
AWS
re:Invent
NOV. 28 – DEC. 2, 2022 | LAS VEGAS, NV



PRT307

SPONSORED BY CONFLUENT

Building real-time serverless data applications with Confluent and AWS

Joseph Morais

Cloud Solutions Architect
Confluent

Adam Wagner

Serverless Solutions Architect
AWS



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Agenda

Rise of data in motion

Confluent makes real-time data streams top priority

Event stream processing with AWS Lambda

AWS Lambda lets you build streaming applications faster, with less operational overhead

Event streaming with Confluent

Rearchitected Kafka, together with the features you need to rapidly deploy production use cases

Confluent and AWS Lambda integrations

Building event-streaming applications using ksqlDB and Lambda

Demo

Best practices when using Lambda as a stateless stream processor

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.



The rise of data in motion

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.



Data is the fuel for a modern business

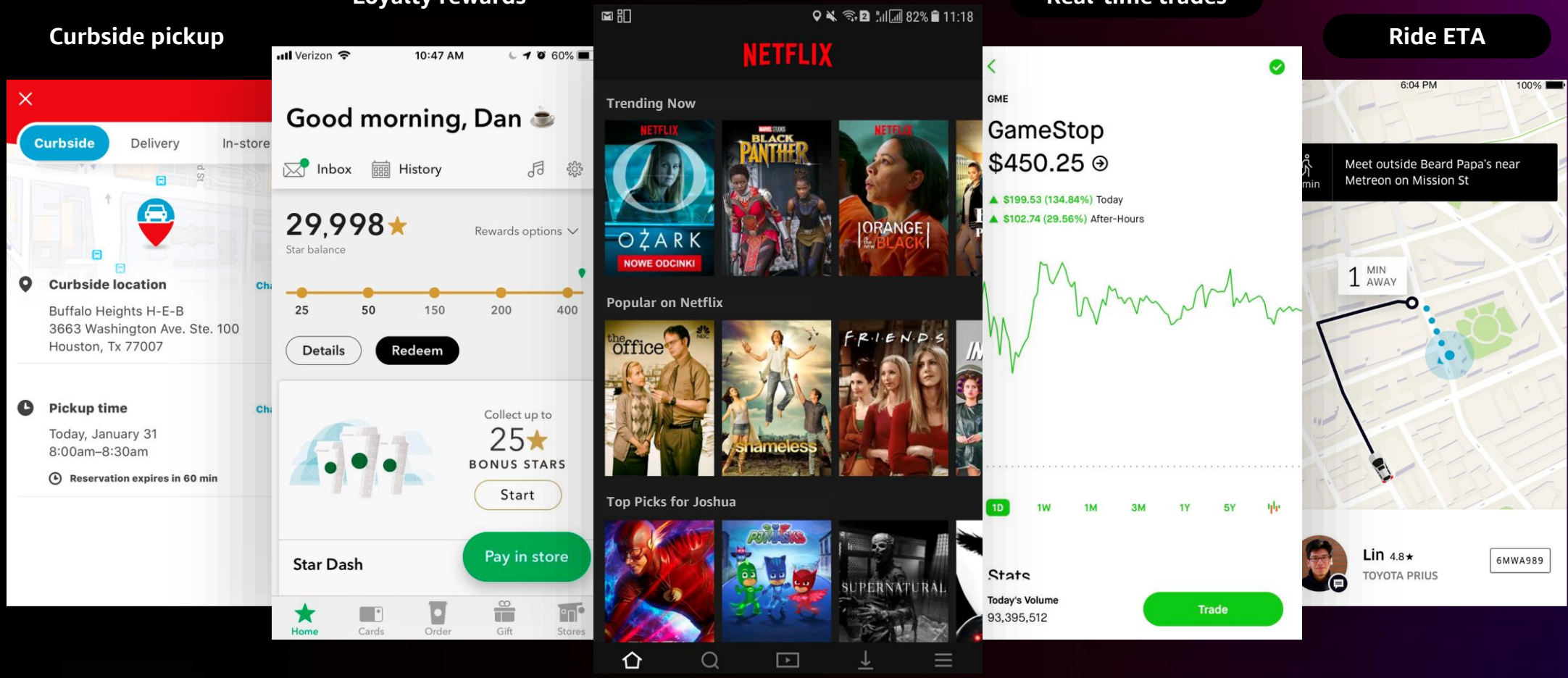
Curbside pickup

Loyalty rewards

Personalized recommendations

Real-time trades

Ride ETA

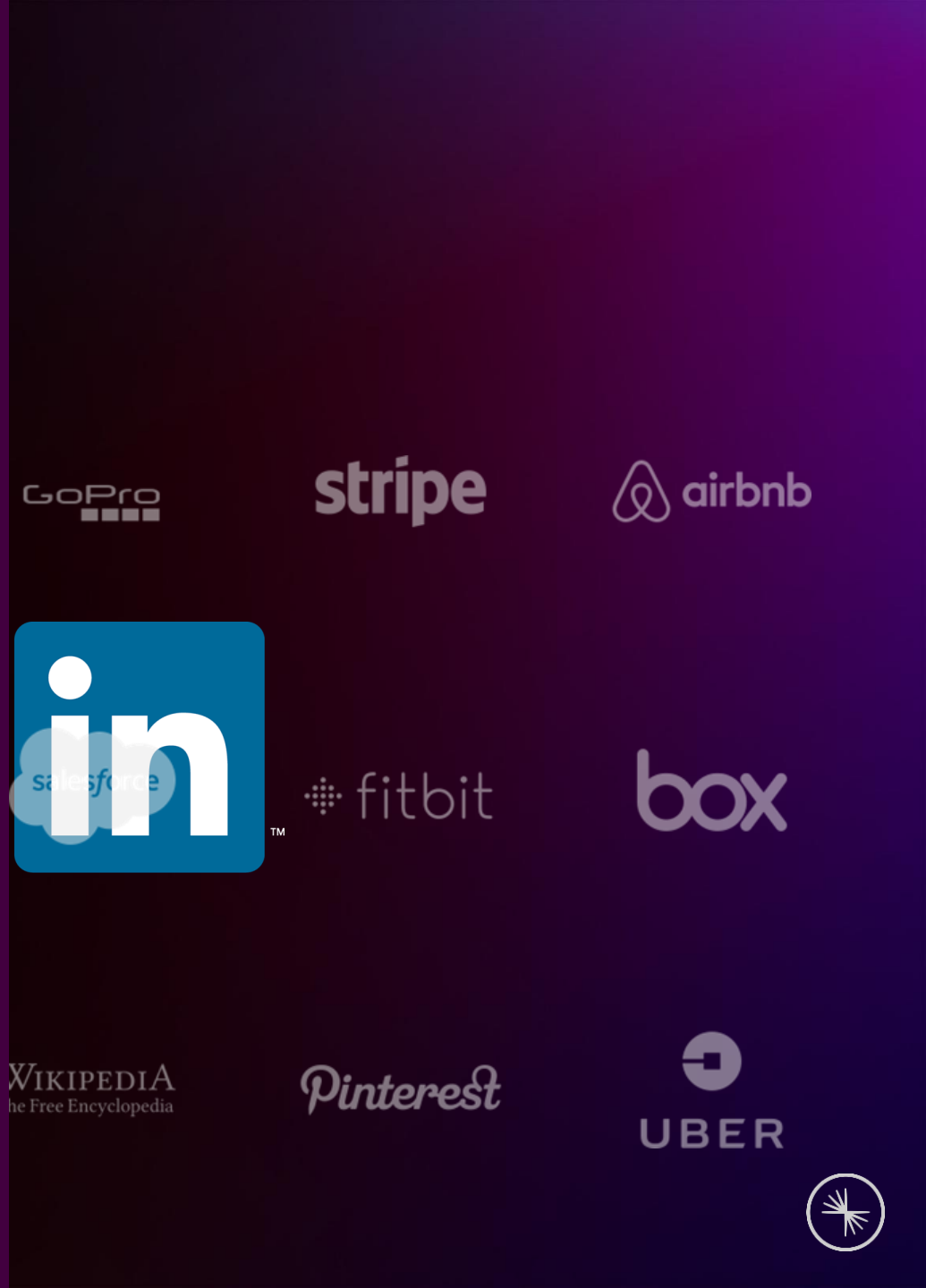


© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

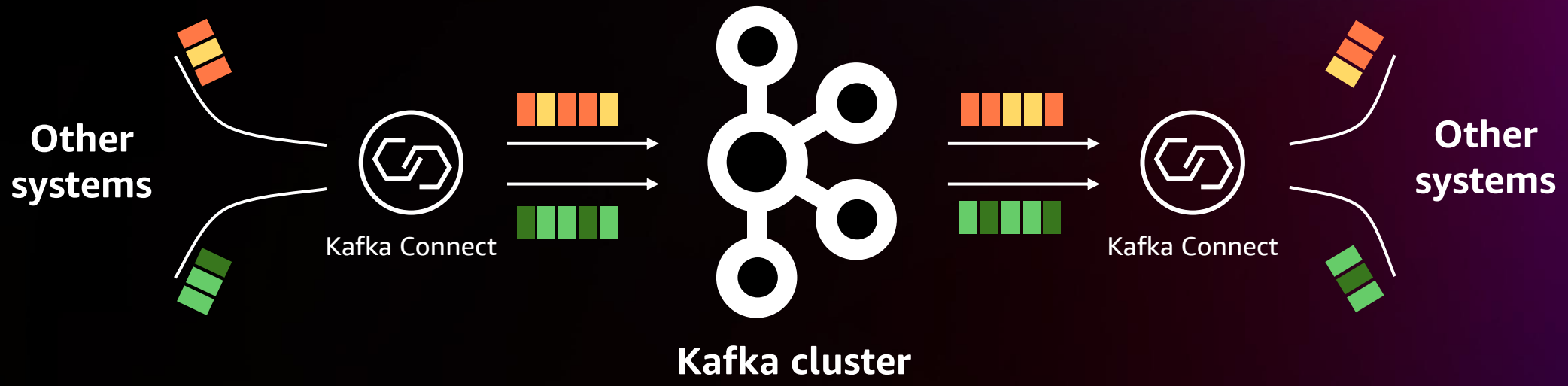
The rise of data in motion

70%

of fortune 500 companies using Apache Kafka
(majority are Confluent customers)



Apache Kafka is a data streaming platform



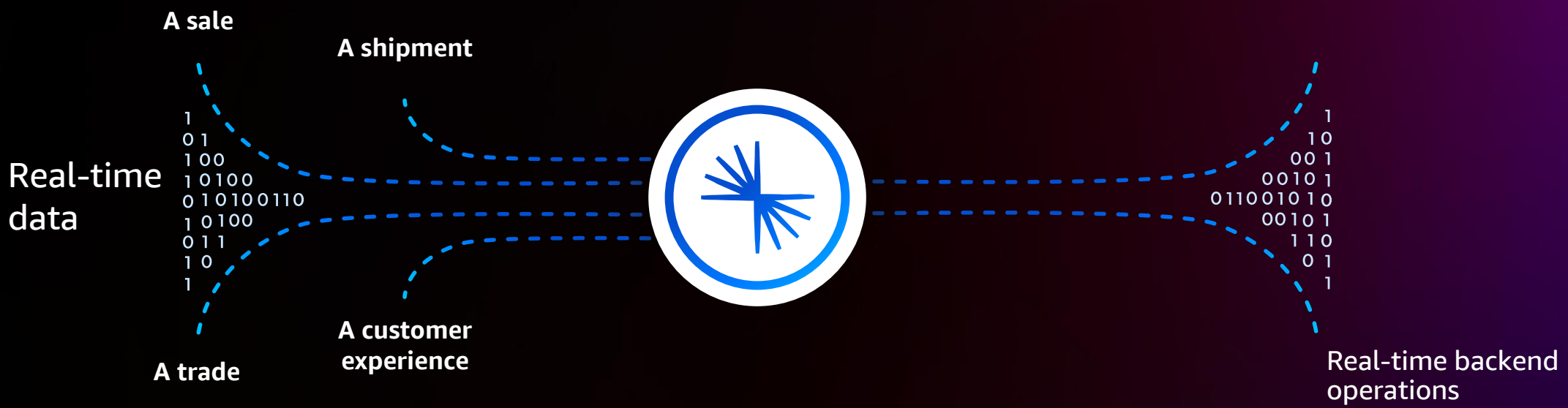
amazon S3 · Microsoft Azure Blob Storage · ORACLE · elastic · Apache Ignite · mongoDB · Solr · FTP · hazelcast · InfluxDB · hadoop · MQTT.ORG · cassandra · Apache Kudu · ATTUNITY · Apache HBase · Couchbase · Bloomberg · syslog-ng · ...many more

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Rich frontend customer experiences

CONFLUENT MAKES REAL-TIME DATA STREAMS TOP PRIORITY



citi “We need to shift our thinking from everything at rest to everything in motion.”

