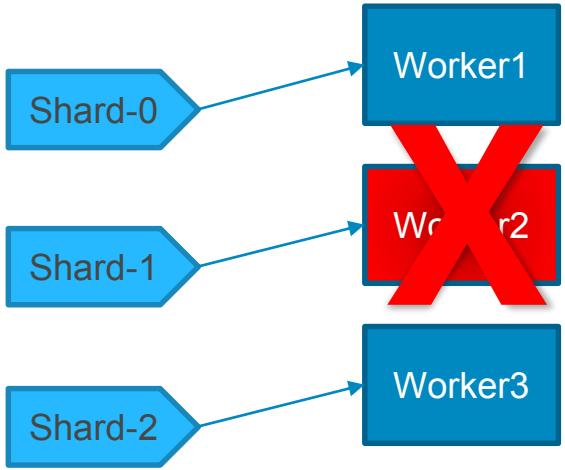
LeaseKey	LeaseOwner	LeaseCounter
Shard-0	Worker1	85
Shard-1	Worker2	94
Shard-2	Worker3	76

Worker Fail Over



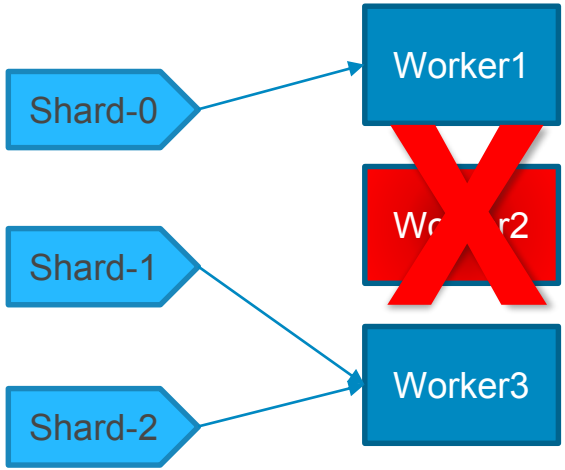
LeaseKey	LeaseOwner	LeaseCounter
Shard-0	Worker1	~~85~~ 86
Shard-1	Worker2	94
Shard-2	Worker3	~~76~~ 77

Worker Fail Over



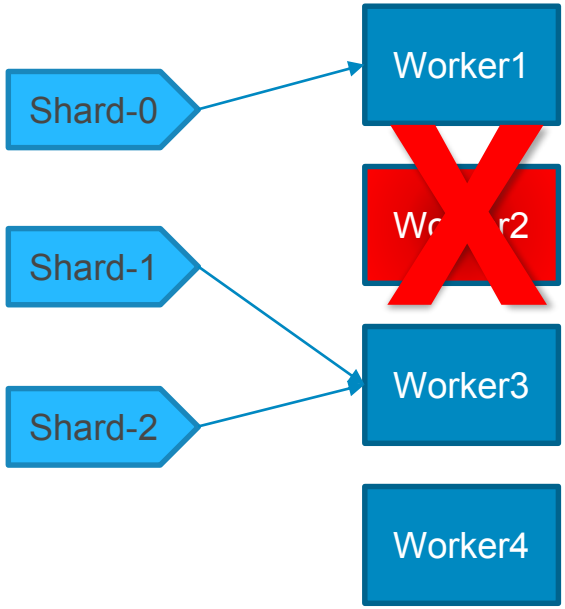
LeaseKey	LeaseOwner	LeaseCounter
Shard-0	Worker1	~~85~~ ~~86~~ 87
Shard-1	Worker2	94
Shard-2	Worker3	~~76~~ ~~77~~ 78

