Analyzing Real-time Streaming Data with Amazon Kinesis

Ken Payne, AWS Solutions Architect

28 June 2017
Key takeaways from this session:

• Amazon Kinesis makes it easy to collect, process, and analyze real-time, streaming data.

• Amazon Kinesis offers key capabilities to cost effectively process streaming data at any scale, along with the flexibility to choose the tools that best suit the requirements of your application.

• Customer Story: Palringo
Why Streaming Data?
Most data is produced continuously

- Mobile Apps
- Web Clickstream
- Application Logs
- Metering Records
- IoT Sensors
- Smart Buildings

## Streaming Data Scenarios Across Verticals

<table>
<thead>
<tr>
<th>Scenarios/Verticals</th>
<th>Accelerated Ingest-Transform-Load</th>
<th>Continuous Metrics Generation</th>
<th>Machine Learning and Actionable Insights</th>
</tr>
</thead>
<tbody>
<tr>
<td>Digital Ad Tech/Marketing</td>
<td>Publisher, bidder data aggregation</td>
<td>Advertising metrics like coverage, yield, and conversion</td>
<td>User engagement with ads, optimized bid/buy engines</td>
</tr>
<tr>
<td>IoT</td>
<td>Sensor, device telemetry data ingestion</td>
<td>Operational metrics and dashboards</td>
<td>Device operational intelligence &amp; alerts</td>
</tr>
<tr>
<td>Gaming</td>
<td>Online data aggregation, e.g., top 10 players</td>
<td>Massively multiplayer online game (MMOG) live dashboards</td>
<td>Leader board generation, player-skill match</td>
</tr>
<tr>
<td>Consumer Online</td>
<td>Clickstream analytics</td>
<td>Metrics like impressions and page views</td>
<td>Recommendation engines, proactive care</td>
</tr>
<tr>
<td>Operation Security</td>
<td>DevOps tools, ingesting VPC flow logs</td>
<td>Subscribe to CloudWatch Logs and analyze logs in real time</td>
<td>Anomaly detection</td>
</tr>
</tbody>
</table>
Processing real-time, streaming data

What are the key requirements?

- Durable
- Continuous
- Fast

- Correct/Ordered
- Reactive
- Reliable
What is Amazon Kinesis?
Amazon Kinesis: Streaming data done the AWS way
Makes it easy to capture, deliver, and process real-time data streams

Easy to provision, deploy, and manage

Elastically scalable & serverless

Real-time latencies

Pay as you go, no upfront costs

Right services for your specific use cases
Amazon Kinesis: Streaming data made easy. Services make it easy to capture, deliver and 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 all developers, data scientists
Easily load massive volumes of streaming data into S3, Amazon Redshift and Amazon Elasticsearch

**Amazon Kinesis Analytics**
For all developers, data scientists
Easily analyze data streams using standard SQL queries
Amazon Kinesis Customer Base Diversity

**Amazon Kinesis as databus**  
Migrated from Kafka to Kinesis | Enterprise

**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**

**100 GB/day click streams from 250+ sites | Enterprise**

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

**10 million events/day | Retail**

**Funnel all production events through Amazon Kinesis | High Tech**
Amazon Kinesis Streams 3rd Party Connectors
Amazon Kinesis Partners & SI
Amazon Kinesis Streams
Amazon Kinesis Streams
Build your own data streaming applications

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 Kinesis Client Library (KCL), Kinesis Analytics, Apache Spark/Storm, AWS Lambda, and more.

