AWS re:Invent

ANT317

Perform real-time analytics using open-source software with Amazon MSK

Rajeev Chakrabarti
Principal Specialist Streaming Solutions Architect
Amazon Web Services

Praveen Gattu
Software Development Manager
Amazon Kinesis Data Analytics

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

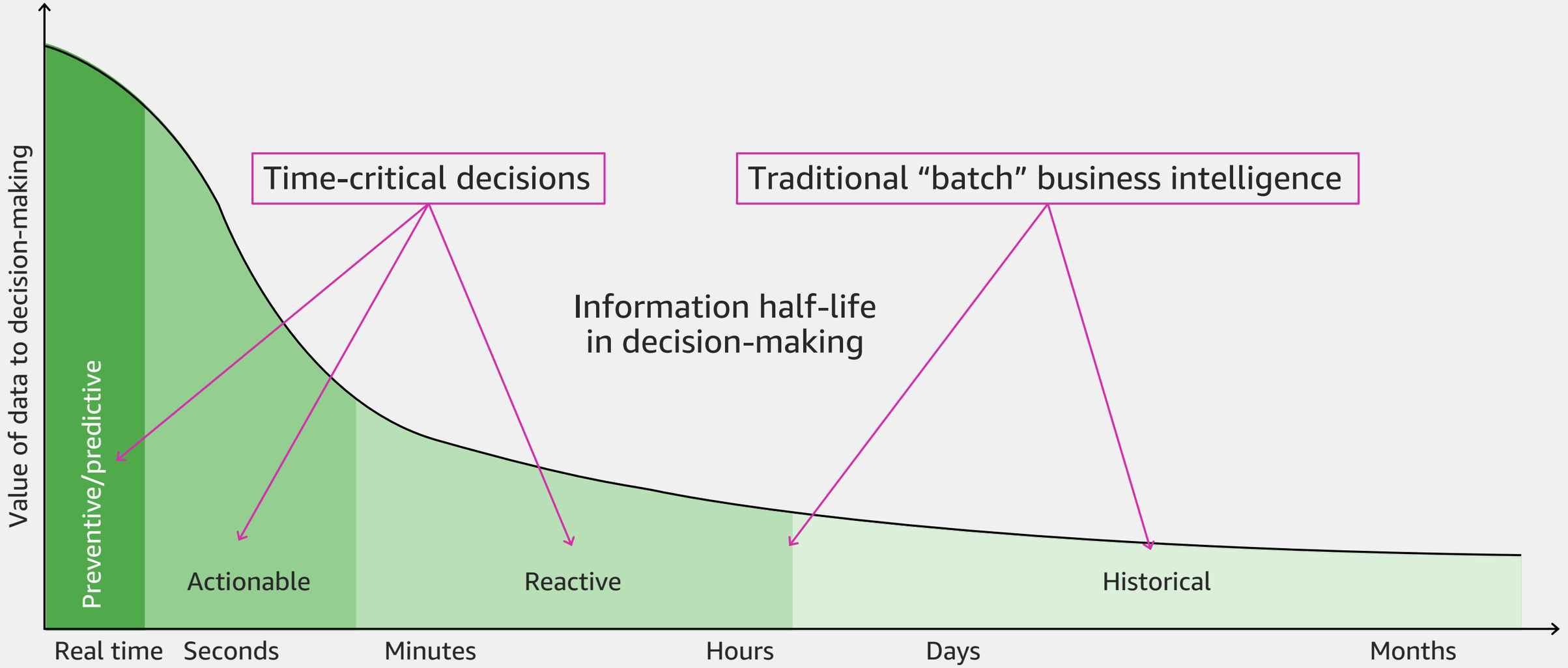
Agenda

- Data streaming and processing overview
- Apache Kafka overview
- Apache Flink overview
- Lets build a clickstream use case
    - Ingesting into Amazon Managed Streaming for Kafka
    - Amazon Kinesis Data Analytics: data partitioning, calculate user statistics
    - (Advanced) Kinesis Data Analytics: Perform user sessionization

Data streaming and processing overview

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Why streaming data?



Source: Perishable insights, Mike Gualtieri, Forrester

Common real-time analytics use cases



Milliseconds

Messaging between microservices

Response analytics (web and mobile app notifications)

Seconds

Log ingestion

IoT device maintenance

Change data capture (CDC)

Minutes

Streaming ETL into data lakes and data warehouses

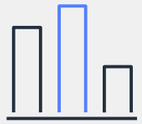
Challenges of data streaming



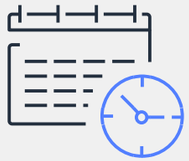
Difficult to set up



Tricky to scale



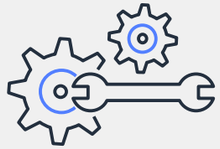
Hard to achieve high availability



Integration requires development



Error prone and complex to manage



Expensive to maintain

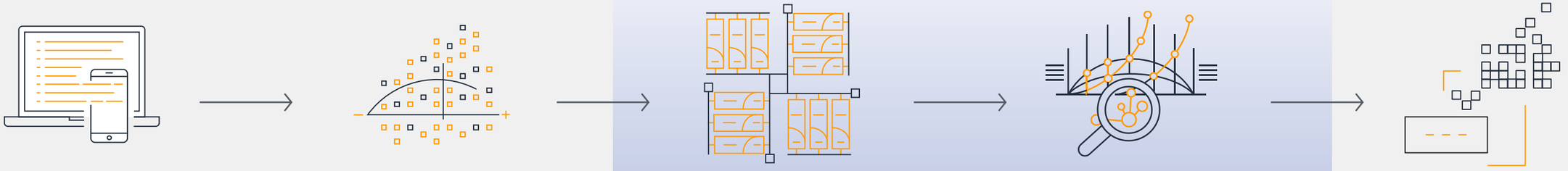
Streaming real-time data with AWS

Easily collect, process, and analyze data streams in real time

- Easy to use
- Elastic
- High availability and durability
- Seamless integration with AWS services
- Fully managed
- Pay for what you use

Enabling real-time analytics

Data streaming technology enables a customer to ingest, process, and analyze high volumes of high-velocity data from a variety of sources in real time



Source

Devices and/or applications that produce real-time data at high velocity

Stream ingestion

Data from tens of thousands of data sources can be written to a single stream

Stream storage

Data are stored in the order they were received for a set duration of time and can be replayed indefinitely during that time

Stream processing

Records are read in the order they are produced, enabling real-time analytics or streaming ETL

Destination

Data lake (most common)

Database (least common)

Stream storage



Data is stored in the order it was received for a set duration of time and can be replayed indefinitely during that time



Amazon Kinesis Data Streams



Amazon Managed Streaming for Kafka

Stream processing



Records are read in the order they are produced, enabling real-time analytics or streaming ETL

AWS services

Amazon EMR

Amazon Kinesis Data Analytics for Java

Flink

Third-party services

Apache Spark

APACHE STORM™ Distributed • Resilient • Real-time

DATADOG

mongoDB.

