AWS Summits 2014

Introducing Amazon Kinesis

Ryan Waite, GM AWS Data Services
Adi Krishnan, Sr. PM, Amazon

March 26, 2014

amazon web services

Amazon Kinesis
Managed Service for Streaming Data Ingestion & Processing

- **Origins of Kinesis**
  - The motivation for continuous, real-time processing
  - Developing the ‘Right tool for the right job’
- **What can you do with streaming data today?**
  - Customer Scenarios
  - Current approaches
- **What is Amazon Kinesis?**
  - Kinesis is a building block
  - Putting data into Kinesis
  - Getting data from Kinesis Streams: Building applications with KCL
- **Connecting Amazon Kinesis to other systems**
  - Moving data into S3, DynamoDB, Redshift
  - Leveraging existing EMR, Storm infrastructure

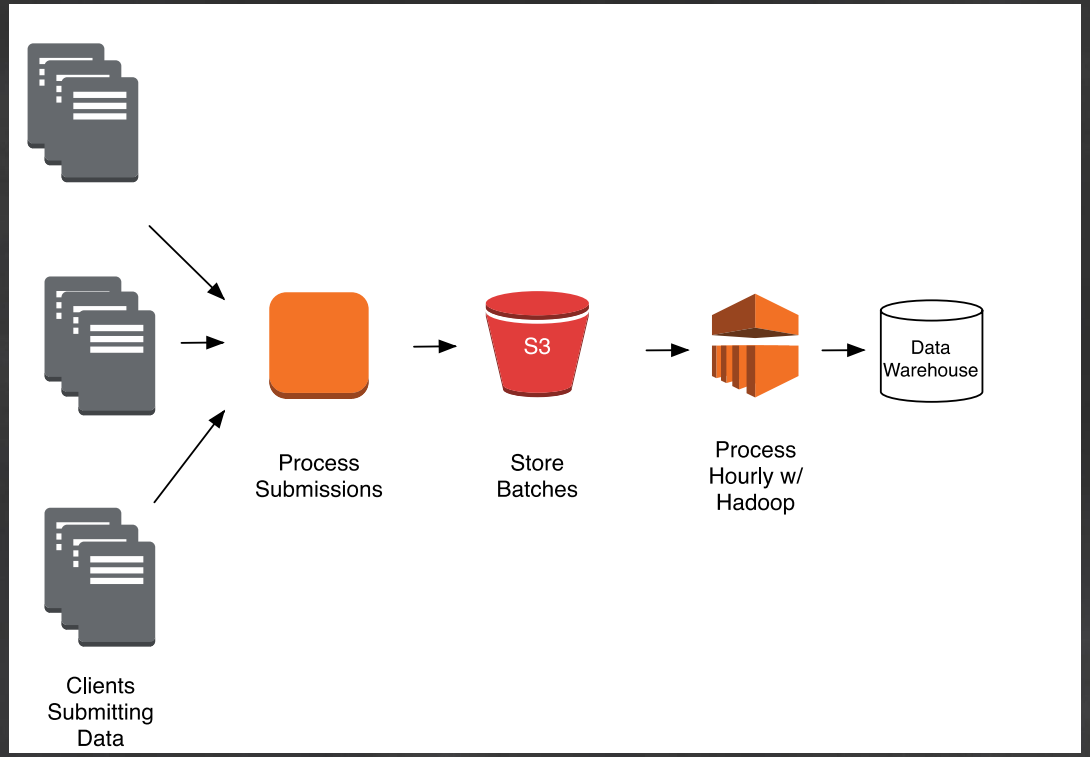
The Motivation for Continuous Processing

Some statistics about what AWS Data Services

- Metering service
  - 10s of millions records per second
  - Terabytes per hour
  - Hundreds of thousands of sources
  - Auditors guarantee 100% accuracy at month end
- Data Warehouse
  - 100s extract-transform-load (ETL) jobs every day
  - Hundreds of thousands of files per load cycle
  - Hundreds of daily users
  - Hundreds of queries per hour

Metering Service

Internal AWS Metering Service



Workload

- 10s of millions records/sec
- Multiple TB per hour
- 100,000s of sources

Pain points

- Doesn’t scale elastically
- Customers want real-time alerts
- Expensive to operate
- Relies on eventually consistent storage