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Serverless stream processing with AWS Lambda

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.



Serverless applications

Function



Node.js
Python
Java
C#
Go
Ruby
PowerShell
Runtime API

aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.



Serverless applications

Event source





Function



Changes in data state



Requests to endpoints



Changes in Resource state



Node.js
Python
Java
C#
Go
Ruby
PowerShell
Runtime API

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.



Serverless applications

Event source



Changes in data state



Requests to endpoints



Changes in Resource state



Function



Node.js
Python
Java
C#
Go
Ruby
PowerShell
Runtime API

Services



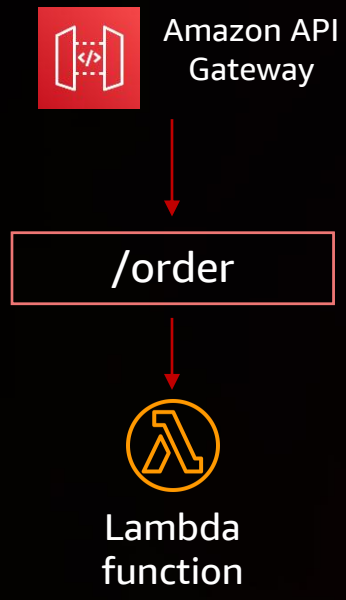


aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

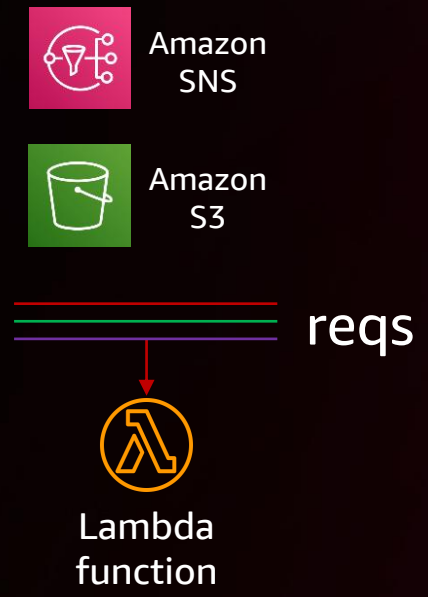


Lambda invocation models

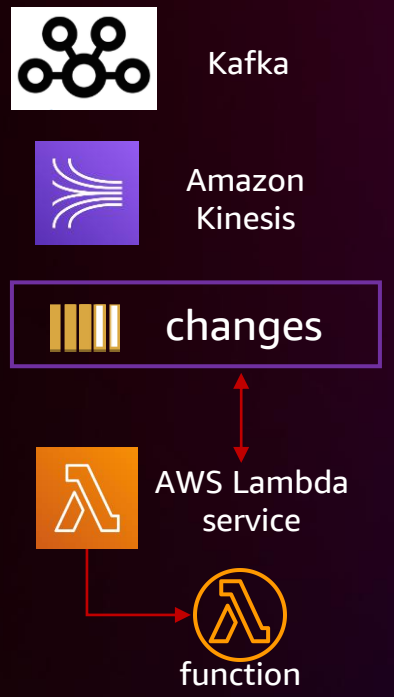
Synchronous (push)



Asynchronous (event)



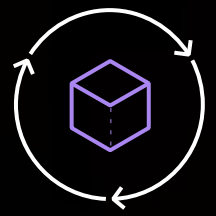
Stream (poll-based)

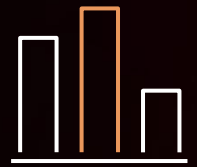


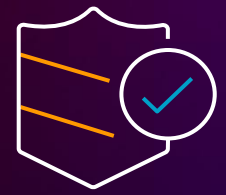
aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS serverless accelerates innovation

Lambda provides an easier way to process streaming data at scale







Faster time to market

Eliminate operational overhead

Automatic Scaling

Lower total cost of ownership

Built-in high availability & security

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Benefits of serverless stream processing



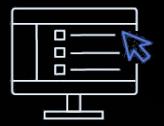
Build faster – Write less code

With AWS Lambda, organizations do not have to worry about operations like polling the streams for messages. Developers can focus on business logic instead of streaming specific tasks. Lambda takes care of polling with built-in pollers. No need to use any stream specific libraries or APIs to process/poll the streams.



Security and built in availability

Lambda runs your code on high-availability compute infrastructure with security isolation.



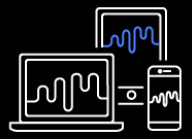
Reduce operational overhead

No need to worry about infrastructure management tasks like capacity provisioning and patching.



Lower TCO

Reduce costs by paying only for the compute time you use. Only run when there are events to process.



Automatic scaling

Scale up or down as needed from a few requests per day to thousands per second and ensure consistently low latency.

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Event streaming with Confluent

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

The Confluent advantage



Cloud native

Reimagined Kafka experience for the cloud



Complete

Enable developers to reliably & securely build next-generation applications faster



Everywhere

Be everywhere our customers want to be

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.



10x Kafka

Confluent Cloud offers a truly fully managed, cloud-native data streaming platform for Apache Kafka, with 10x faster scaling, infinitely more storage, and built-in resilience



Elasticity

Scale and shrink to handle 0 to GBps+ workloads and peak customer demands 10x faster and easier

Storage

Never worry about Kafka storage limits again with Infinite Storage that's 10x more scalable and performant

Resiliency

Leave Kafka reliability worries behind with 99.99% uptime SLA and 10x built-in durability

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Operationalizing Kafka on your own is difficult

Kafka is hard in experimentation. It only gets harder (and riskier) as you add mission-critical data and use cases.