Anodot

databricks

Qubole

VOLTDB

MEMSQL

splunk>

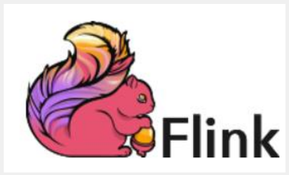
Stream processing: Apache Flink



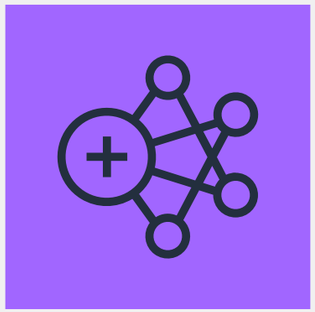
Allows stateful distributed stream processing with exactly once processing for sophisticated stream processing applications

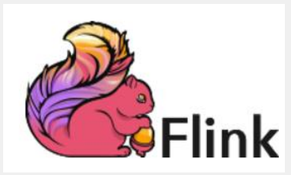
AWS services





Amazon Kinesis Data Analytics for Java



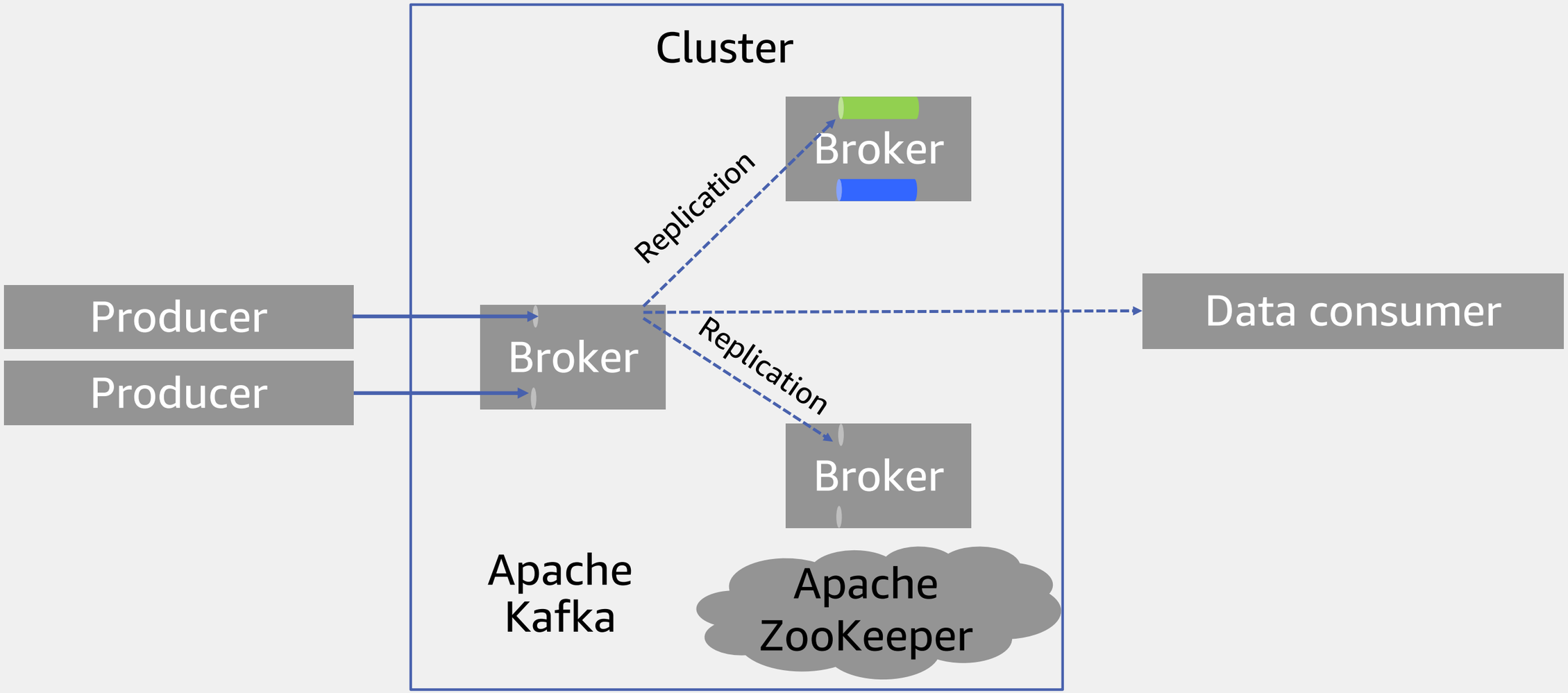


Amazon EMR

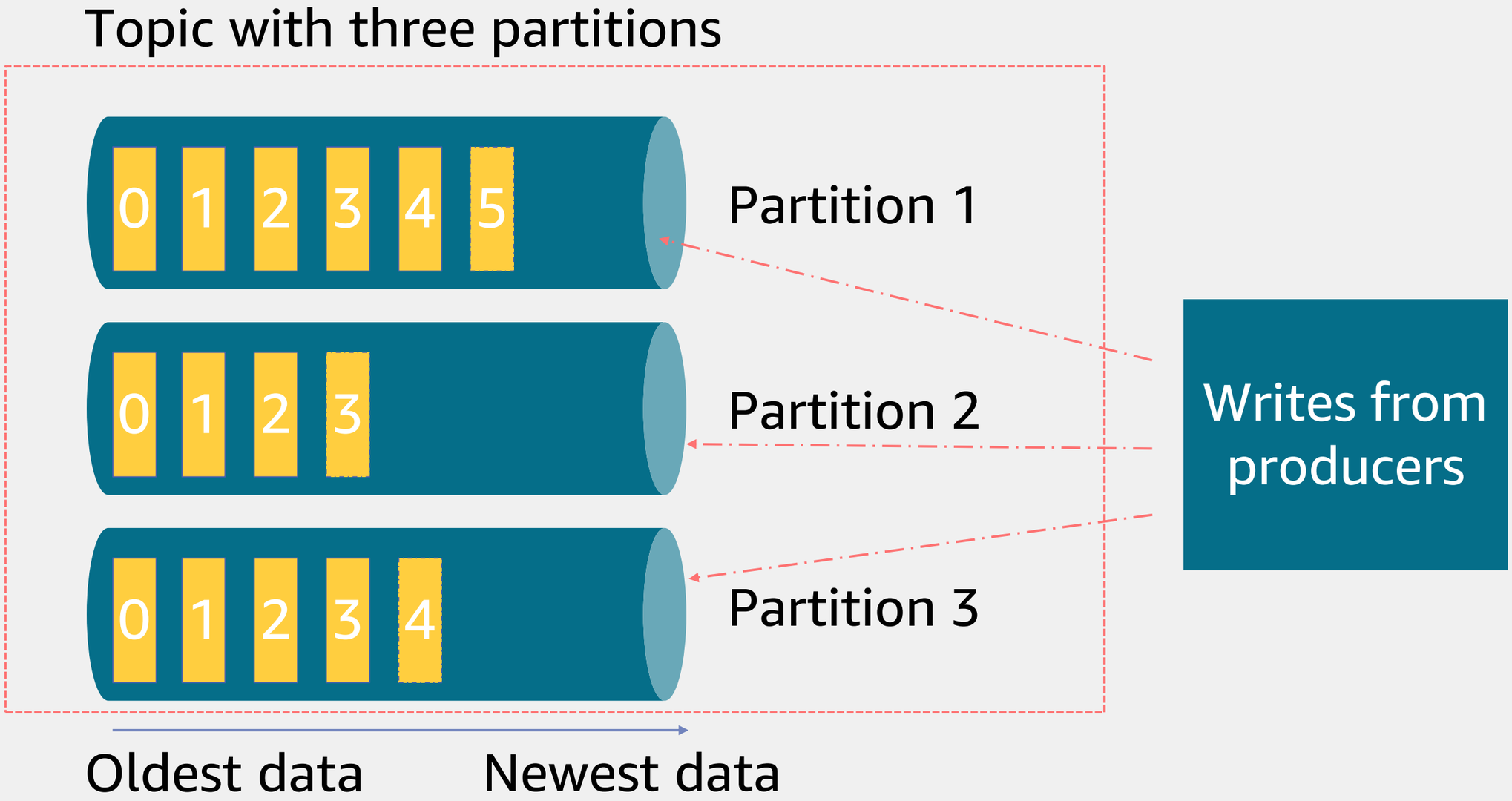
Apache Kafka overview

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