Our Big Data Transition

Old requirements

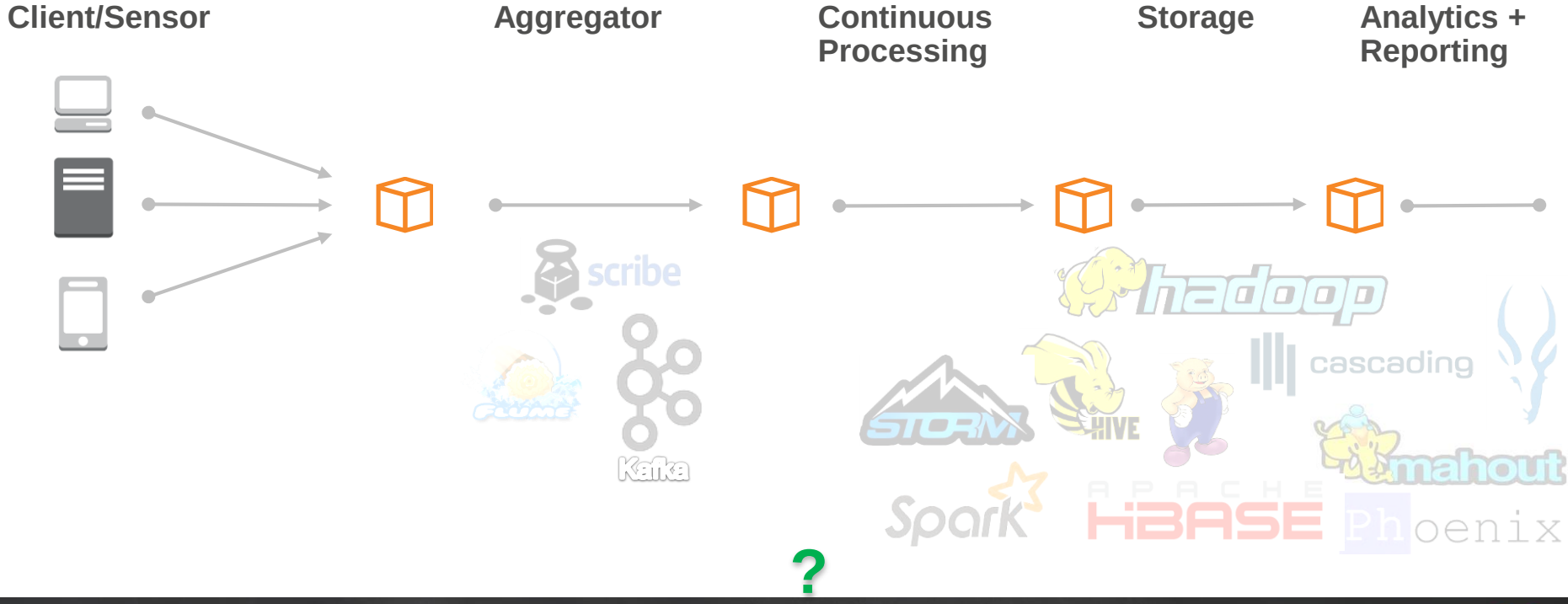
- Capture huge amounts of data and process it in hourly or daily batches

New requirements

- Make decisions faster, sometimes in real-time
- Scale entire system elastically
- Make it easy to “keep everything”
- Multiple applications can process data in parallel

A General Purpose Data Flow

Many different technologies, at different stages of evolution



Big data comes from the small

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

Metering Record

```
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

```
<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

```
"SeattlePublicWater/Kinesis/123/Realtime"
- 412309129140
```

MQTT Record

```
<R,AMZN    ,T,G,R1>
```

NASDAQ OMX Record

Kinesis

Movement or activity in response to a stimulus.

A fully managed service for real-time processing of high-volume, streaming data. Kinesis can store and process terabytes of data an hour from hundreds of thousands of sources. Data is replicated across multiple Availability Zones to ensure high durability and availability.

Customer View

Customer Scenarios across Industry Segments

Scenarios	1 Accelerated Ingest-Transform-Load	2 Continual Metrics/ KPI Extraction	3 Responsive Data Analysis
Data Types	IT infrastructure, Applications logs, Social media, Fin. Market data, Web Clickstreams, Sensors, Geo/Location data		
Software/ Technology	IT server , App logs ingestion	IT operational metrics dashboards	Devices / Sensor Operational Intelligence
Digital Ad Tech./ Marketing	Advertising Data aggregation	Advertising metrics like coverage, yield, conversion	Analytics on User engagement with Ads, Optimized bid/ buy 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
Consumer Online/ E-Commerce	Online customer engagement data aggregation	Consumer engagement metrics like page views, CTR	Customer clickstream analytics, Recommendation engines

What Biz. Problem needs to be solved?

	Mobile/ Social Gaming	Digital Advertising Tech.
	Deliver continuous/ real-time delivery of game insight data by 100's of game servers	Generate real-time metrics, KPIs for online ad performance for advertisers/ publishers
	Custom-built solutions operationally complex to manage, & not scalable	Store + Forward fleet of log servers, and Hadoop based processing pipeline
	• Delay with critical business data delivery • Developer burden in building reliable, scalable platform for real-time data ingestion/ processing • Slow-down of real-time customer insights	• Lost data with Store/ Forward layer • Operational burden in managing reliable, scalable platform for real-time data ingestion/ processing • Batch-driven real-time customer insights
	Accelerate time to market of elastic, real-time applications – while minimizing operational overhead	Generate freshest analytics on advertiser performance to optimize marketing spend, and increase responsiveness to clients

‘Typical’ Technology Solution Set

Solution Architecture Set

- **Streaming Data Ingestion**
  - Kafka
  - Flume
  - Kestrel / Scribe
  - RabbitMQ / AMQP
- **Streaming Data Processing**
  - Storm

- **Do-It-yourself (AWS) based solution**
  - EC2: Logging/ pass through servers
  - EBS: holds log/ other data snapshots
  - SQS: Queue data store
  - S3: Persistence store
  - EMR: workflow to ingest data from S3 and process

- **Exploring Continual data Ingestion & Processing**

Solution Architecture Considerations

Flexibility: Select the most appropriate software, and configure underlying infrastructure

Control: Software and hardware can be tuned to meet specific business and scenario needs.