Worker Fail Over



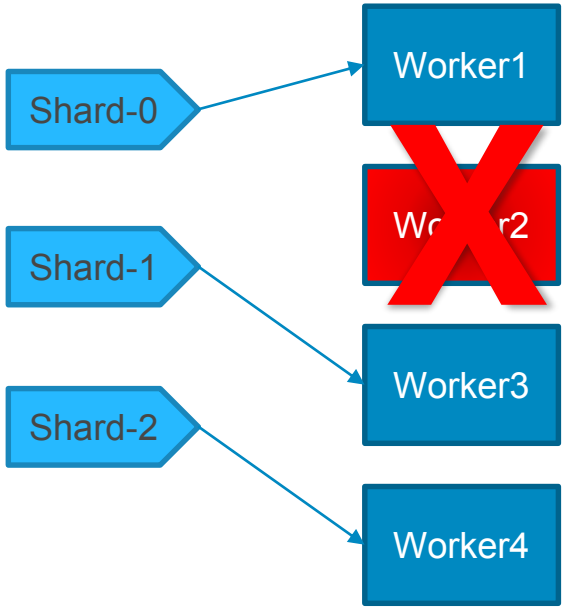
LeaseKey	LeaseOwner	LeaseCounter
Shard-0	Worker1	~~85~~ ~~86~~ ~~87~~ 88
Shard-1	Worker3	~~94~~ 95
Shard-2	Worker3	~~76~~ ~~77~~ ~~78~~ 79

Worker Load Balancing



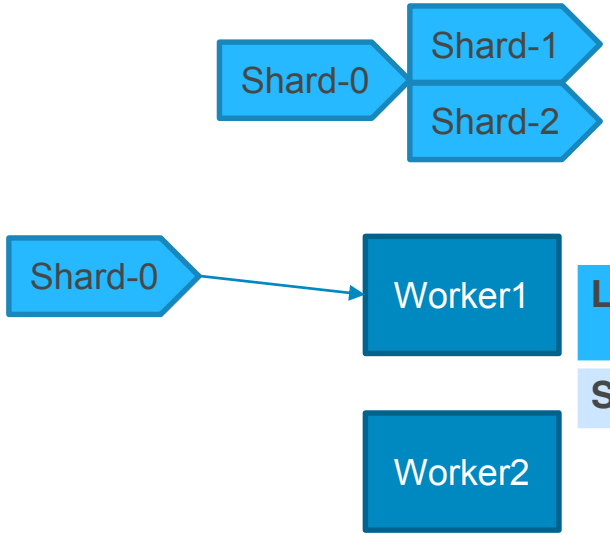
LeaseKey	LeaseOwner	LeaseCounter
Shard-0	Worker1	88
Shard-1	Worker3	96
Shard-2	Worker3	78

Worker Load Balancing



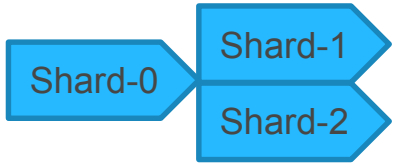
LeaseKey	LeaseOwner	LeaseCounter
Shard-0	Worker1	88
Shard-1	Worker3	96
Shard-2	Worker4	79

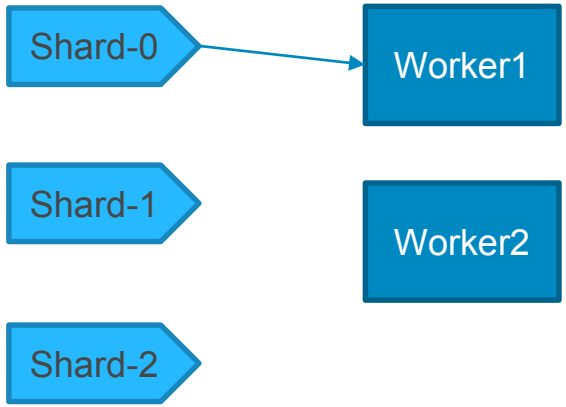
Resharding



LeaseKey	LeaseOwner	LeaseCounter	checkpoint
Shard-0	Worker1	90	SHARD_END

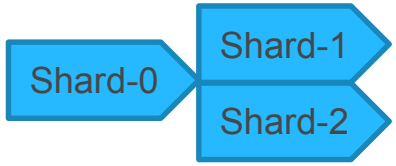
Resharding

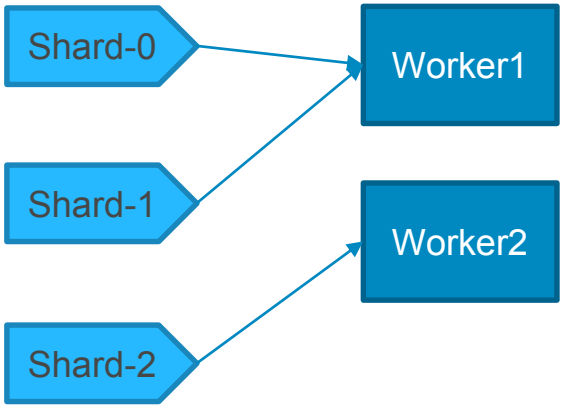




LeaseKey	LeaseOwner	LeaseCounter	checkpoint
Shard-0	Worker1	90	SHARD_END
Shard-1		0	TRIM_HORIZON
Shard-2		0	TRIM_HORIZON