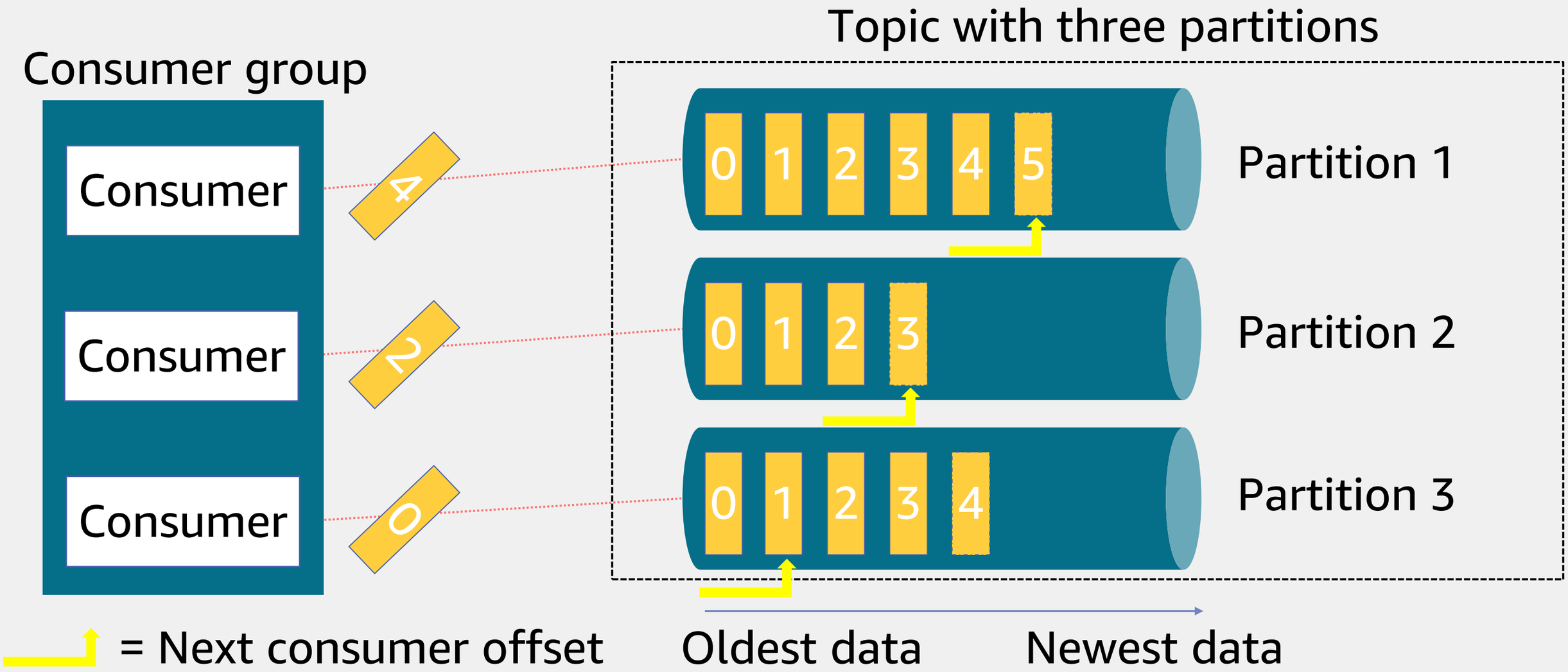
Apache Kafka anatomy 101



Apache Kafka anatomy: Writes to partitions



Apache Kafka anatomy: Reads from partitions



What Amazon MSK does for you

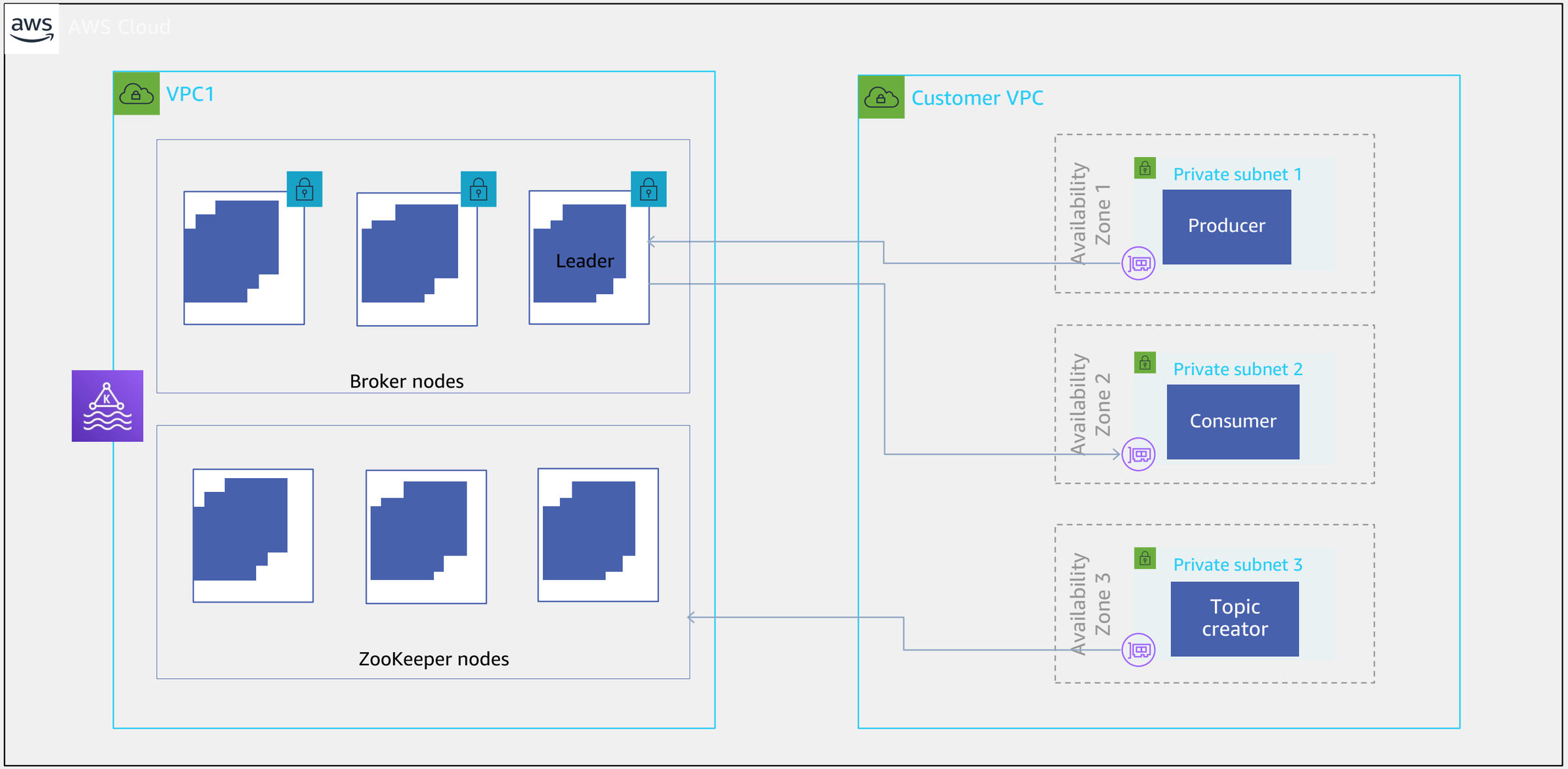


- Makes Apache Kafka more accessible to your organization

$$f(kafka_{usage}) = \sum_{n=1}^{\infty} Streaming\ Apps$$

- Drives best practices through design, defaults, and automation
- Allows developers to focus more on developing applications and less on managing infrastructure
- Amazon MSK is committed to improving open-source Apache Kafka