Ongoing Operational Complexity: Deploy, and manage an end-to-end system

Infrastructure planning and maintenance: Managing a reliable, scalable infrastructure

Developer/ IT staff expense: Developers, Devops and IT staff time and energy expended

Software Maintenance : Tech. and professional services support

Foundation for Data Streams Ingestion, Continuous Processing Right Toolset for the Right Job

Real-time Ingest

- Highly Scalable
- Durable
- Elastic
- Replay-able Reads

Continuous Processing FX

- Load-balancing incoming streams
- Fault-tolerance, Checkpoint / Replay
- Elastic
- Enable multiple apps to process in parallel

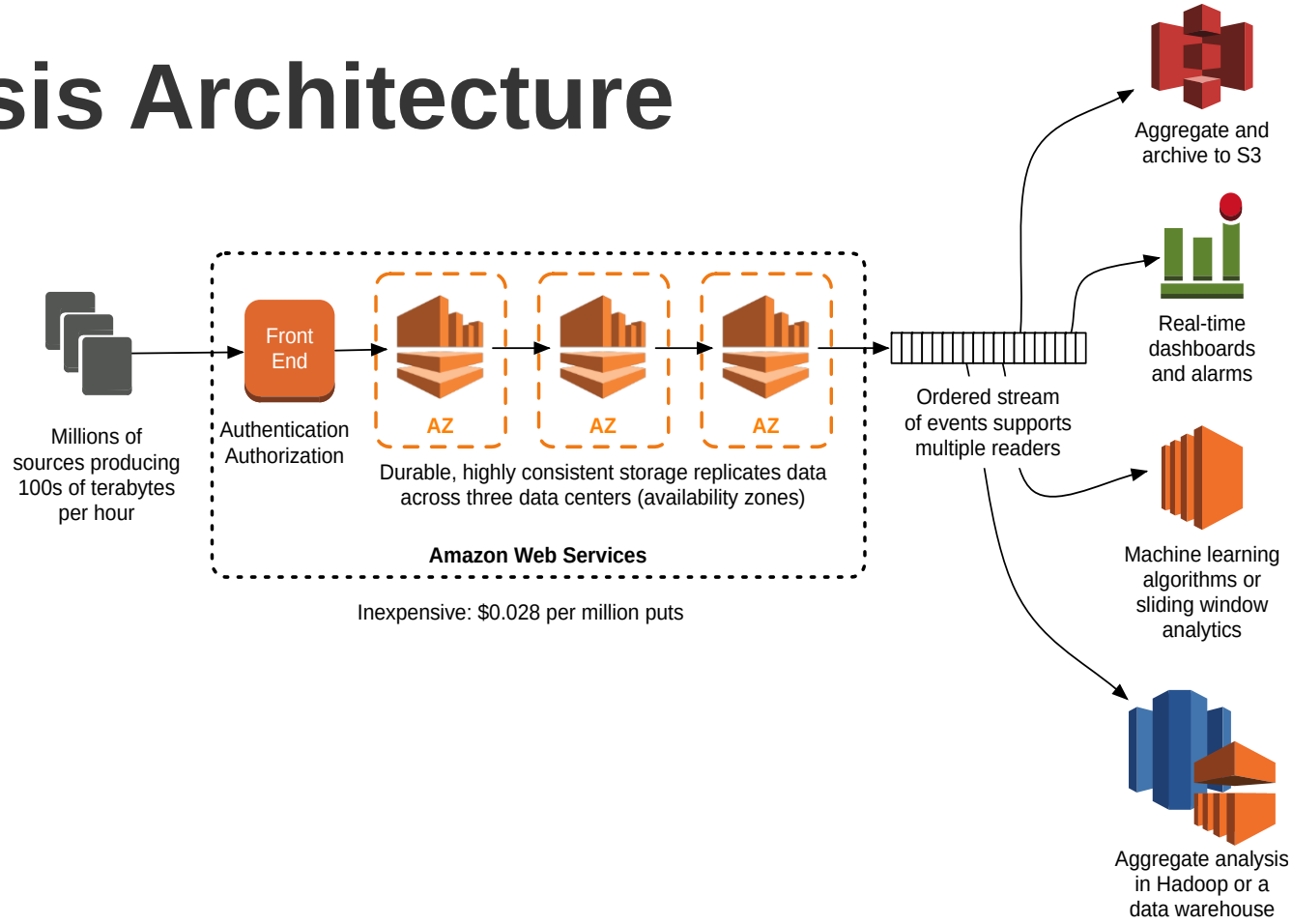
Continuous, real-time workloads

Managed Service

Low end-to-end latency

Enable data movement into Stores/ Processing Engines

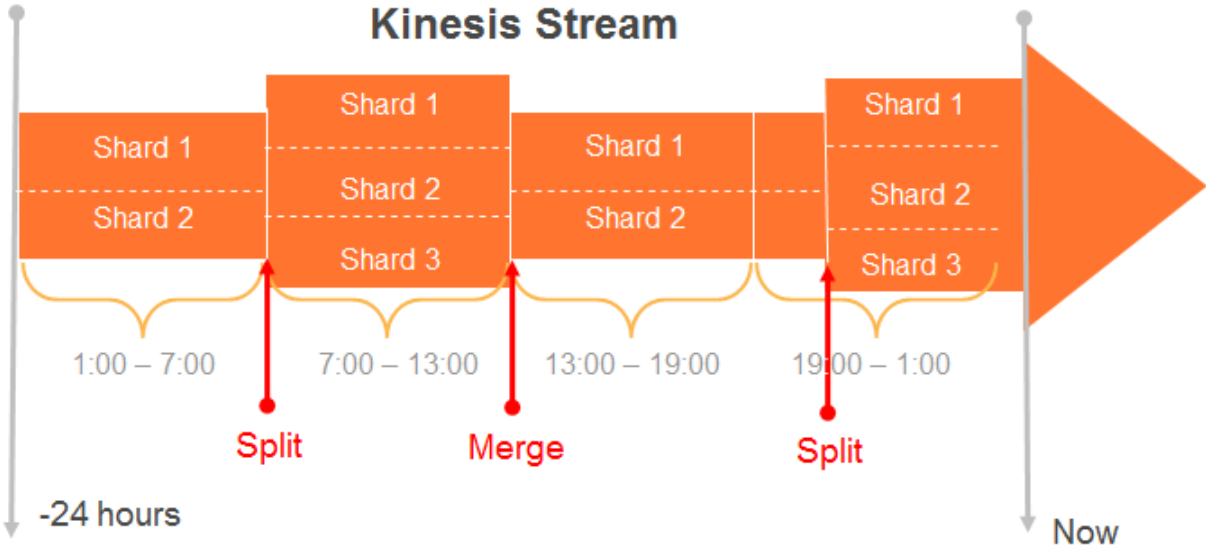
Kinesis Architecture



Amazon Kinesis – An Overview

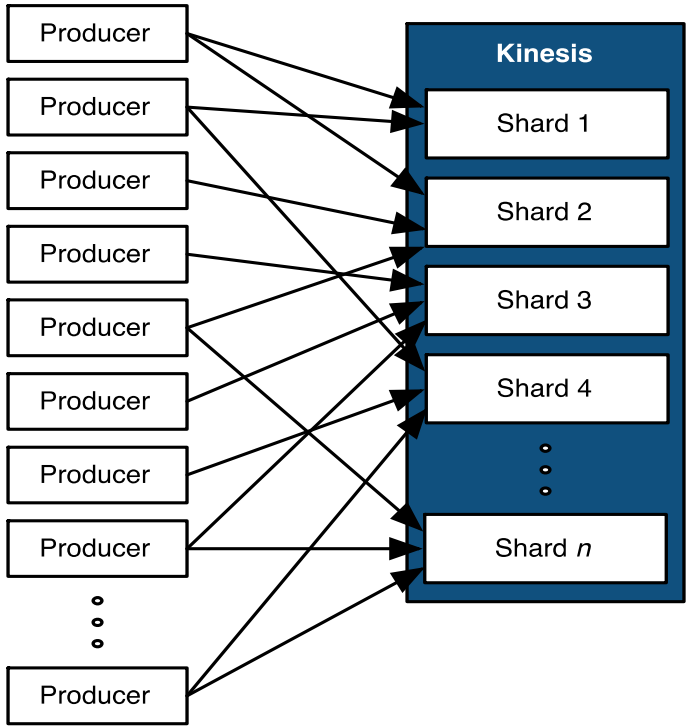
Kinesis Stream:
Managed ability to capture and store data

- 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**
- **Scale** Kinesis streams by adding or removing Shards
- **Replay** data inside of 24Hr. Window

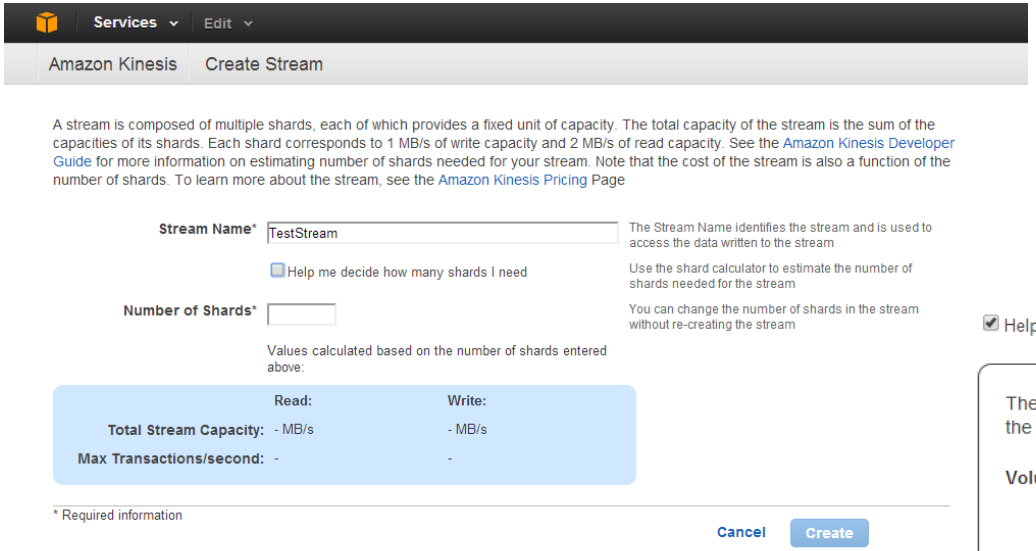


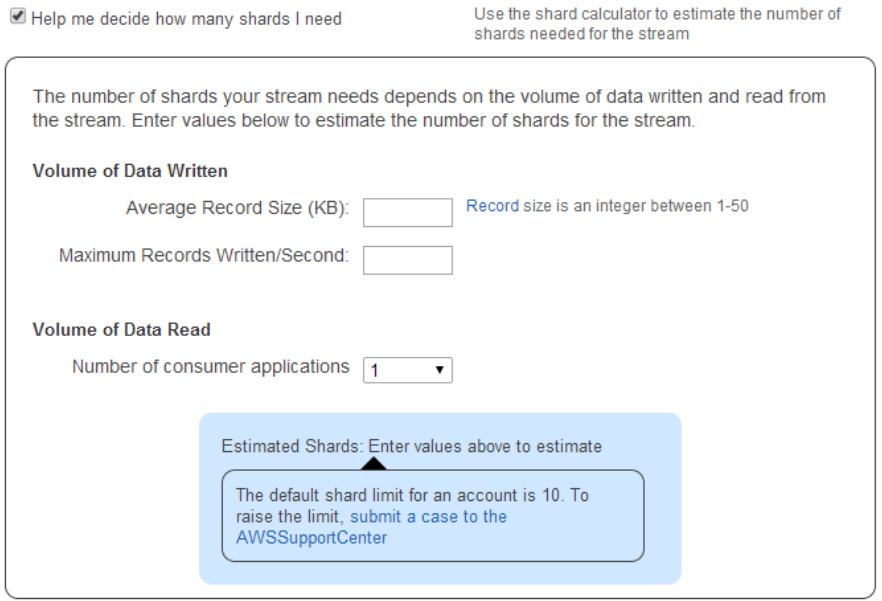
Putting Data into Kinesis
Simple Put interface to store data in Kinesis