Key challenges

Operational burden and resources

Manage and scale platform to support ever-growing demand

Security and governance

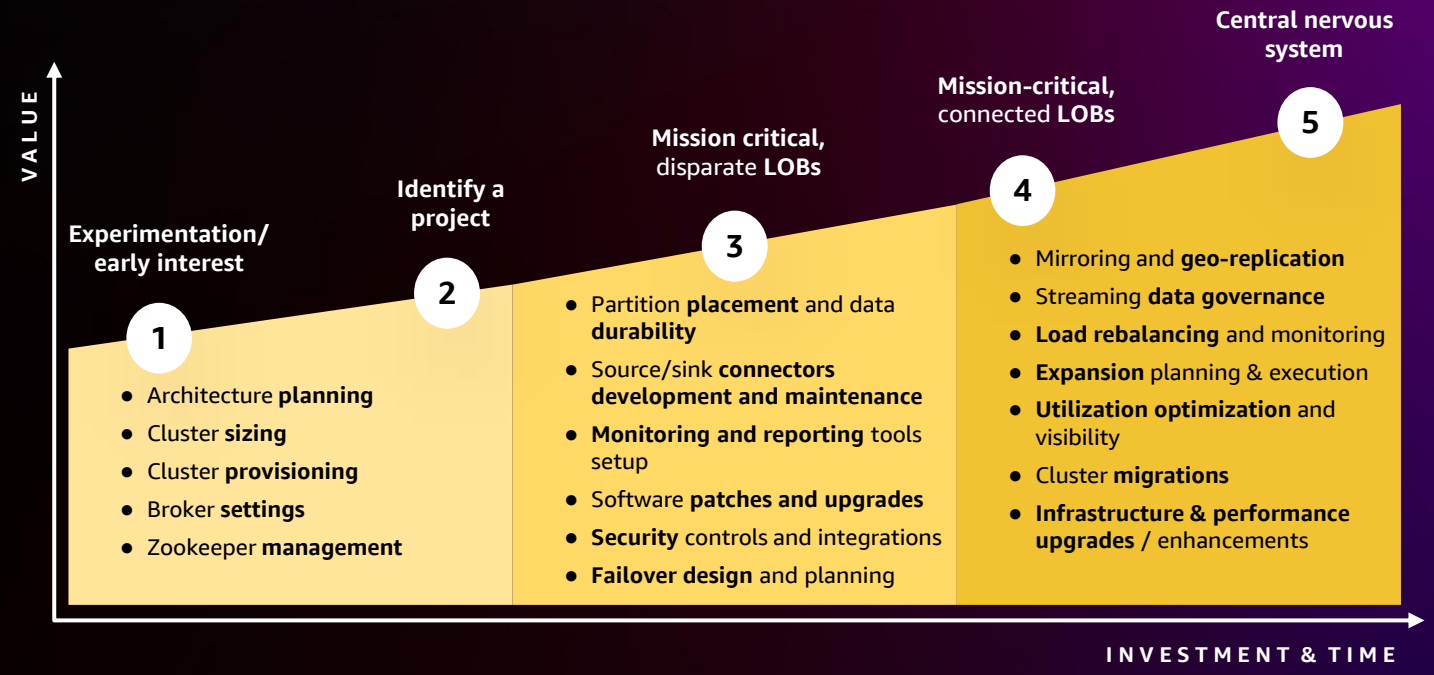
Ensure streaming data is as safe and secure as data-at-rest as Kafka usage scales

Real-time connectivity and processing

Leverage valuable legacy data to power modern, cloud-based applications and experiences

Global availability

Maintain high availability across environments with minimal downtime



nuuly
URBAN OUTFITTERS

"We are in the business of selling and renting clothes. We are not in the business of managing an event streaming platform… **If we had to manage everything ourselves, I would've had to hire *at least* 10 more people to keep the systems up and running.**"

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Confluent runs everywhere

SELF-MANAGED SOFTWARE

Confluent Platform

The enterprise distribution of Apache Kafka

In the data center



FULLY-MANAGED SOFTWARE

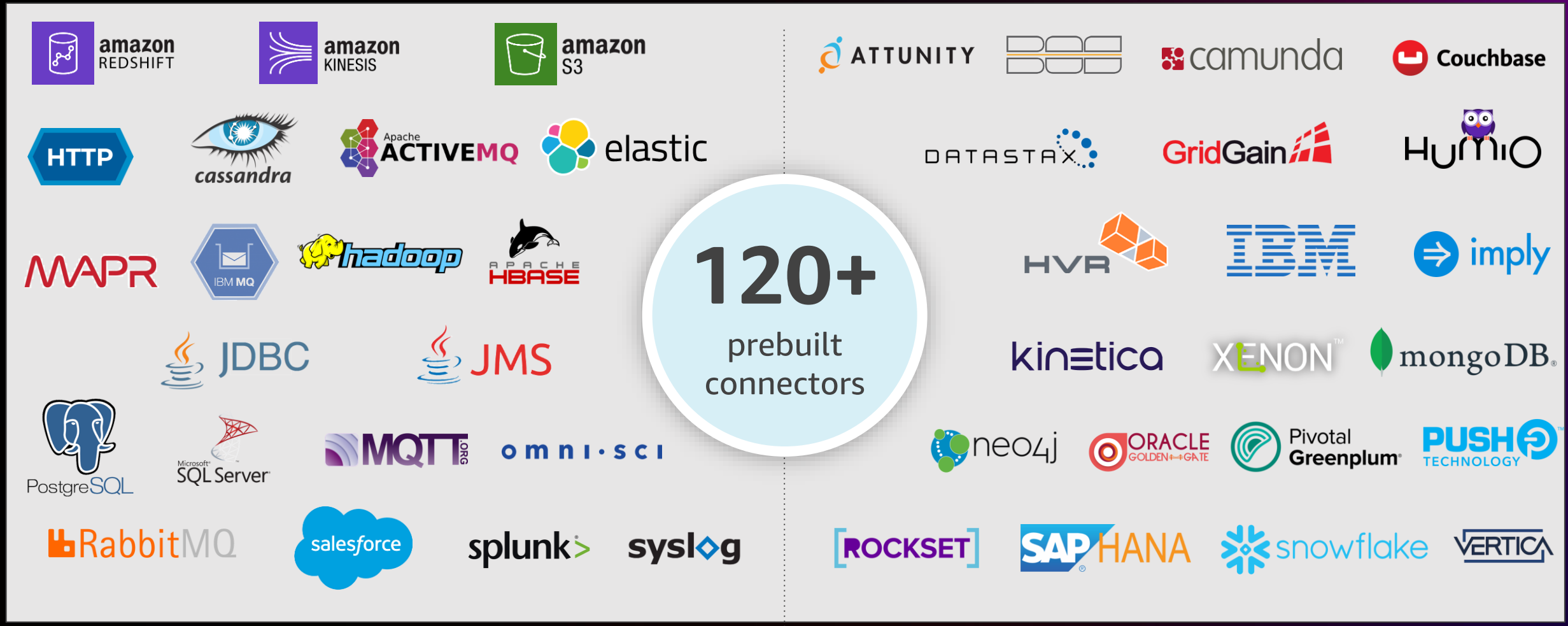
Confluent Cloud

Apache Kafka re-engineered for the cloud

In the cloud

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Instantly connect popular data sources & sinks