How connectivity works



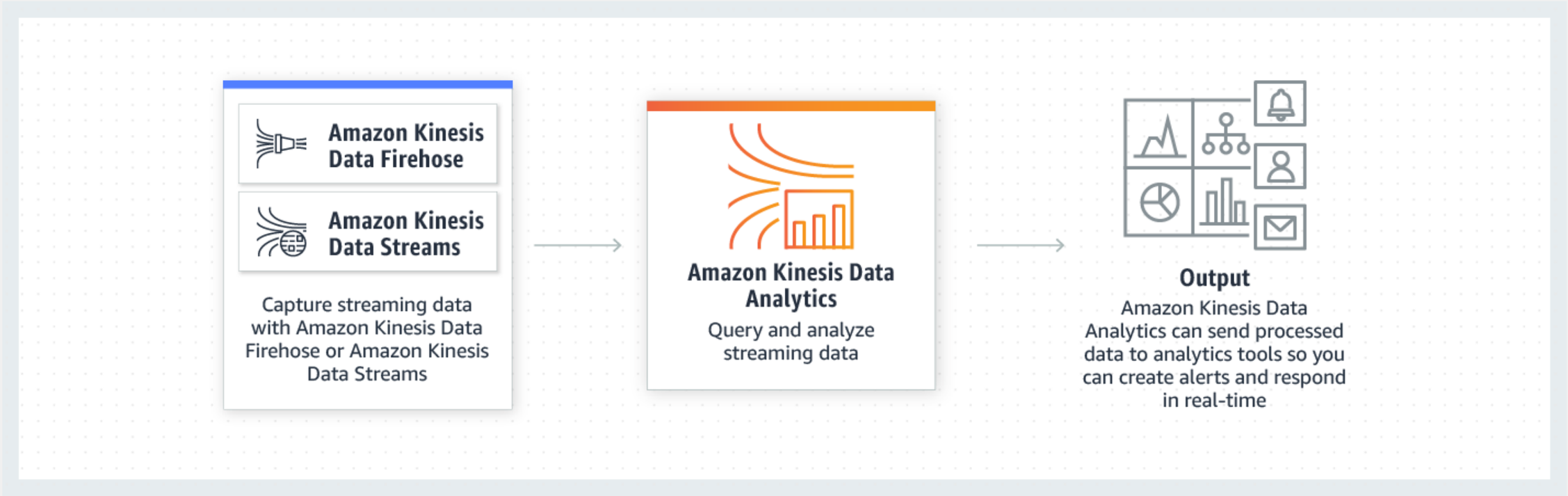
Kinesis Data Analytics for Java

AWS re:Invent

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

aws

Amazon Kinesis Data Analytics



- Interact with streaming data in real time using SQL or integrated Java applications
- Build fully managed and elastic stream processing applications

Apache Flink applications in Kinesis Data Analytics

1

Build Java applications in your integrated development environment of choice using Apache Flink