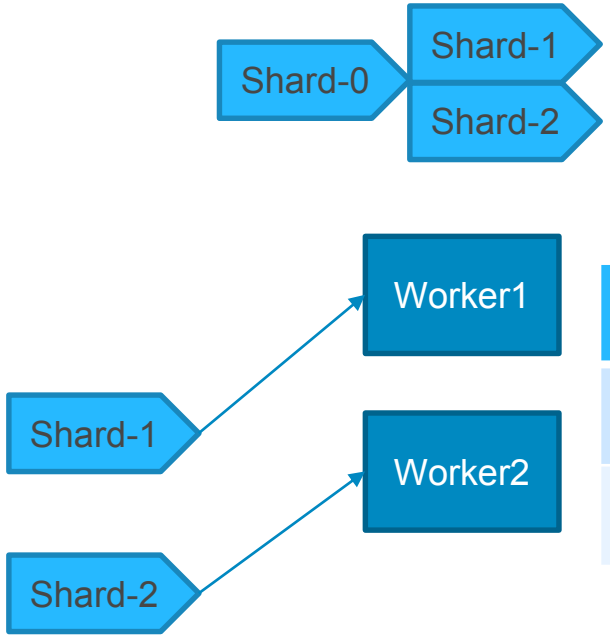
Resharding





LeaseKey	LeaseOwner	LeaseCounter	checkpoint
Shard-0	Worker1	90	SHARD_END
Shard-1	Worker1	2	TRIM_HORIZON
Shard-2	Worker2	3	TRIM_HORIZON

Resharding



LeaseKey	LeaseOwner	LeaseCounter	checkpoint
Shard-1	Worker1	2	TRIM_HORIZON
Shard-2	Worker2	3	TRIM_HORIZON

Putting this into production

Cost & Scale

500MM tweets/day = ~ 5,800 tweets/sec

2k/tweet is ~12MB/sec (~1TB/day)

$0.015/hour per shard, $0.014/million PUTS

Kinesis cost is $0.47/hour

Redshift cost is $0.850/hour (for a 2TB node)