- Producers use a **PUT** call to store data in a Stream
- `PutRecord {Data, PartitionKey, StreamName}`
- A **Partition Key** is supplied by producer and used to distribute the PUTs across Shards
- Kinesis **MD5 hashes** supplied partition key over the hash key range of a Shard
- A unique **Sequence #** is returned to the Producer upon a successful PUT call



Creating and Sizing a Kinesis Stream





Getting Started with Kinesis – Writing to a Stream

```
POST / HTTP/1.1
Host: kinesis.<region>.<domain>
x-amz-Date: <Date>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>, SignedHeaders=content-
type;date;host;user-agent;x-amz-date;x-amz-target;x-amzn-requestid,
Signature=<Signature>
User-Agent: <UserAgentString>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: Kinesis_20131202.PutRecord

{
  "StreamName": "exampleStreamName",
  "Data": "XzxkYXRhPl8x",
  "PartitionKey": "partitionKey"
}
```


Sending & Reading Data from Kinesis Streams

Sending

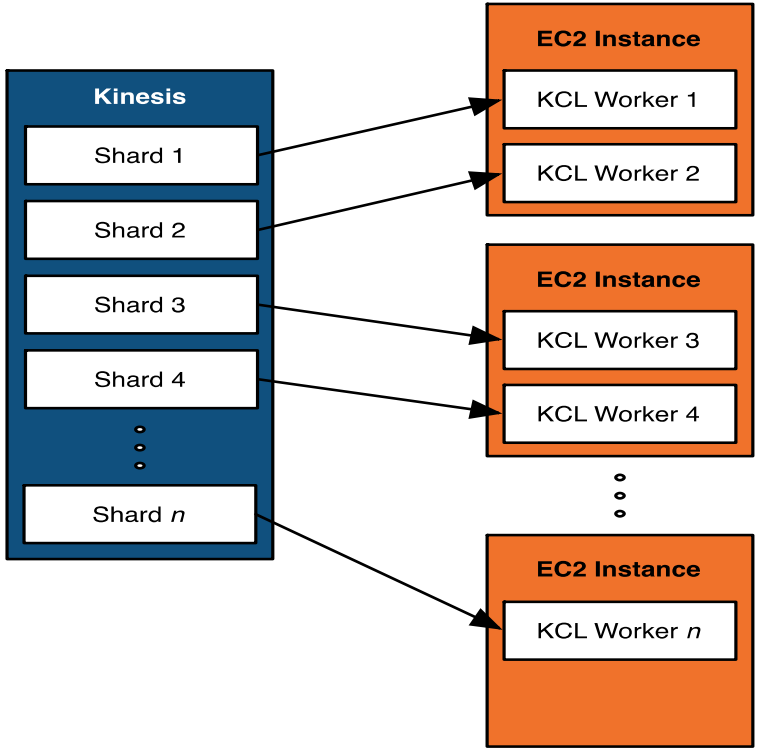
- HTTP Post
- AWS SDK
- LOG4J
- Flume
- Fluentd

Reading

- Get* APIs
- Kinesis Client Library + Connector Library
- Apache Storm
- Amazon Elastic MapReduce

Building Kinesis Processing Apps: Kinesis Client Library
Client library for fault-tolerant, at least-once, Continuous Processing

- Java client library, source available on Github
- Build & Deploy app with KCL on your EC2 instance(s)
- KCL is intermediary b/w your application & stream
  - Automatically starts a Kinesis Worker for each shard
  - Simplifies reading by abstracting individual shards
  - Increase / Decrease Workers as # of shards changes
  - Checkpoints to keep track of a Worker's location in the stream, Restarts Workers if they fail
- Integrates with AutoScaling groups to redistribute workers to new instances



Processing Data with Kinesis : Sample RecordProcessor

```
public class SampleRecordProcessor implements IRecordProcessor {
      @Override
      public void initialize(String shardId) {
         LOG.info("Initializing record processor for shard: " + shardId);
         this.kinesisShardId = shardId;
      }

      @Override
      public void processRecords(List<Record> records, IRecordProcessorCheckpointer checkpointer) {
         LOG.info("Processing " + records.size() + " records for kinesisShardId " + kinesisShardId);

         // Process records and perform all exception handling.
         processRecordsWithRetries(records);

         // Checkpoint once every checkpoint interval.
         if (System.currentTimeMillis() > nextCheckpointTimeInMillis) {
             checkpoint(checkpointer);
             nextCheckpointTimeInMillis = System.currentTimeMillis() + CHECKPOINT_INTERVAL_MILLIS;
            }
      }
}
```

Processing Data with Kinesis : Sample Worker

```
IRecordProcessorFactory recordProcessorFactory = new
SampleRecordProcessorFactory();
Worker worker = new Worker(recordProcessorFactory,
kinesisClientLibConfiguration);

int exitCode = 0;
try {
    worker.run();
} catch (Throwable t) {
    LOG.error("Caught throwable while processing data.", t);
    exitCode = 1;
}
```

Amazon Kinesis Connector Library

Customizable, Open Source code to Connect Kinesis with S3, Redshift, DynamoDB

Kinesis

ITransformer

- Defines the transformation of records from the Amazon Kinesis stream in order to suit the user-defined data model

IFilter

- Excludes irrelevant records from the processing.