100+ Confluent supported

20+ partner supported, Confluent verified

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

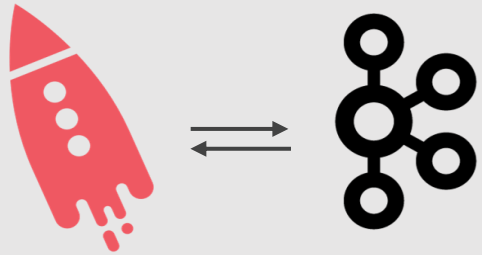
ksqlDB at a glance

What is it?

ksqlDB is an event-streaming database for working with streams and tables of data

All the key features of a modern streaming solution

Aggregations	**Joins**
Windowing	**Dual query support**
Event-time	**Out-of-order handling**
Exactly-once semantics	**User-defined functions**



```
CREATE TABLE activePromotions AS
  SELECT rideId,
       qualifyPromotion(distanceToDst) AS promotion
  FROM locations
  GROUP BY rideId
  EMIT CHANGES
```

How does it work?

It separates compute from storage, and scales elastically in a fault-tolerant manner

It remains highly available during disruption, even in the face of failure to a quorum of its servers

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

3 modalities of stream processing with Confluent

Kafka clients

```
ConsumerRecords<String, String> records =
consumer.poll(100);
Map<String, Integer> counts = new DefaultMap<String,
Integer>();
for (ConsumerRecord<String, Integer> record : records) {
  String key = record.key();
  int c = counts.get(key)
  c += record.value()
  counts.put(key, c)
}
for (Map.Entry<String, Integer> entry : counts.entrySet())
{
  int stateCount;
  int attempts;
  while (attempts++ < MAX_RETRIES) {
   try {
     stateCount = stateStore.getValue(entry.getKey())
     stateStore.setValue(entry.getKey(), entry.getValue() +
stateCount)
     break;
   } catch (StateStoreException e) {
     RetryUtils.backoff(attempts);
   }
  }
}
```

Kafka streams

```
builder
   .stream("input-stream",
         Consumed.with(Serdes.String(), Serdes.String()))
   .groupBy((key, value) -> value)
   .count()
   .toStream()
   .to("counts", Produced.with(Serdes.String(),
Serdes.Long()));
```

ksqlDB

```
SELECT x, count(*) FROM stream GROUP BY x EMIT
CHANGES;
```

Flexibility ⟷ **Simplicity**

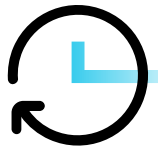
aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.



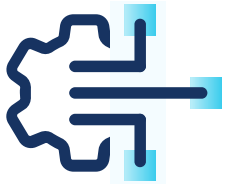
Rapidly build, test, and deploy data flows

to meet the changing data needs of the business



Get started in minutes

Use a point-and-click interface to connect sources and destinations and transform on demand



Future-proof your integrations

Evolve data flow and transformation logic by flexibly toggling between the GUI, code editor, or CLI



Accelerate real-time initiatives

Build modular, sharable and reusable data flows and collaborate on live streams with DevOps integrations and an open platform

Raiffeisen BANK

"Confluent's low-code, visual interface will enable more developers, across our entire organization, to leverage data in motion. With a unified, end-to-end view of our streaming data pipelines, it will improve our developer productivity by making real-time applications, pipeline development and troubleshooting much easier."

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Everywhere:
Cluster linking

Federated streaming,
hybrid and multi-cloud

Data syndication and replication across and between clouds and on premises, with self-service APIs, data governance, and visual tooling

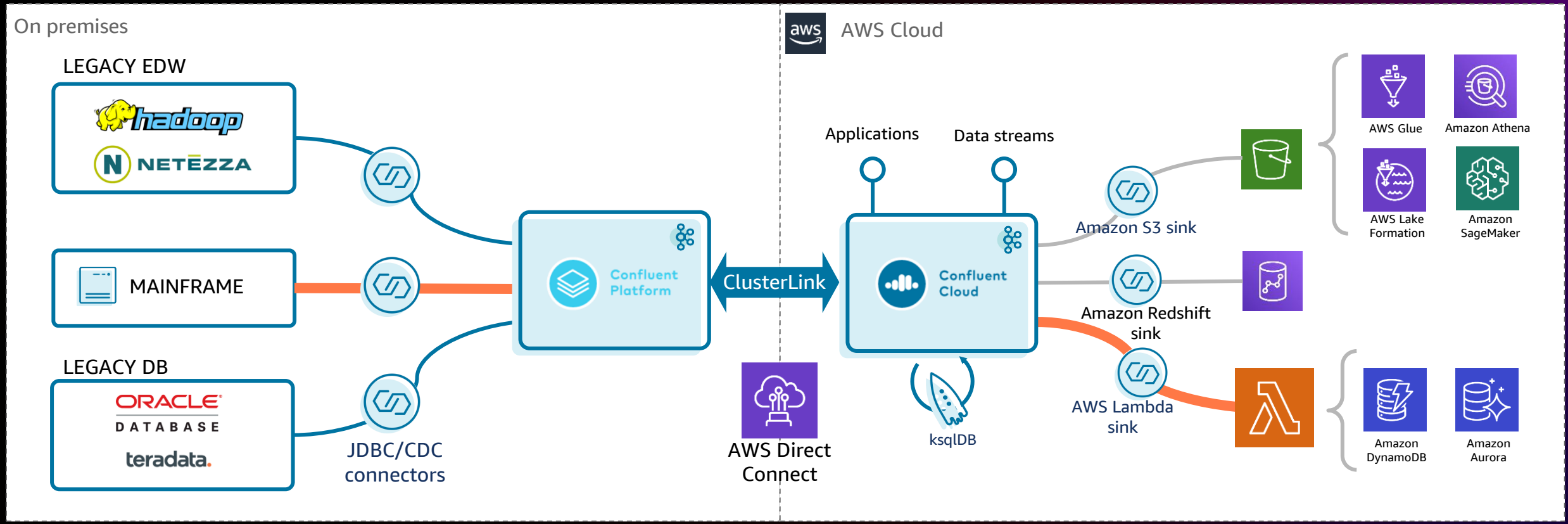
Reliable and real-time data streams between all customer sites, so you can run always-on streaming analytics on the data of the entire enterprise, despite regional or cloud provider outages