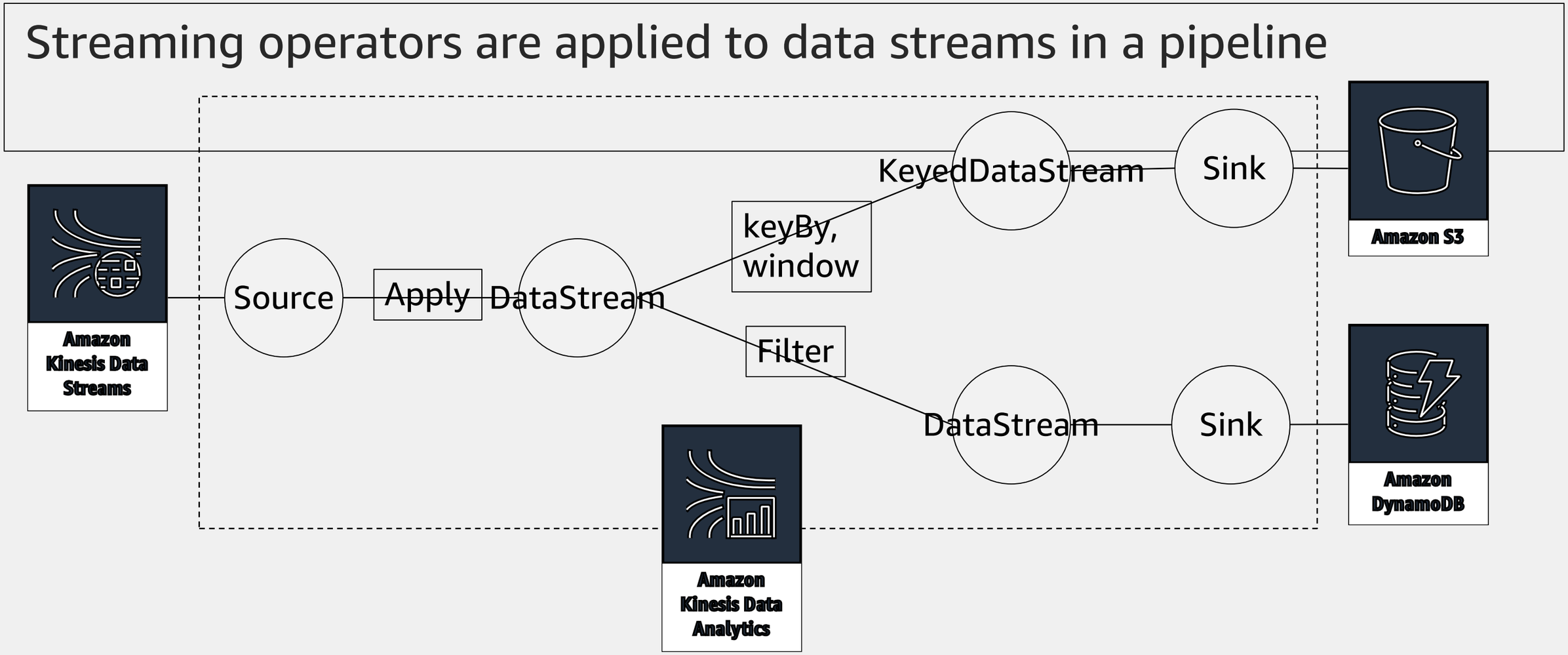
2

Upload your application code to Kinesis Data Analytics

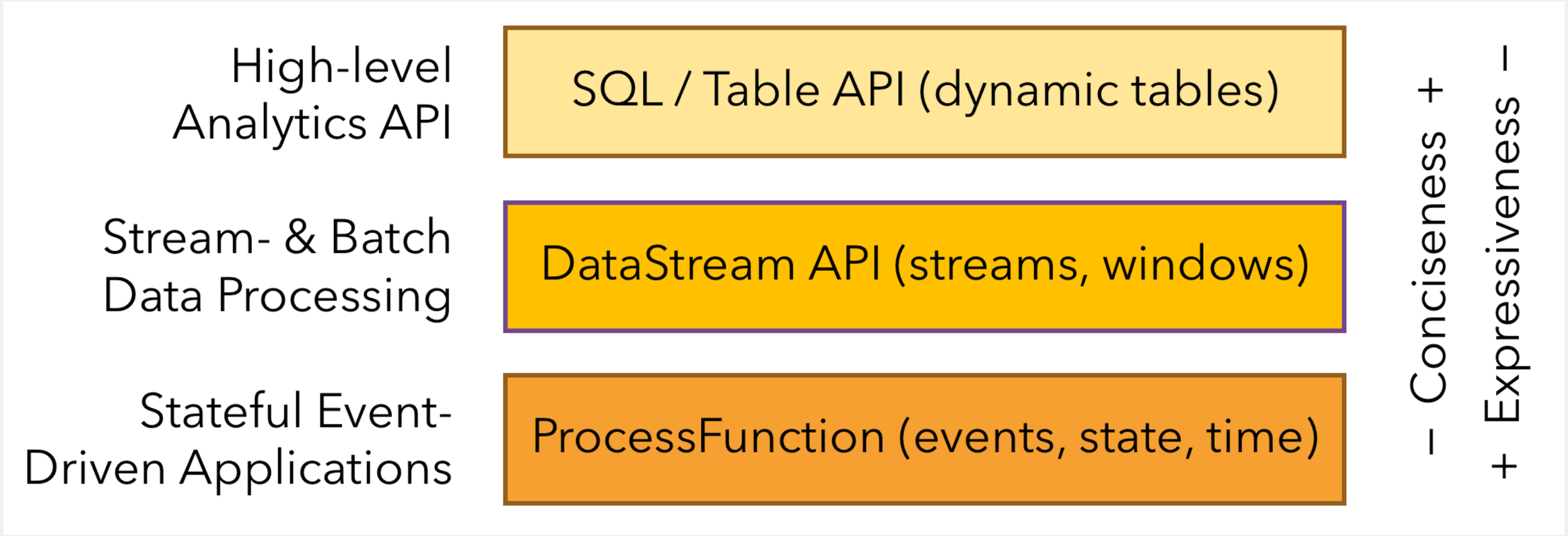
3

Run your application in a fully managed and elastic service

How do you build an application?



Why Apache Flink?



Apache Flink in a nutshell

```
DataStream<String> lines = env.addSource(
                                new FlinkKafkaConsumer<>(…));

DataStream<Event> events = lines.map((line) -> parse(line));

DataStream<Statistics> stats = events
        .keyBy("id")
        .timeWindow(Time.seconds(10))
        .apply(new MyWindowAggregationFunction());

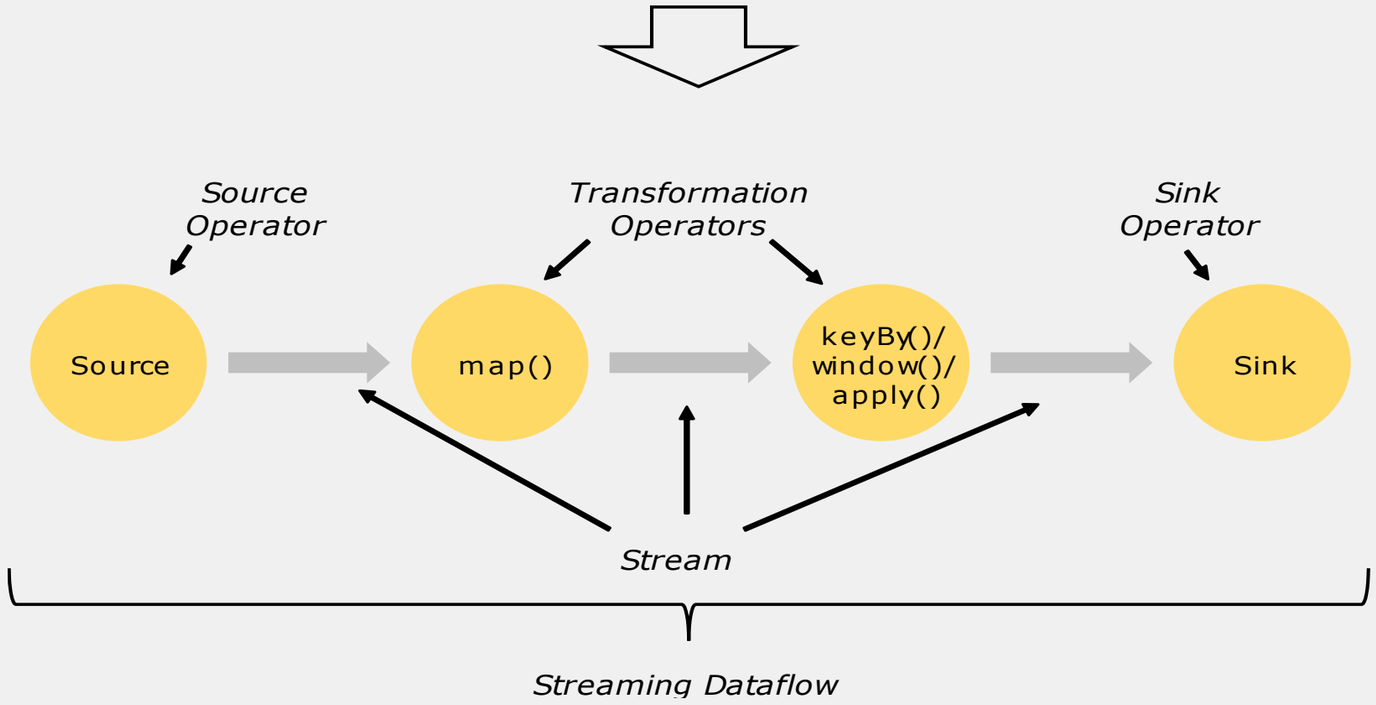
stats.addSink(new RollingSink(path));
```

Source

Transformation

Transformation

Sink



Fault tolerance and checkpoints

Application interruptions

- Crashes: code bugs, hardware issues, network outages, etc.
- Application updates, framework upgrades
- Scaling