Low cost: Cost-efficient for workloads of any scale.
Sending and reading data from streams

Sending

AWS SDK
Kinesis
Flume
Fluentd

Sending

Consuming

Get* APIs
Kinesis Client Library + Connector Library
AWS Lambda
Amazon EMR
Apache
Storm
Apache
Spark

{  "StreamName": "string",  "PartitionKey": "string",  "Data": blob}
Scaling Amazon Kinesis Streams

- Streams are made of **Shards**
- Each Shard ingests data up to 1MB/sec, and up to 1000 TPS
- Each Shard emits up to 2 MB/sec
- All data is stored for **24 hours – 7 days**
- **Scale** Kinesis streams by splitting or merging Shards
- **Replay** data inside of 24Hr -7days Window
State Management with Kinesis Client Library

- One record processor maps to one shard and processes data records from that shard.
- One worker maps to one or more record processors.
- Balances shard-worker associations when worker / instance counts change.
- Balances shard-worker associations when shards split or merge.
Amazon Kinesis Streams with AWS Lambda
**AWS Lambda and Amazon Kinesis integration**

**How it Works**

- Multiple functions can be mapped to one stream
- Multiple streams can be mapped to one Lambda function
- Each mapping is a unique key pair: Kinesis stream to Lambda function
- Each mapping has unique shard iterators
Amazon Kinesis Firehose
## Streaming Data Scenarios Across Verticals

<table>
<thead>
<tr>
<th>Scenarios/Verticals</th>
<th>Accelerated Ingest-Transform-Load</th>
<th>Continuous Metrics Generation</th>
<th>Machine Learning and Actionable Insights</th>
</tr>
</thead>
<tbody>
<tr>
<td>Digital Ad Tech/Marketing</td>
<td>Publisher, bidder data aggregation</td>
<td>Advertising metrics like coverage, yield, and conversion</td>
<td>User engagement with ads, optimized bid/buy engines</td>
</tr>
<tr>
<td>IoT</td>
<td>Sensor, device telemetry data ingestion</td>
<td>Operational metrics and dashboards</td>
<td>Device operational intelligence &amp; alerts</td>
</tr>
<tr>
<td>Gaming</td>
<td>Online data aggregation, e.g., top 10 players</td>
<td>Massively multiplayer online game (MMOG) live dashboards</td>
<td>Leader board generation, player-skill match</td>
</tr>
<tr>
<td>Consumer Online</td>
<td>Clickstream analytics</td>
<td>Metrics like impressions and page views</td>
<td>Recommendation engines, proactive care</td>
</tr>
<tr>
<td>Operation Security</td>
<td>DevOps tools, ingesting VPC flow logs</td>
<td>Subscribe to CloudWatch Logs and analyze in real time</td>
<td>Anomaly detection</td>
</tr>
</tbody>
</table>
Amazon Kinesis Firehose

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

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.

Elastic: Scales to match data throughput

Serverless ETL using AWS Lambda - Firehose can invoke your Lambda function to transform incoming source data.
Capture IT and app logs, device and sensor data, and more

Enable near-real time analytics using existing tools

- Send data from IT infrastructure, mobile devices, sensors
- Integrated with AWS SDK, agents, AWS IoT
- Fully managed service to prepare & ingest streaming data
- Elastic w/o resource provisioning
- Pay-as-you-go: 3.5 cents / GB transferred
- Transform, batch, compress, and encrypt data before loading
- Loads data into Amazon Redshift tables by using the COPY command
Amazon Kinesis Firehose to Amazon S3

data source

source records

Firehose delivery stream

transformed records

transformation failure

destination S3 bucket

source records

backup S3 bucket
Amazon Kinesis Firehose to Amazon Redshift

data source

source records

Firehose delivery stream

transformed records

transformation failure

delivery failure

intermediate S3 bucket

transformed records

Redshift cluster

COPY

backup S3 bucket

source records

source records
Amazon Kinesis Firehose to Amazon Elasticsearch Service

data source

source records

Firehose delivery stream

transformed records

Elasticsearch cluster

backup S3 bucket

source records

delivery failure

transformation failure
Amazon Kinesis Analytics
# Streaming Data Scenarios Across Verticals