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Accelerate modernization from on premises to AWS

Connect	Bridge	Modernize
Leverage **+120 Confluent prebuilt connectors** to continuously bring valuable data from existing services on-premises, including enterprise data warehouse, databases, and mainframes	**Hybrid cloud** streaming with consistent, event-driven architecture for modern apps	**Increase agility** in getting applications to market and **reduce TCO** when freeing up resources to focus on value-generating activities and not in managing servers



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

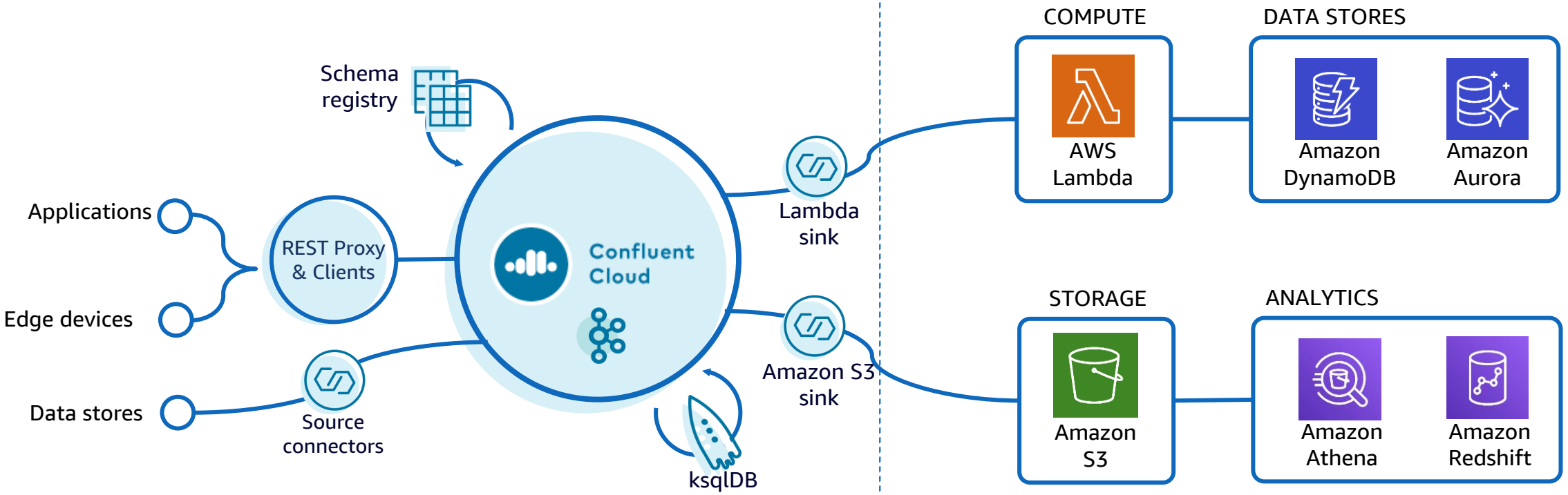
Increase developer agility and speed of innovation

Serverless integration

Connect existing and applications and data stores in a repeatable way without having to manage **Apache Kafka**, **Schema Registry** to maintain application compatibility, **ksqlDB** to develop real-time applications with SQL syntax, and **Connect** for effortless integrations with Lambda and data stores

AWS serverless platform

Stop provisioning, maintaining, or administering servers for backend components such as compute, databases, and storage so that you can focus on increasing agility and innovation for your developer teams



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

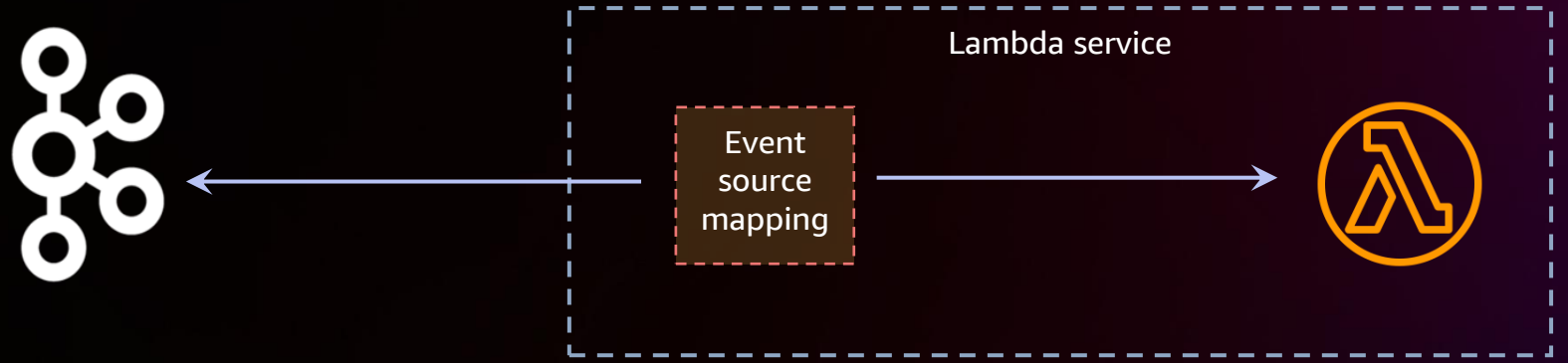
AWS Lambda
Confluent integrations

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Lambda consumer options



Confluent Kafka sink connector



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Confluent Lambda sink connector