IBuffer

- Buffers the set of records to be processed by specifying size limit (# of records)& total byte count

IEmitter

- Makes client calls to other AWS services and persists the records stored in the buffer.

S3

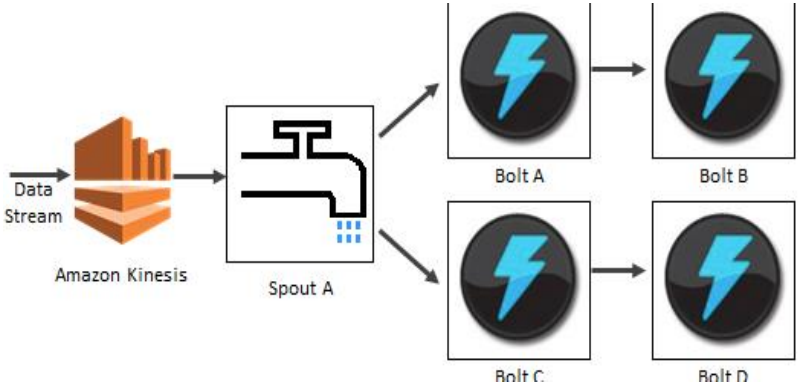
DynamoDB

Redshift

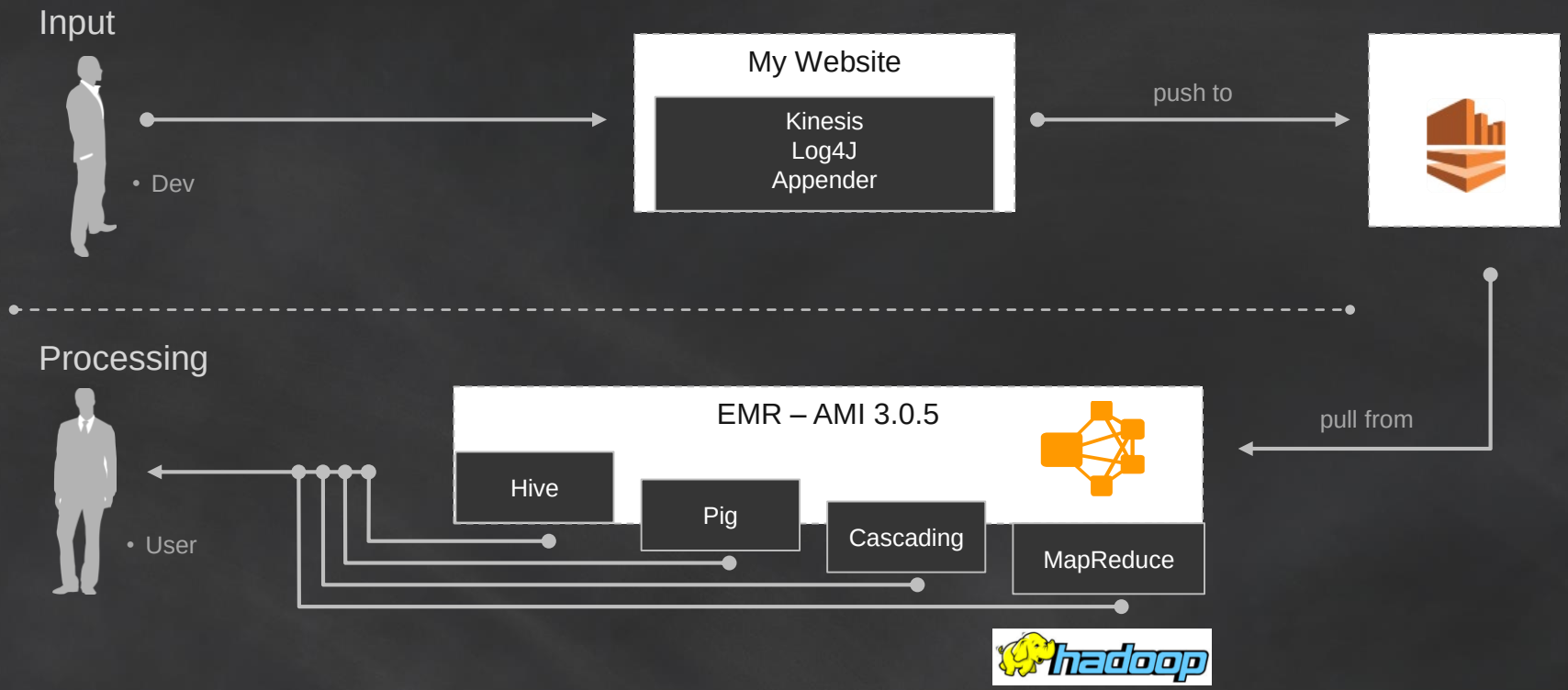
More Options to read from Kinesis Streams Leveraging Get APIs, existing Storm topologies

- **Use the Get APIs for raw reads of Kinesis data streams**
  - `GetRecords {Limit, ShardIterator}`
  - `GetShardIterator {ShardId, ShardIteratorType, StartingSequenceNumber, StreamName}`