<table>
<thead>
<tr>
<th>Scenarios/Verticals</th>
<th>Accelerated Ingest-Transform-Load</th>
<th>Continuous Metrics Generation</th>
<th>Machine Learning and Actionable Insights</th>
</tr>
</thead>
<tbody>
<tr>
<td>Digital Ad Tech/Marketing</td>
<td>Publisher, bidder data aggregation</td>
<td>Advertising metrics like coverage, yield, and conversion</td>
<td>User engagement with ads, optimized bid/buy engines</td>
</tr>
<tr>
<td>IoT</td>
<td>Sensor, device telemetry data ingestion</td>
<td>Operational metrics and dashboards</td>
<td>Device operational intelligence and alerts</td>
</tr>
<tr>
<td>Gaming</td>
<td>Online data aggregation, e.g., top 10 players</td>
<td>Massively multiplayer online game (MMOG) live dashboards</td>
<td>Leader board generation, player-skill match</td>
</tr>
<tr>
<td>Consumer Online</td>
<td>Clickstream analytics</td>
<td>Metrics like impressions and page views</td>
<td>Recommendation engines, proactive care</td>
</tr>
<tr>
<td>Operation Security</td>
<td>DevOps tools, ingesting VPC flow logs</td>
<td>Subscribe to CloudWatch Logs and analyze logs in real time</td>
<td>Anomaly detection</td>
</tr>
</tbody>
</table>
Amazon Kinesis Analytics

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

**Build real-time applications:** Perform continuous 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
A simple streaming query

• Tweets about the AWS Summit
• Selecting from a STREAM of tweets, *an in-application stream*
• Each row has a corresponding ROWTIME

```
SELECT STREAM ROWTIME, author, text
FROM Tweets
WHERE text LIKE '%#AWSSummit%'
```
Better together...
Amazon Kinesis Analytics and Streams

- Kinesis Analytics offers a fully managed alternative for processing your stream
- Simple SQL interface allows you to quickly iterate on new use cases
- Reads directly from an Amazon Kinesis stream
Generate time series analytics

- Compute key performance indicators over time windows
- Combine with historical data in Amazon S3 or Amazon Redshift
Create real-time alarms and notifications

• Build sequences of events from the stream like user sessions in a clickstream or app behavior through logs

• Identify events (or a series of events) of interest and react to the data through alarms and notifications
Feed real-time dashboards

- Validate and transform raw data, and then process to calculate meaningful statistics
- Send processed data downstream for visualization in BI and visualization services
More on real-time dashboards...
Who is Phil?

- **Phil Basford**: @philipbasford
  - Head Of Engineering
  - 3.5 Years at Palringo
  - Co-lead / directed AWS migration
  - Very experienced Java + NodeJS developer
  - Introduced docker to Pal!
What and who is Palringo?

- 10 years old, 3 major pivots.
- We have 50 million signed up users.
- Message Rate:
  - 500 - 1500 MPS Inbound
  - 16,000 MPS Outbound
- A start-up: 80 Employees in 3 Offices
  - (Newcastle, Ipswich, and London)
  - VC Backed.
- An agile product driven company, constantly innovating.
The backstory - Pre April 2016

• **Datacentre**
  - Running our own racks within a datacentre in London Globalswitch.
  - Major outages from DDOS, Power cuts and hardware failures.

• **Core**
  - Massive monolithic system; gluing Chat messaging, the database and business logic together. That was accessed via a binary protocol.
  - Deployment of 8 nodes required total system outages and had to be deployed simultaneously.
  - Known performance issues and limited scalability.

• **Ops Team**
  - 60 developers into 3 gate keepers : Slowing deployment of product increments down from agile teams.
  - The 3 gate keepers were kept up all night with constant alarms and spent all day fire fighting.
Disappointed Business

Sad Developers + Ops
Change everything!
Focus on our Product
(Nov 2015 - April 2016)

- Amazon manage the hardware
- Amazon manage (some of) the software
- Scalability
- Reliability (Multi-AZ)
We set an aim to leave our data centre by April 2017.
Real Time Analytics: Discovery

Palringo

Featured Groups
- Simon says fun
- BEST Gaming Group
- come play hangman with us!
- Official Palringo Lounge

Active Groups
- [aimon says fun]
- [heist freaks]
- [bot central]

Featured Groups
- Mason Competition & Chum
- Chum is Currently Available
- If you wanna live in, act like it
- Chatterbox
- What have you got to say?