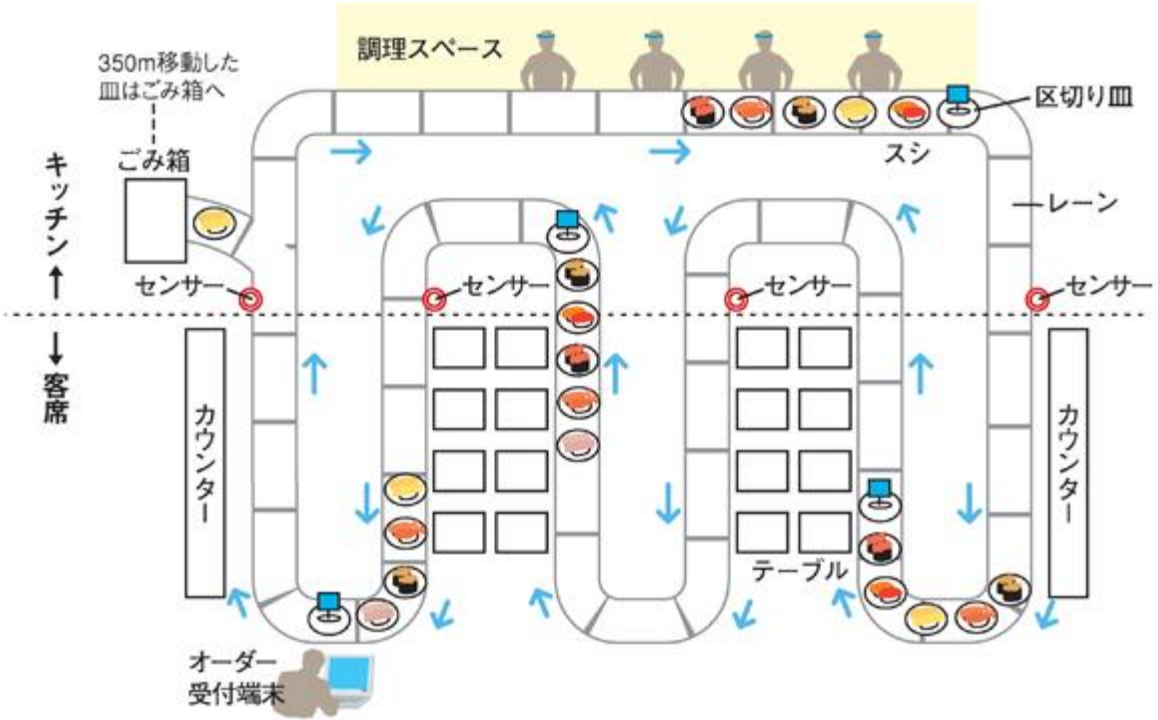
Total: $1.32/hour

Design Challenge(s)

- Dynamic Resharding & Scale Out
- Enforcing Quotas (think proxy fleet with 1Ks servers)
- Distributed Denial of Service Attack (unintentional)
- Dynamic Load Balancing on Storage Servers
- Heterogeneous Workloads (tip of stream vs 7 day)
- Optimizing Fleet Utilization (proxy, control, data planes)
- Avoid Scaling Cliffs
- …