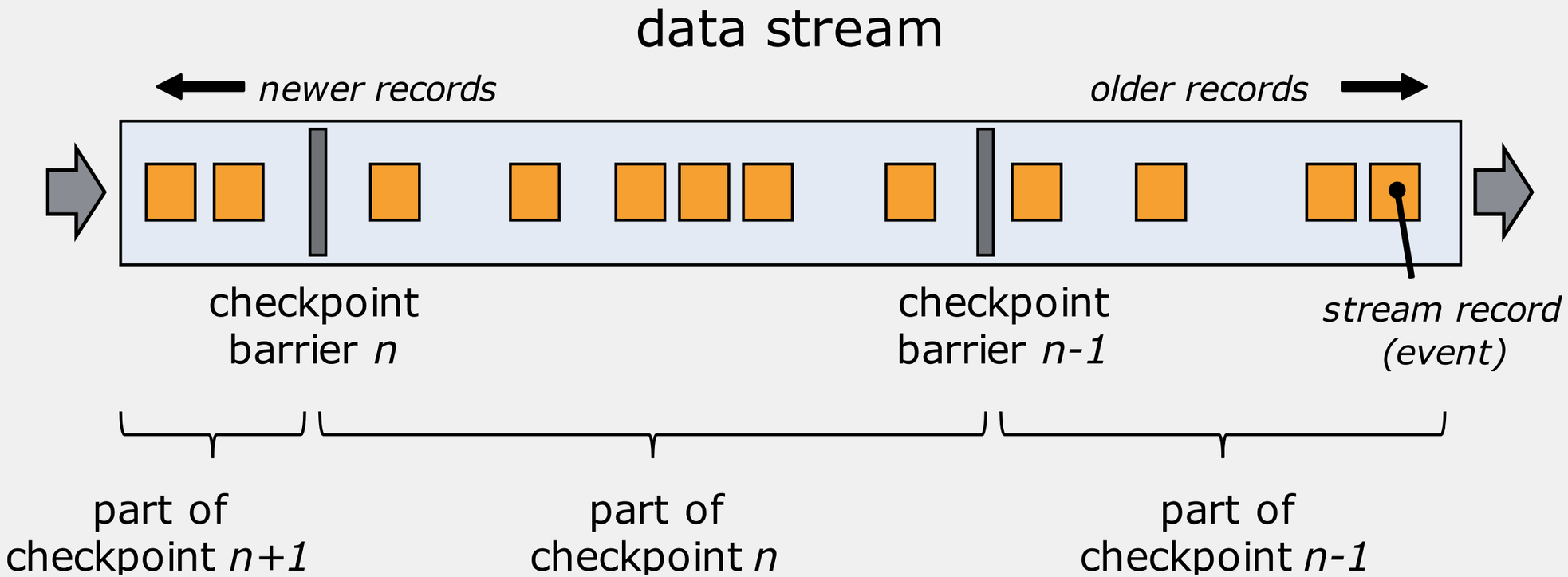
Solution: Periodic checkpoints of application state

Challenges

- State consistency: Every record must be processed exactly once
- Recovery time
- Latency impact to data flow
- Consensus among distributed components

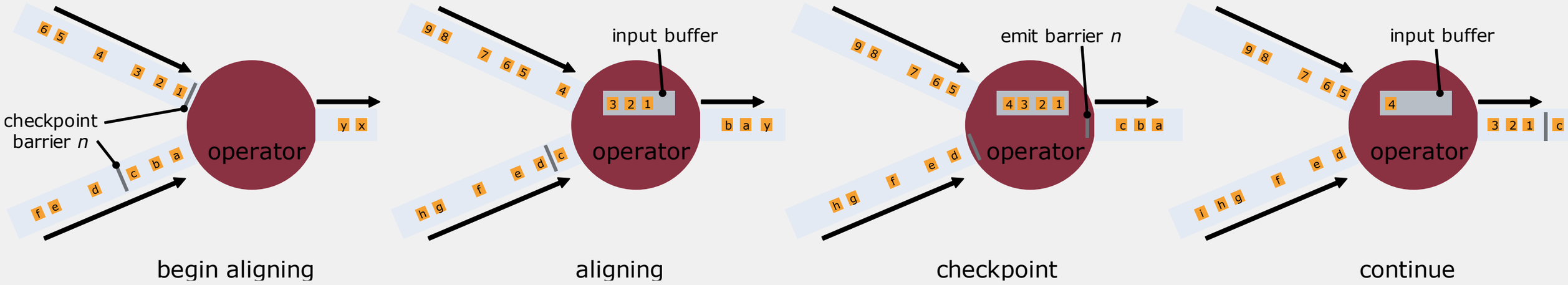
Checkpoints in Apache Flink (Chandy-Lamport)

- Inject *barrier* messages into source streams
- *Barriers* carry unique IDs and separate events into checkpoints
- *Barriers* flow through the data-flow graph along with data records



Checkpoints in Apache Flink

- Operators save state after receiving barrier *n*
- Operators *must align* barriers and *buffer* events more than *n*
- After *alignment*, operator can emit barrier *n* downstream
- Checkpoint *n* is complete when sinks emit barrier *n*



Why does this work?

1. Every operator must process events up to *n*
    - Barrier *n* must flow through every node to complete checkpoint *n*
    - Operators must save state before emitting barrier *n*
2. Checkpoint only covers events up to *n*
    - Events more than *n* are not processed until the state covering *n* is saved
    - Alignment synchronizes inputs at barrier *n*

But wait, there is more!

Savepoints

- Externalized snapshots
- User-managed
- Application versioning
    - Stop/resume without data loss
    - Roll back to a specific state version
    - Update, fork, test applications with specified state
- *Caveat:* Application code and savepoint state must be compatible

Apache Flink supports more than 25 operators

Example operators	Typical usage
Map, FlatMap, Filter, Iterative	**Basic transformations**
Key By, Split, Shuffle, Custom Partition	**Change logical or physical structure of the stream**
Window, Reduce, Fold, Sum, Min, Max	**Analytics and aggregations**
Join, Union, coGroup	**Combine multiple data streams**

and much, much more

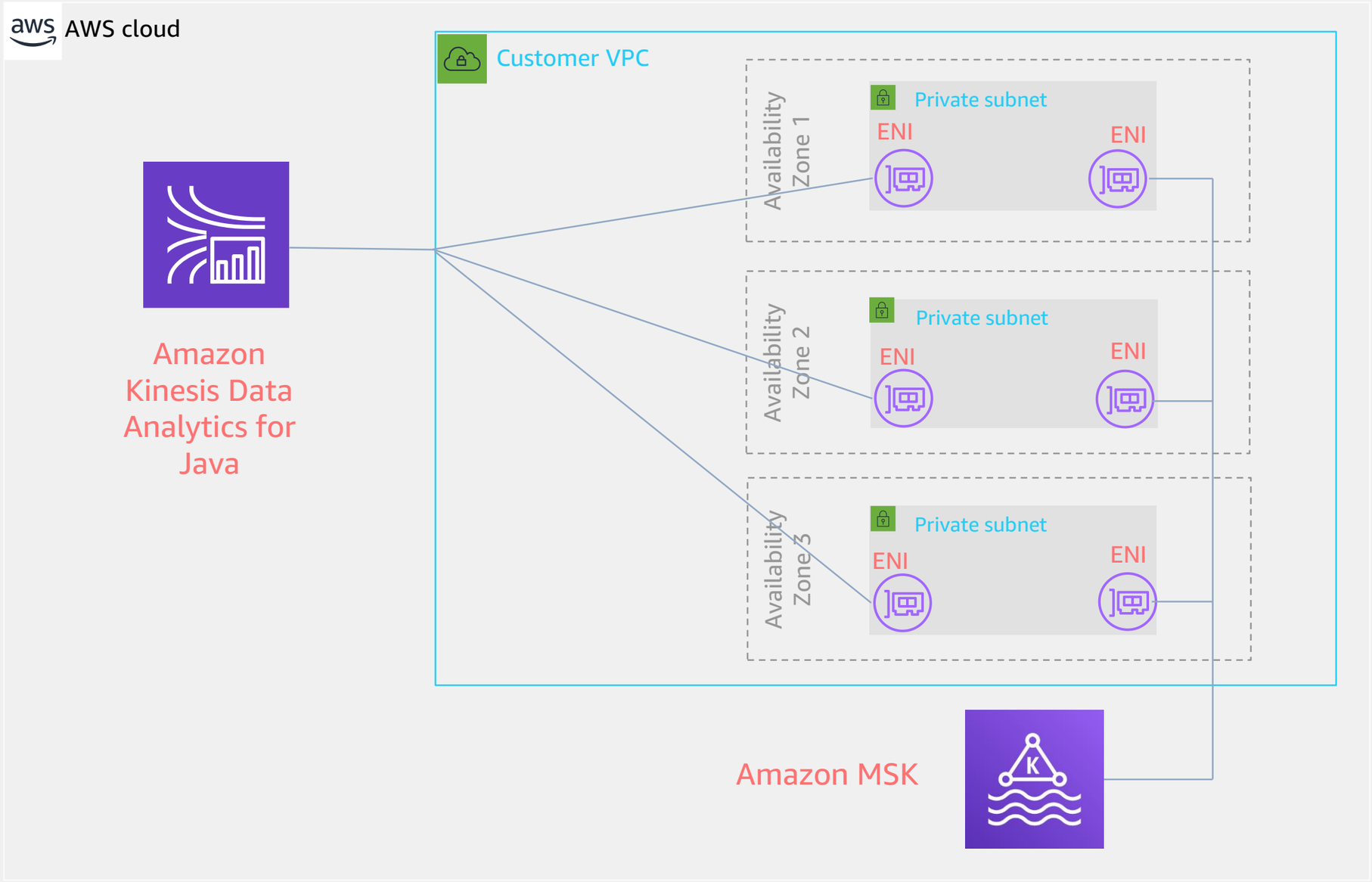
What Kinesis Data Analytics does for you