- Sink connector polls Kafka partitions and invokes your function
- Lambda can be invoked synchronously or asynchronously
- At least once semantics
- Provides a dead letter queue (DLQ) for any failed invocations

aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

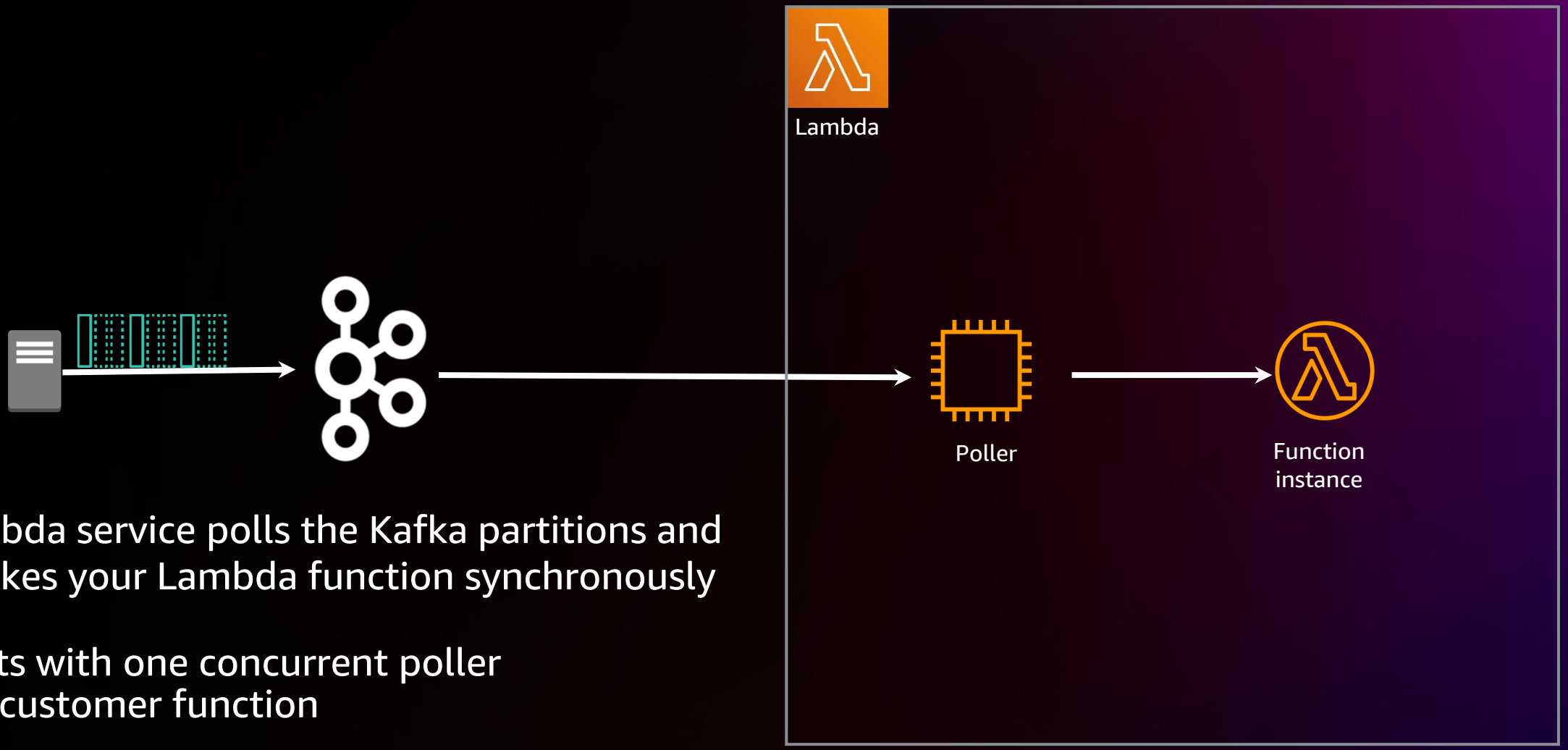
Confluent Lambda sink connector – Scaling and error handling



- Sink connector scales up to a soft maximum of 10 connectors
- Error handling semantics similar to sync and async Lambda invocations
  - Async: Lambda service retries twice (three total attempts)
  - Sync: By default, fails and stops processing for that partition
    Option to log to another Kafka topic and continue processing
- Option to batch records – Configured through aws.lambda.batch.size

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

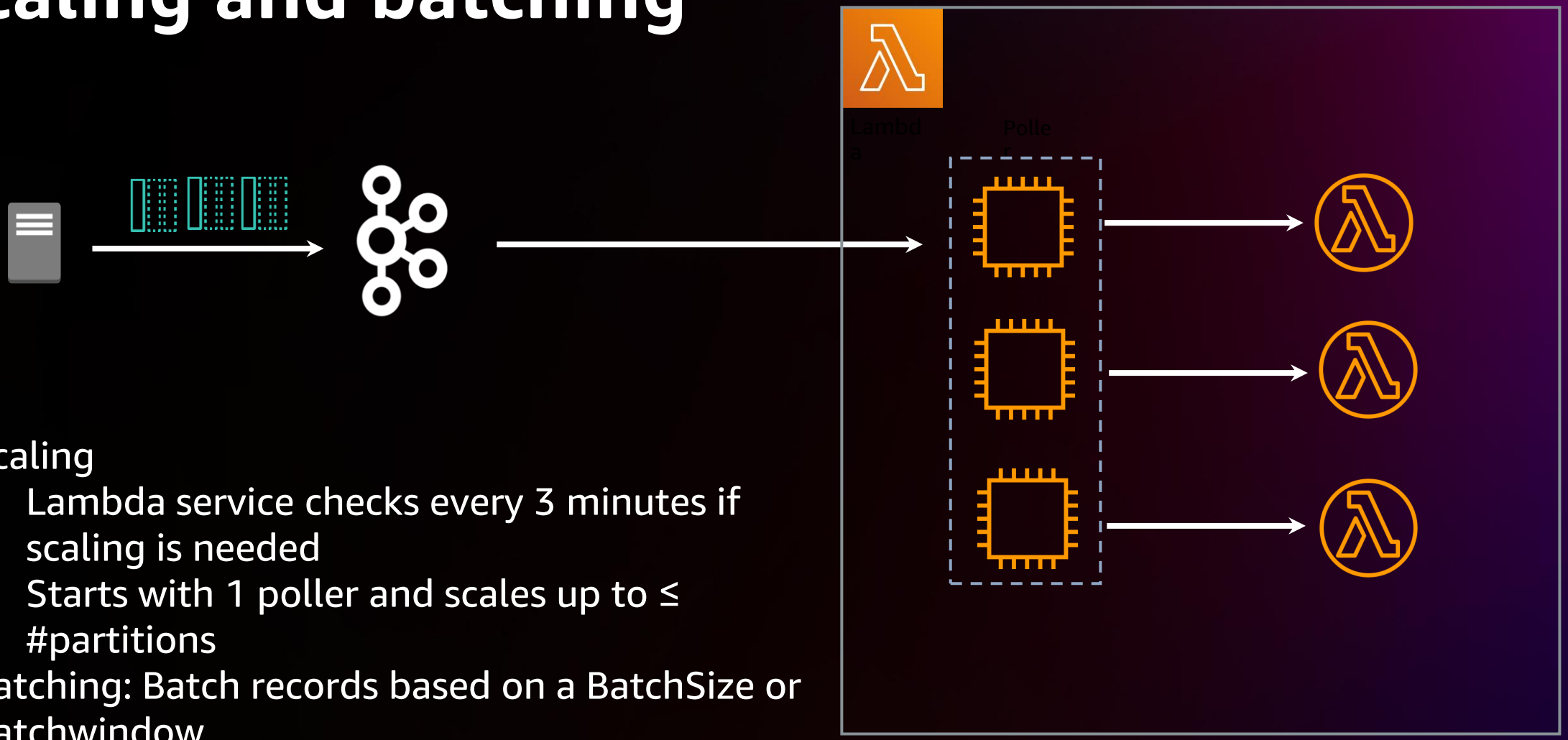
Lambda event source mapping (ESM) for Kafka