- **Integrate Kinesis Streams with Storm Topologies**
  - Bootstraps, via Zookeeper to map Shards to Spout tasks
  - Fetches data from Kinesis stream
  - Emits tuples and Checkpoints (in Zookeeper)



Using EMR to read, and process data from Kinesis Streams

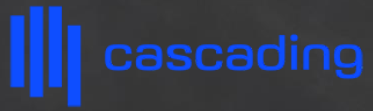


Hadoop ecosystem Implementation & Features

Hadoop Input format

Hive Storage Handler

Pig Load Function



Cascading Scheme and Tap

- Logical names
  - –Labels that define units of work (Job A vs Job B)
- Checkpoints
  - – Creating an input start and end points to allow batch processing
- Error Handling
  - –Service errors
  - –Retries
- Iterations
  - – Provide idempotency (pessimistic locking of the Logical name)

Intended use

- Unlock the power of Hadoop on fresh data
  - Join multiple data sources for analysis
  - Filter and preprocess streams
  - Export and archive streaming data

Customers using Amazon Kinesis

Mobile/ Social Gaming

Deliver continuous/ real-time delivery of game insight data by 100's of game servers

Custom-built solutions operationally complex to manage, & not scalable

- Delay with critical business data delivery
- Developer burden in building reliable, scalable platform for real-time data ingestion/ processing
- Slow-down of real-time customer insights

Accelerate time to market of elastic, real-time applications – while minimizing operational overhead

Digital Advertising Tech.

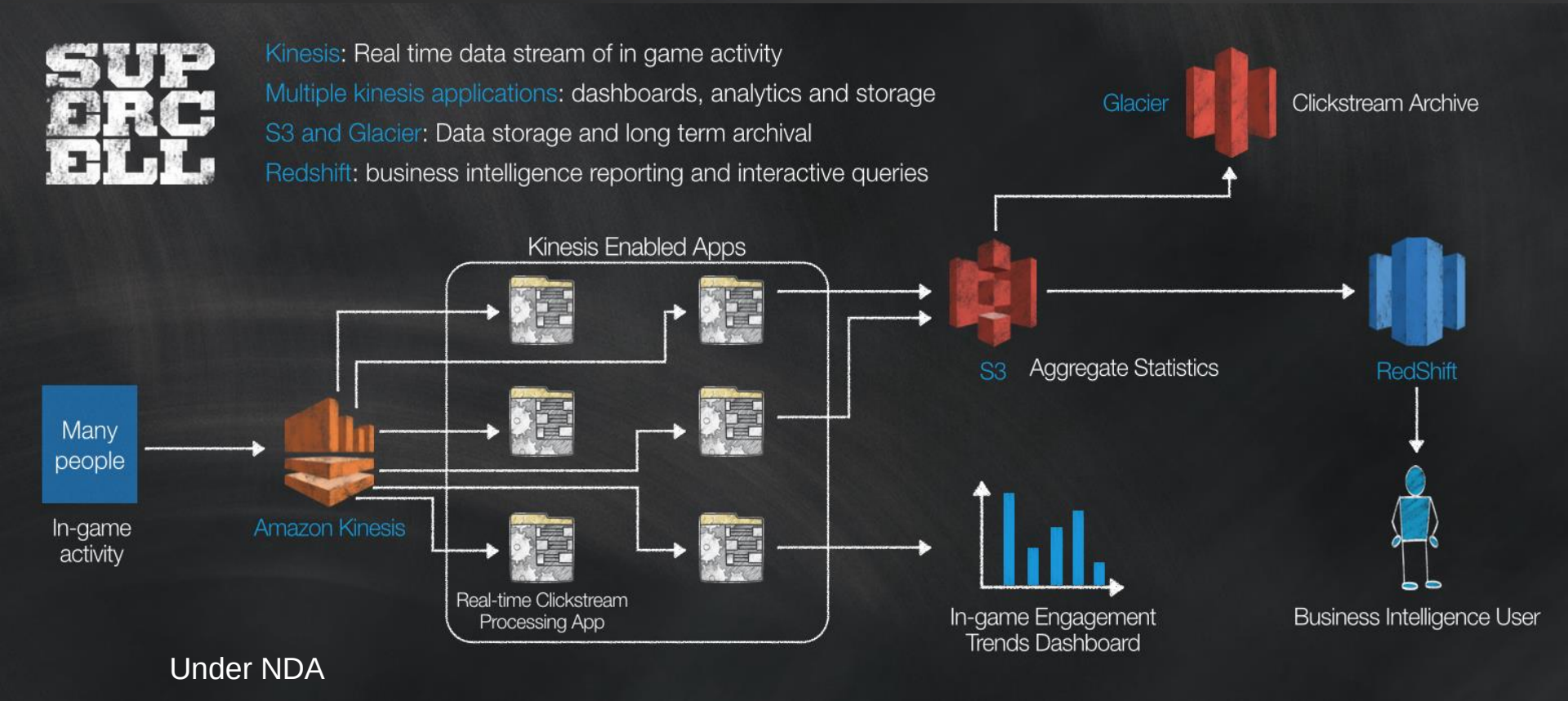
Generate real-time metrics, KPIs for on performance for advertisers/ publishers

Store + Forward fleet of log servers, and Hadoop based processing pipeline

- Lost data with Store/ Forward layer
- Operational burden in managing reliable, scalable platform for real-time data ingestion/ processing
- Batch-driven real-time customer insights

Generate freshest analytics on advertiser performance to optimize marketing spend, and increase responsiveness to clients