Fully managed service for Apache Flink application

- Simple APIs for application lifecycle management: Create, start, stop, update applications
- Cluster provisioning and auto scaling: Pay only for what you use
- Drive best practices through design and defaults (e.g., checkpoints, savepoints turned on by default)
- Monitoring and failure recovery
- Committed to improving open-source Apache Flink

Apache Flink connectivity with MSK

Apache Flink on Kinesis Data Analytics for Java applications



Apache Flink connector for Kafka

- Apache Flink has connectors for Apache Kafka that ship with Flink
- Apache Kafka connectors can be used for reading and writing data from or to Apache Kafka topics
- Multiple connectors are available and are compatible with different Apache Kafka and Flink versions
- flink-connector-kafka_2.11, supported since Flink 1.7.0, supports Apache Kafka versions above 1.0.0 and is a universal Kafka connector
- To use the connector, import the connector into your maven project

```
<dependency>
        <groupId>org.apache.flink</groupId>
        <artifactId>flink-connector-kafka-0.11_2.11 </artifactId>
        <version>1.6.2</version>
</dependency>
```

Lets build a clickstream use case

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Problem statement

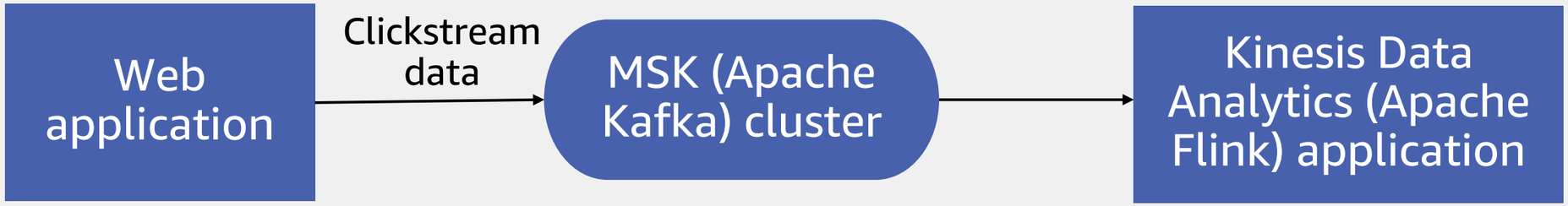
Analyze user activity coming to e-commerce website in real time

- Number of users visiting website
- Number of users grouped by visits to departments
- Number of clicks users go through before going through order pipeline

Sample Producer data

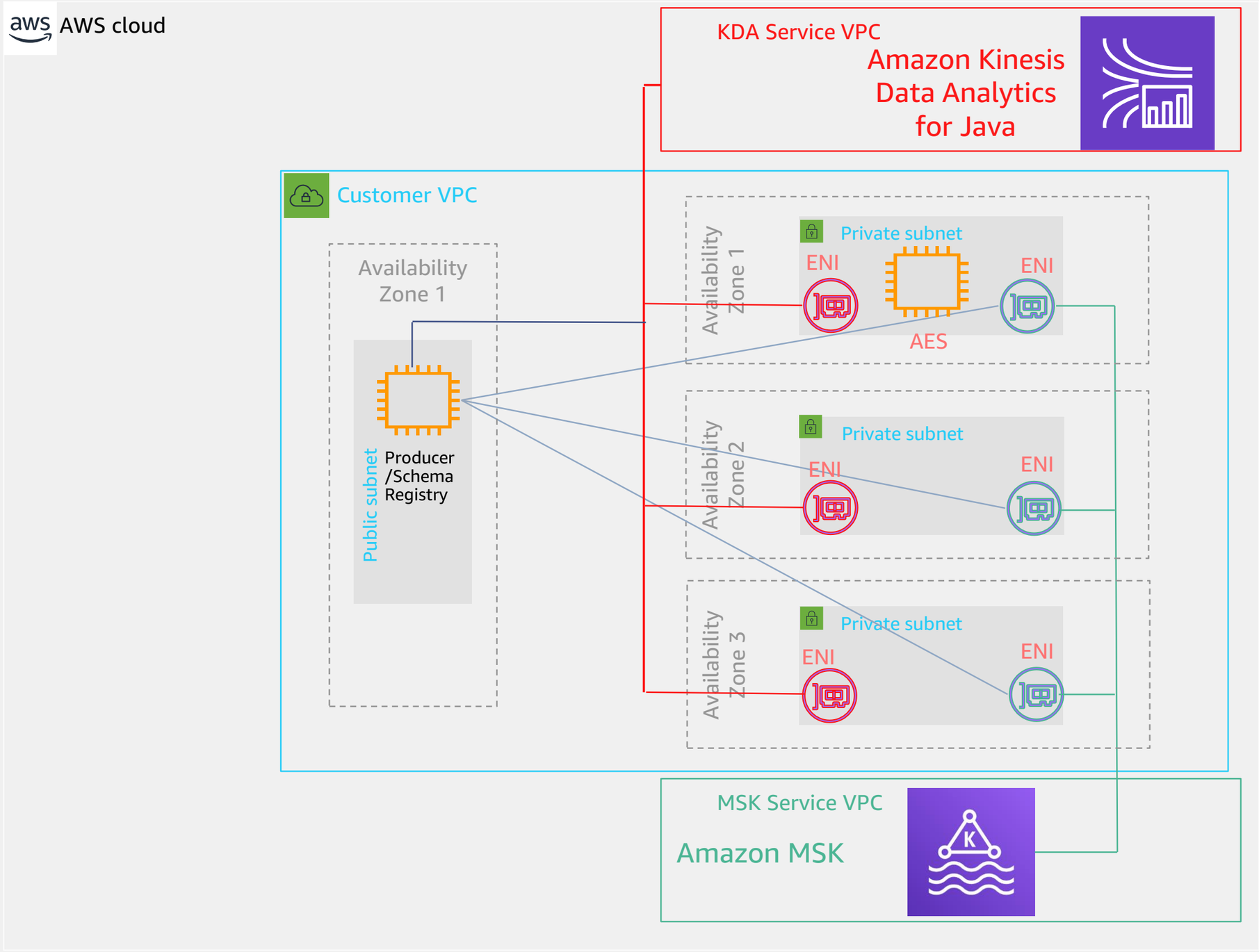
userid	ip	devicetype	event_type	product_type	eventtimestamp	globalseq	prevglobalseq
1633	66.249.1.116	tablet	home_page	N/A	2019-11-20 11:32:38	1	0
1633	66.249.1.116	tablet	product_catalog	N/A	2019-11-20 11:32:39	2	1
1633	66.249.1.116	tablet	home_page	N/A	2019-11-20 11:32:39	3	2
1633	66.249.1.116	tablet	product_catalog	N/A	2019-11-20 11:32:39	4	3
1633	66.249.1.116	tablet	home_page	N/A	2019-11-20 11:32:39	5	4
1633	66.249.1.116	tablet	product_catalog	N/A	2019-11-20 11:32:39	6	5
1633	66.249.1.116	tablet	product_detail	soundbars	2019-11-20 11:32:39	7	6
1633	66.249.1.116	tablet	add_to_cart	soundbars	2019-11-20 11:32:39	8	7
1633	66.249.1.116	tablet	order	soundbars	2019-11-20 11:32:39	9	8
1633	66.249.1.116	tablet	product_catalog	N/A	2019-11-20 11:32:39	10	9
1633	66.249.1.116	tablet	home_page	N/A	2019-11-20 11:32:39	11	10
1633	66.249.1.116	tablet	product_catalog	N/A	2019-11-20 11:32:39	12	11
1633	66.249.1.116	tablet	home_page	N/A	2019-11-20 11:32:39	13	12
1633	66.249.1.116	tablet	product_catalog	N/A	2019-11-20 11:32:39	14	13
1633	66.249.1.116	tablet	product_detail	AirPods	2019-11-20 11:32:39	15	14
1633	66.249.1.116	tablet	product_catalog	N/A	2019-11-20 11:32:39	16	15
1633	66.249.1.116	tablet	home_page	N/A	2019-11-20 11:32:39	17	16
1633	66.249.1.116	tablet	product_catalog	N/A	2019-11-20 11:32:39	18	17
1633	66.249.1.116	tablet	product_detail	soundbars	2019-11-20 11:32:39	19	18