Active Groups
- [filipino all over the...]
- Taboo Competition
- [insomaniacs]
- Free beer and pizza
Real Time Analytics: Stats

[simon says fun]

JOIN

PROFILE STATS

14-day Trend

A total of 176,973 messages were sent by 1,947 users in [simon says fun].

Top 25 Users

[similingo]

Message Types

- Text (98.35%)
- Photo (1.13%)
- Message Pack (0.17%)
- Voice (0.35%)

Fun Facts!

AppleBlue is really inquisitive...
37.03% of lines contained a question!

Pansheep was also looking for answers with a question ratio of 26.08%.

let everyone know what they were up to, with 77 actions.
C ++ Monolithic

BINLOG Stream

8 History MYSQL 5.1 Shards

Main DB

Group Stats DB

Group Heat DB

10 PHP and Python scripts

Read Only Slave

Netapp

Java Services
Stage One - History Micro Service

C++
Monolithic

8 History MYSQL 5.1 Shards
Stage One - History Micro Service

C++
Monolithic

RabbitMQ Servers
Auto Scaling

HA Queue

History Writer
Security Group
Auto Scaling

Amazon DynamoDB
Stage Two - Group Heat and Stats

RabbitMQ Servers
Auto Scaling

Group Stats DB
Group Heat DB

10 PHP and Python scripts

HA Queue
CREATE OR REPLACE STREAM "DESTINATION_SQL_STREAM" (GROUP_ID bigint, USER_ID bigint, SENT INTEGER, MSG_COUNT INTEGER) ;

-- and inserts into output stream (DESTINATION_SQL_STREAM)
CREATE OR REPLACE PUMP "STREAM_PUMP" AS INSERT INTO "DESTINATION_SQL_STREAM"

-- Count the number of messages
SELECT STREAM TARGET_ID, SOURCE_ID, MAX(SENT/1000000), count(*) AS MSG_COUNT

-- Get data from input stream.
FROM "SOURCE_SQL_STREAM_001"

-- We are only interested in Group messages
WHERE IS_GROUP = true

-- Uses a minute tumbling time window : group by group, originator, boo
GROUP BY TARGET_ID, SOURCE_ID, FLOOR("SOURCE_SQL_STREAM_001".ROWTIME - TIMESTAMP '1970-01-01 00:00:00') MINUTE / 1 TO MINUTE);
exports.handler = (event, context, callback) => {

    let heatMap = {
        GroupHeats: []
    };

    event.Records.forEach((record) => {
        const data = record.kinesis.data;

        parsePayload(data, (payload) => {

            let hist = {
                "GroupId": payload.GROUP_ID,
                "SubId": payload.USER_ID,
                "Sent": payload.SENT,
                "MessageCount": payload.MSG_COUNT
            };

            heatMap.GroupHeats.push(hist);
        });
    });

    sendToService(heatMap);
};
Benefits

• They are now real time, not near real time.
  • No longer a long poll of history, instead it is from the ‘real’ message feed.

• The “Business” was amazed by the insights.
  • We placed a visualisation over the results and started to learn insights.

• Reduction in lines of code.
  • Kinesis Analytics handled the tumbling time or filtering, we just had to write SQL.
  • New media types introduced faster.
  • Bugs found and corrected.
  • Consolidation into Node JS.
Benefits cont.

• **Separation of scaling**
  • Each component could scale and cope with spikes in message rates.

• **Operational Independence**
  • Self-healing of our components.
  • Plus full managed AWS components

• **Skills**
  • No big learning curve, we did not have to understand how to deploy and run an open source solution.
  • Only SQL and NodeJS skills were required, we could use our existing talent pool (no new hires).
Alternatives:

- Serverless vs. Lift + Shift

Costs:

- $150.40 per month (kinesis)
- $173.04 per month (kinesis analytics)
- $11.23 per month (lambda)

Implementation:

- 3 Day Kinesis + Lambda + Micro Service.
- 3 days QA
- 1 day rollout.
- 1 week optimisations.
Gotchas

• Incomplete unicode support - Arabic + Emoji.
• Batch size vs iterator age.
• Async HTTP microservices to detach connection.
• HTTP Keep-Alive
• String process : splitting, counting occurrences, word count, and line count.
AWS Listen to us!