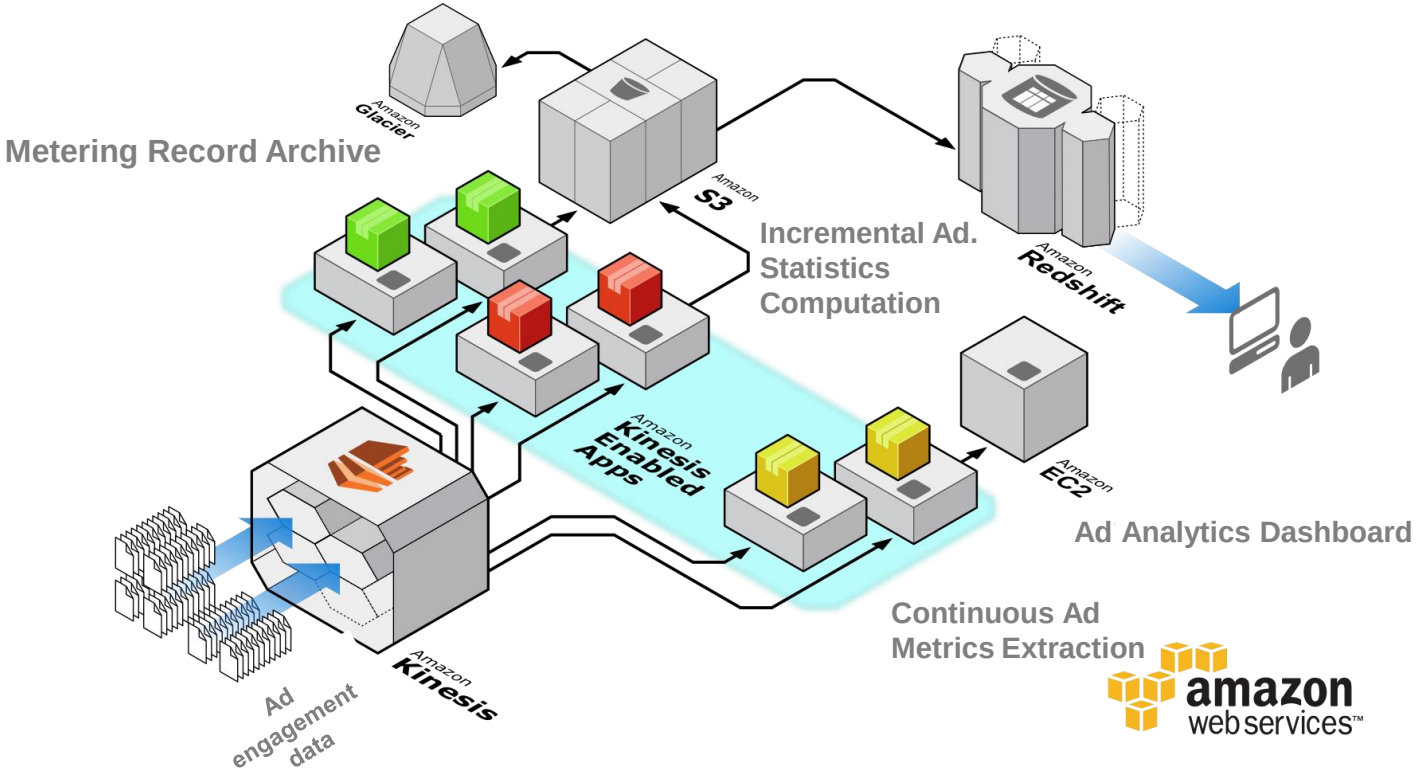
Gaming Analytics with Amazon Kinesis



Under NDA

Digital Ad. Tech Metering with Kinesis

bizo
Business Audience Marketing



Kinesis Pricing
Simple, Pay-as-you-go, & no up-front costs

Pricing Dimension	Value
Hourly Shard Rate	$0.015
Per 1,000,000 PUT transactions:	$0.028

- Customers specify throughput requirements in shards, that they control
- Each Shard delivers 1 MB/s on ingest, and 2MB/s on egress
- Inbound data transfer is free
- EC2 instance charges apply for Kinesis processing applications

Amazon Kinesis: Key Developer Benefits

Easy Administration

Managed service for real-time streaming data collection, processing and analysis. Simply create a new stream, set the desired level of capacity, and let the service handle the rest.

Real-time Performance

Perform continual processing on streaming big data. Processing latencies fall to a few seconds, compared with the minutes or hours associated with batch processing.

High Throughput. Elastic

Seamlessly scale to match your data throughput rate and volume. You can easily scale up to gigabytes per second. The service will scale up or down based on your operational or business needs.

S3, Redshift, & DynamoDB Integration

Reliably collect, process, and transform all of your data in real-time & deliver to AWS data stores of choice, with Connectors for S3, Redshift, and DynamoDB.

Build Real-time Applications

Client libraries that enable developers to design and operate real-time streaming data processing applications.

Low Cost

Cost-efficient for workloads of any scale. You can get started by provisioning a small stream, and pay low hourly rates only for what you use.

Try out Amazon Kinesis

- Try out Amazon Kinesis
  - http://aws.amazon.com/kinesis/
- Thumb through the Developer Guide
  - http://aws.amazon.com/documentation/kinesis/
- Visit, and Post on Kinesis Forum
  - https://forums.aws.amazon.com/forum.jspa?forumID=169#

Thank you!

AWS Summits 2014

Introducing Amazon Kinesis

Managed Service for Streaming Data Ingestion, & Processing

Ryan Waite, GM AWS Data Services

Adi Krishnan, Sr. PM, Amazon

March 26, 2014

amazon web services