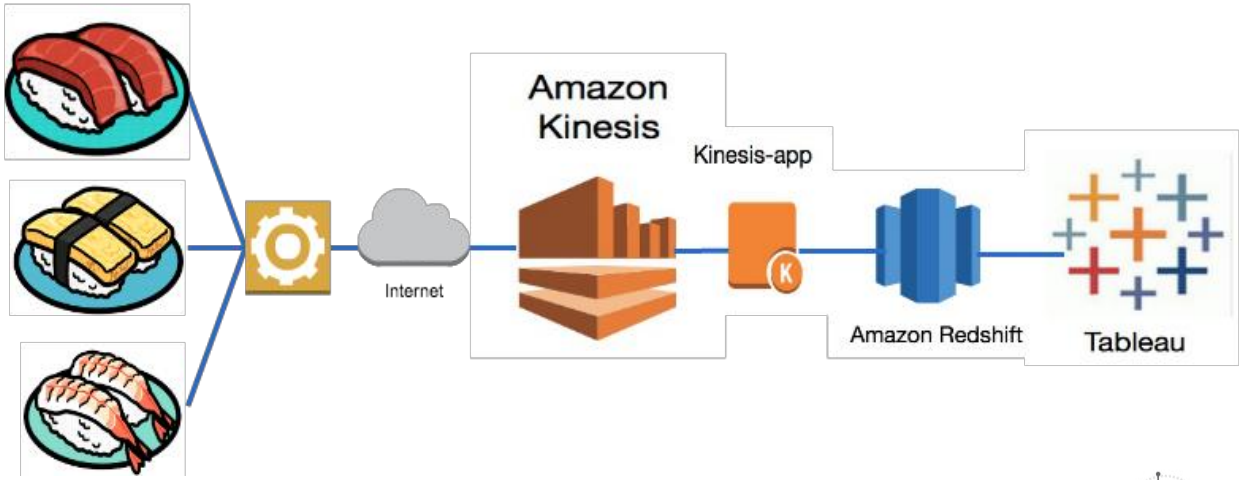
Sushiro: Kaiten Sushi Restaurants

380 stores stream data from sushi plate sensors and stream to Kinesis



Sushiro: Kaiten Sushi Restaurants

380 stores stream data from sushi plate sensors and stream to Kinesis



Real-Time Streaming Data with Kinesis Streams

1 TB+/day game data analyzed in real-time I **Gaming**

5 billion events/wk from connected devices I **IoT**

17 PB of game data per season I **Entertainment**

100 billion ad impressions/day, 30 ms response time I **Ad Tech**

100 GB/day click streams 250+ sites I **Enterprise**

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

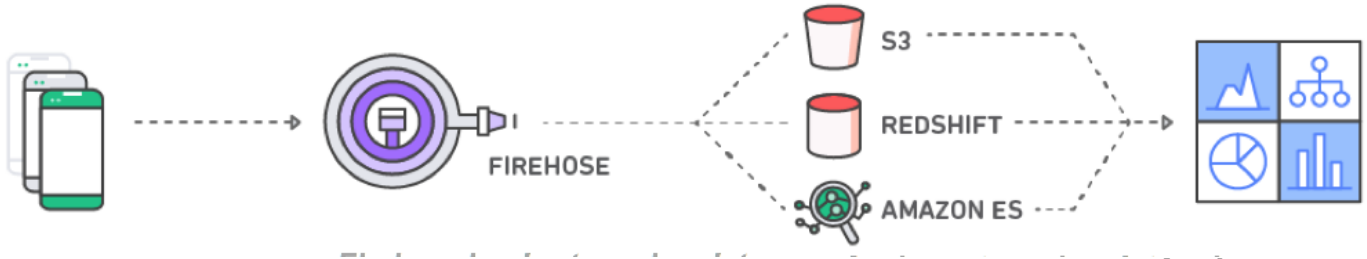
17 million events/day I **Technology**

1 billion transactions per day I **Bitcoin**

Streams provide a foundational abstraction on which to build higher level services

Amazon Kinesis Firehose

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



Capture and submit streaming data

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

Analyze streaming data using your favorite BI tools

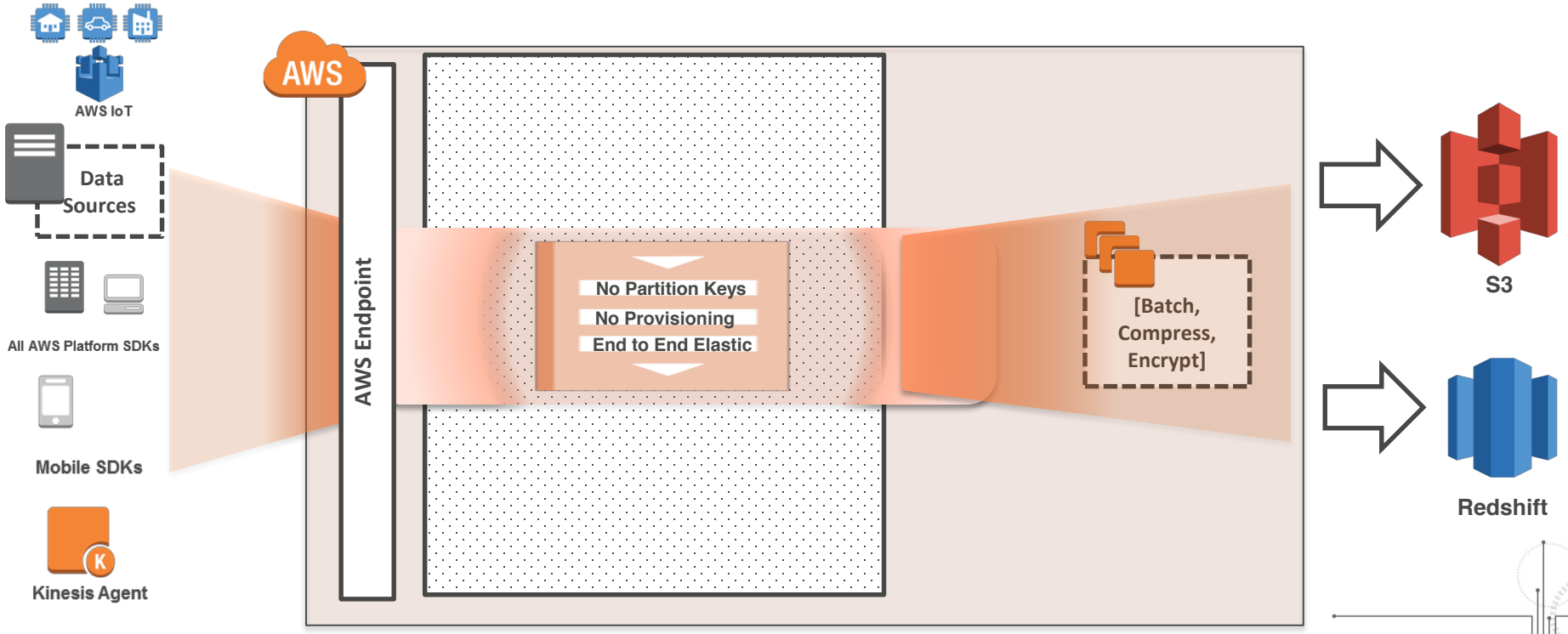
Zero administration: Capture and deliver streaming data into Amazon S3, Amazon Redshift, and other destinations **without writing an application or managing infrastructure.**