Feedback loop
- Elliott Williamson (AM)
- Ken Payne (SA)
- Paul Maddox
- Kinesis Team

Rewards
- Early Access
- Road Maps
- Insight
- Requests
We completed our migration by April 2017. No more data centre!
Coming Soon!

• Deep learning and AI for Discovery
• Business Intelligence and Data Lake
• Palringo Framework
• More Microservices
• Protocol V3
The End
Conclusions

- Amazon Kinesis offers: managed service to build applications, streaming data ingestion, and continuous processing

- Ingest aggregate data using Amazon Producer Library

- Process data using Amazon Connector Library and open source connectors

- Determine your partition key strategy

- Learn more about Amazon Kinesis at https://aws.amazon.com/kinesis/

London Amazon Redshift

Wednesday, July 5, 2017 - 6:00 PM to 8:00 PM
60 Holborn Viaduct, London
http://goo.gl/maps/yMZPT

{1:“Redshift Deep Dive and new features since last Meetup” | 2: “OLX presenting Advanced Analytics and Machine Learning with Redshift” | 3:“Other customer/partner case studies” | 4:“Next steps for the community”}
The diminishing value of data

**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
Putting Data in Amazon Kinesis Firehose

- **AWS SDK**
  - PutRecord()
  - PutRecordBatch()

- **Amazon Kinesis Agent**
  - Monitors files and sends new data records to your delivery stream
  - Handles file rotation, checkpointing, and retry upon failures
  - Pre-processing capabilities such as format conversion and log parsing
  - Emits Amazon CloudWatch metrics for monitoring and troubleshooting
Amazon Kinesis Firehose pricing
Simple, pay-as-you-go, and no up-front costs

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Per 1 GB of data ingested</td>
<td>$0.035</td>
</tr>
</tbody>
</table>
Amazon Kinesis Streams pricing

Simple, pay-as-you-go, and no up-front costs

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shard Hour</td>
<td>$0.015</td>
</tr>
<tr>
<td>PUT Payload (per 1,0000,000 PUTs)</td>
<td>$0.014</td>
</tr>
<tr>
<td>Extended Data Retention (Up to 7 days), per Shard Hour</td>
<td>$0.020</td>
</tr>
</tbody>
</table>
Amazon Kinesis Firehose or Amazon Kinesis Streams?
Amazon Kinesis Firehose vs. 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 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.
SQL on streaming data

- SQL is an API to your data
- Ask for what you want, system decides how to get it
- For all data, not just “flat” data in a database
- Opportunity for novel data organization and algorithms
- A standard (ANSI 2008, 2011) and the most commonly used data manipulation language
Appendix
Amazon Kinesis Streams latest features (2016, 2017)

- Extended Retention
- Shard-Level Metrics
- Time-based seek
- Kinesis Analytics Support
- ModifyShards API
- Lambda and Spark Streaming support
Amazon Kinesis Firehose latest features (2016, 2017)

- Delivery for S3, Redshift and Elasticsearch
- Kinesis Agent and log transformation
- Error Reporting and Troubleshooting

- Amazon Kinesis Firehose data transformation
- Kinesis Analytics Support
Demo
Common Integration Pattern with Amazon EMR

Tumbling Window Reporting

Streaming Input
- Amazon Kinesis Streams

Spark
- Tumbling/Fixed Window Aggregation

Periodic Output
- Amazon EMR

COPY from Amazon EMR

Amazon Redshift
Amazon Kinesis Client Library

- Build Kinesis Applications with Kinesis Client Library (KCL)
- Open source client library available for Java, Ruby, Python, Node.JS dev
- Deploy on your EC2 instances
- 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
Build your Streaming Pipeline with Amazon Kinesis, AWS Lambda and Amazon EMR