Sample architecture



Demo environment

Apache Flink on Kinesis Data Analytics for Java applications



Setup

- Set up source
- Set up MSK cluster
- Create Kinesis Data Analytics application

Demo

Data inserted into MSK

Demo

- Code that counts the number of customer visits (a simple tumble window example)

- Code that counts the number of customer visits by department (a group-by statement and a tumble window example)

- A session window where data are grouped by customer ID, and we count number of times customers click on a page before placing the order (a session window example?)

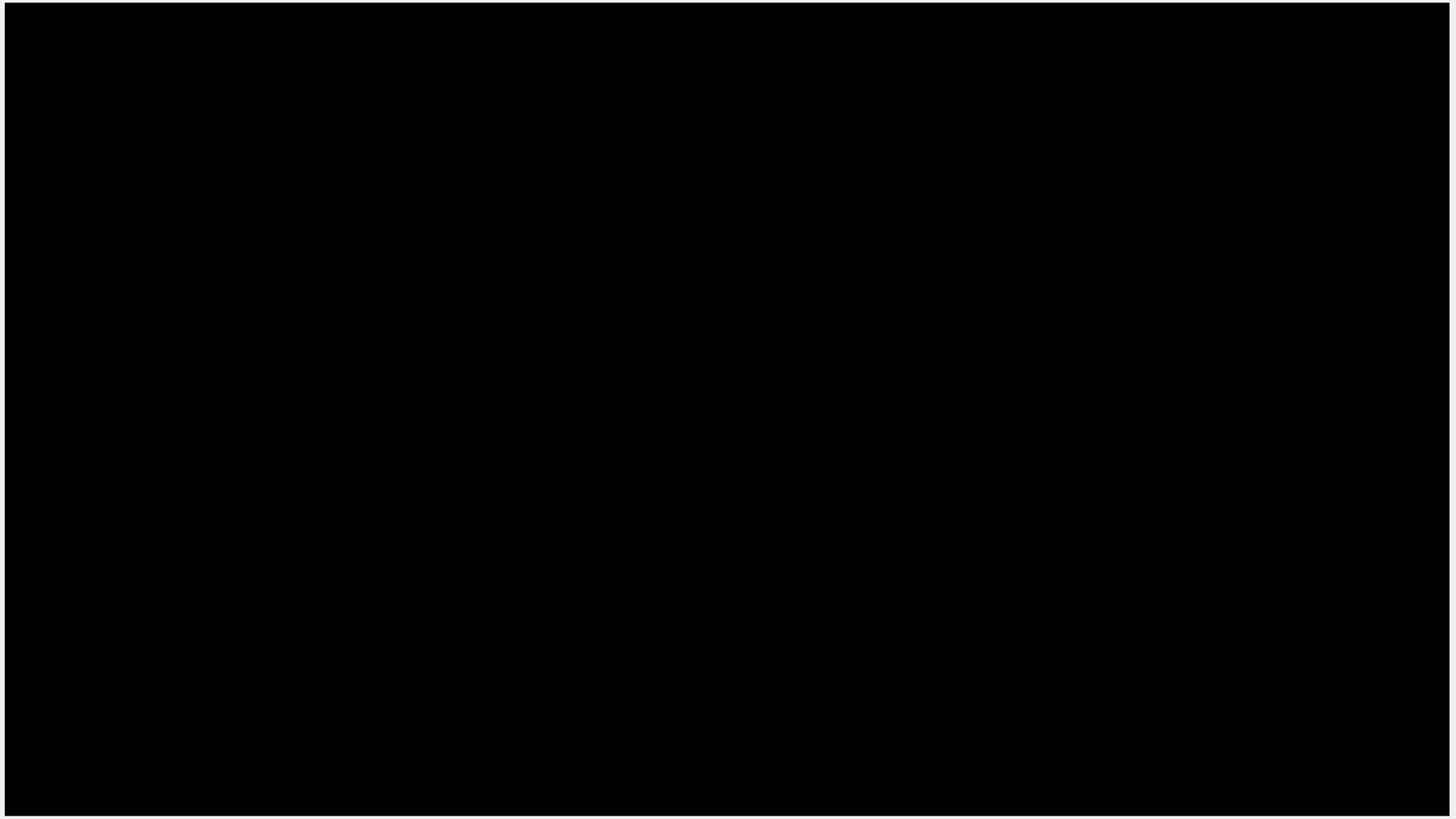
Demo

KDA application deployment

Running the application

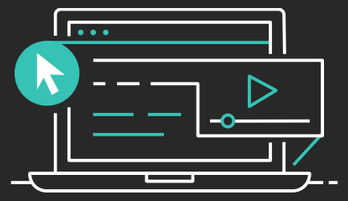
Viewing real-time activity charts in Amazon Elasticsearch

Demo



Learn big data with AWS Training and Certification

Resources created by the experts at AWS to help you build and validate data analytics skills



New free digital course, Data Analytics Fundamentals, introduces Amazon S3, Amazon Kinesis, Amazon EMR, AWS Glue, and Amazon Redshift



Classroom offerings, including Big Data on AWS, feature AWS expert instructors and hands-on labs



Validate expertise with the **AWS Certified Big Data - Specialty** exam or the new **AWS Certified Data Analytics - Specialty** beta exam

Visit aws.amazon.com/training/paths-specialty/

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

aws training and certification

Thank you!

AWS re:Invent

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

aws



Please complete the session survey in the mobile app.

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.