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

Amazon Kinesis Firehose

Fully Managed Service for Delivering Data Streams into AWS Destinations



Amazon Kinesis Analytics

Analyze data streams continuously with standard SQL

Connect to Kinesis streams, Firehose delivery streams

Run standard SQL queries against data streams

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 a Kinesis Stream or Firehose Delivery Stream and apply ANSI standard SQL.
- **Build real-time applications:** Perform continual processing on streaming data with sub-second processing latencies
- **Easy Scalability :** Elastically scales to match data throughput

Realtime Analytics Patterns

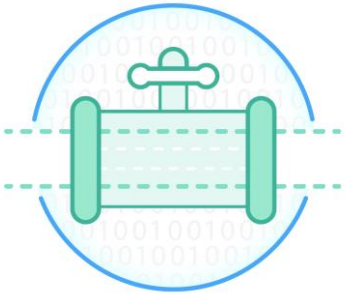
- Simple counting (e.g. failure count)
- Counting with Windows (e.g. failure count every hour)
- Preprocessing: filtering, transformations (e.g. data cleanup)
- Alerts , thresholds (e.g. alarm on high temperature)
- Data Correlation, Detect missing events, detecting erroneous data (e.g. detecting failed sensors)
- Joining event streams (e.g. detect a hit on soccer ball)
- Merge with data in database, collect, update data conditionally

Realtime Analytics Patterns (contd.)

- Detecting Event Sequence Patterns (e.g. small transaction followed by large transaction)
- Tracking - follow some related entity's state in space, time etc. (e.g. location of airline baggage, vehicle, customer by beacon)
- Detect trends – Rise, turn, fall, outliers, complex trends like triple bottom etc., (e.g. algorithmic trading, SLA, load balancing).

Amazon Kinesis: Streaming data made easy

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



Kinesis Streams

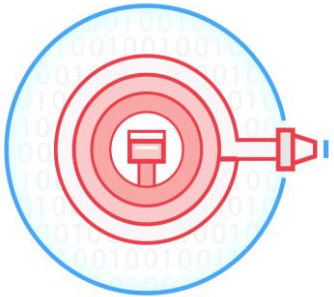
For Technical Developers

Build your own custom application to process or analyze streaming data

Kinesis Analytics

For all developers, analysts and data scientists

Easily analyze streaming data using standard SQL



Kinesis Firehose

For all developers, data scientists, IT professionals

Transform and load streaming data into S3, Redshift, Elasticsearch, and more…

Stream Processing End2End

Kinesis Analytics

Transform

- Extract Fields
- Clean
- Enrich

Analyze

- Filter
- Temporal joins
- Combine w/ reference data
- Projections
- Correlate
- Windowed Aggregates
- Anomaly Detection
- …

Durable ingest, repeatable processing → *In stream processing* → *low latency delivery to persist, alert, visualize*

IoT Sensors (Example: Hello Inc.)

Sleep monitoring devices send data like bedroom temperature, humidity, ambient light, noise level, and particulate count

Kinesis Streams reliably collects, stores, and exposes sensor data for processing

Firehose loads data into S3 and Redshift for data science and durable storage

Consumers get better sleep by monitoring and adjusting their sleeping conditions

Customer Clickstream

Send clickstream data to Kinesis Streams

Kinesis Streams stores and exposes clickstream data for processing

Firehose loads data into S3 and Redshift for data science and durable storage

Readers see personalized content suggestions and engage more

amazon web services

Closing Thoughts

Streaming data is highly prevalent and relevant;

Stream data processing is on the rise;

A key part of business critical workflows today, a powerful abstraction for building a new class of applications & data intensive services tomorrow.

A rich area for distributed systems, programming model, IoT, and new service(s) research.

Questions