- Lambda service polls the Kafka partitions and invokes your Lambda function synchronously
- Starts with one concurrent poller and customer function

aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Lambda ESM consumer for Kafka – Scaling and batching



- Scaling
  - Lambda service checks every 3 minutes if scaling is needed
  - Starts with 1 poller and scales up to ≤ #partitions
- Batching: Batch records based on a BatchSize or Batchwindow

aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Lambda ESM consumer for Kafka – New features

- Offset lag Amazon CloudWatch metric
  - A measure of how many records behind the latest record, the Lambda function has processed
- Custom consumer group IDs
  - Allows you to specify the name of the consumer group that Lambda uses to consume from the topic
  - This can be a consumer group that’s already created
- Event filtering
  - Enables the filtering of messages by the ESM before batching them and sending to the Lambda function

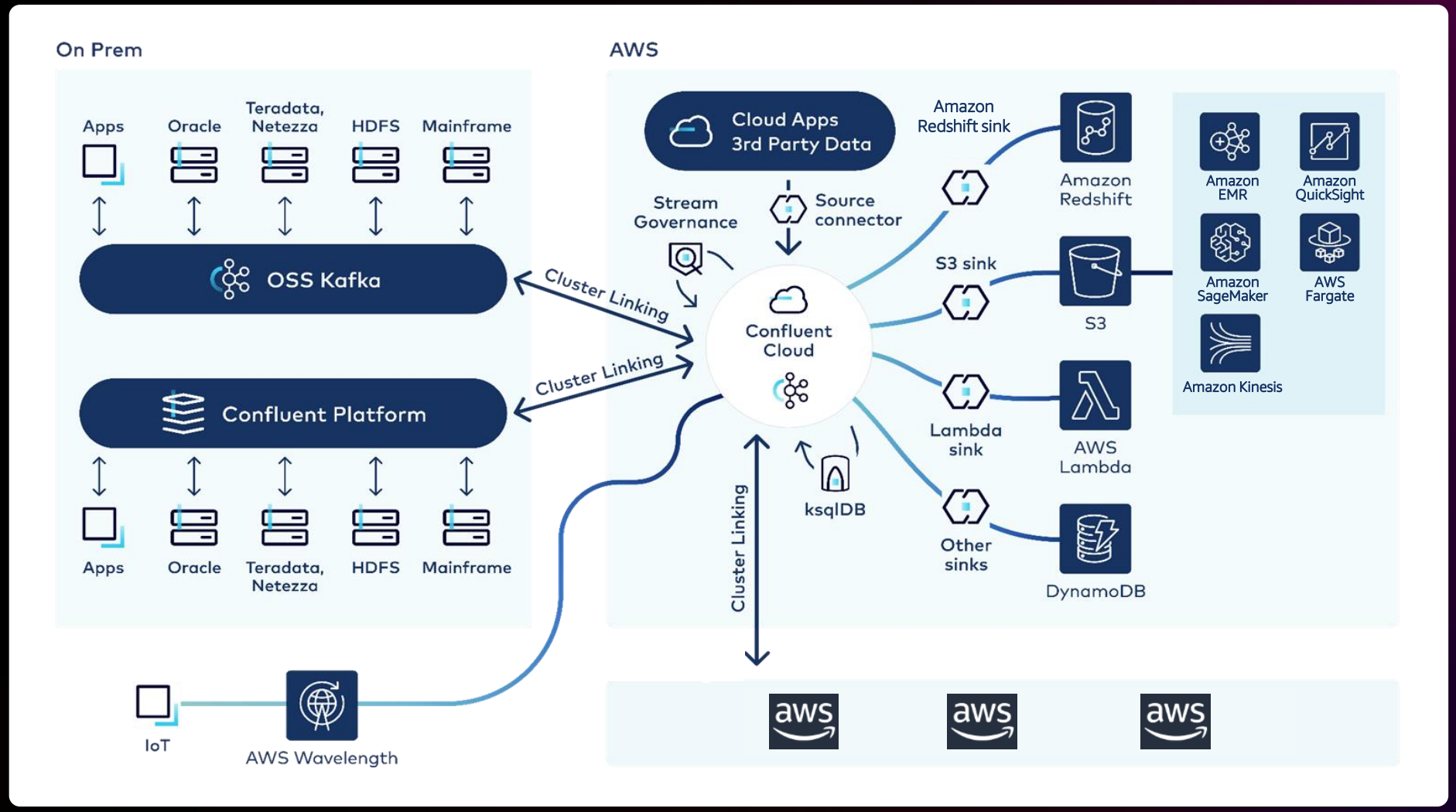
Offset lag blog

Custom consumer group ID blog

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Integrate all the things

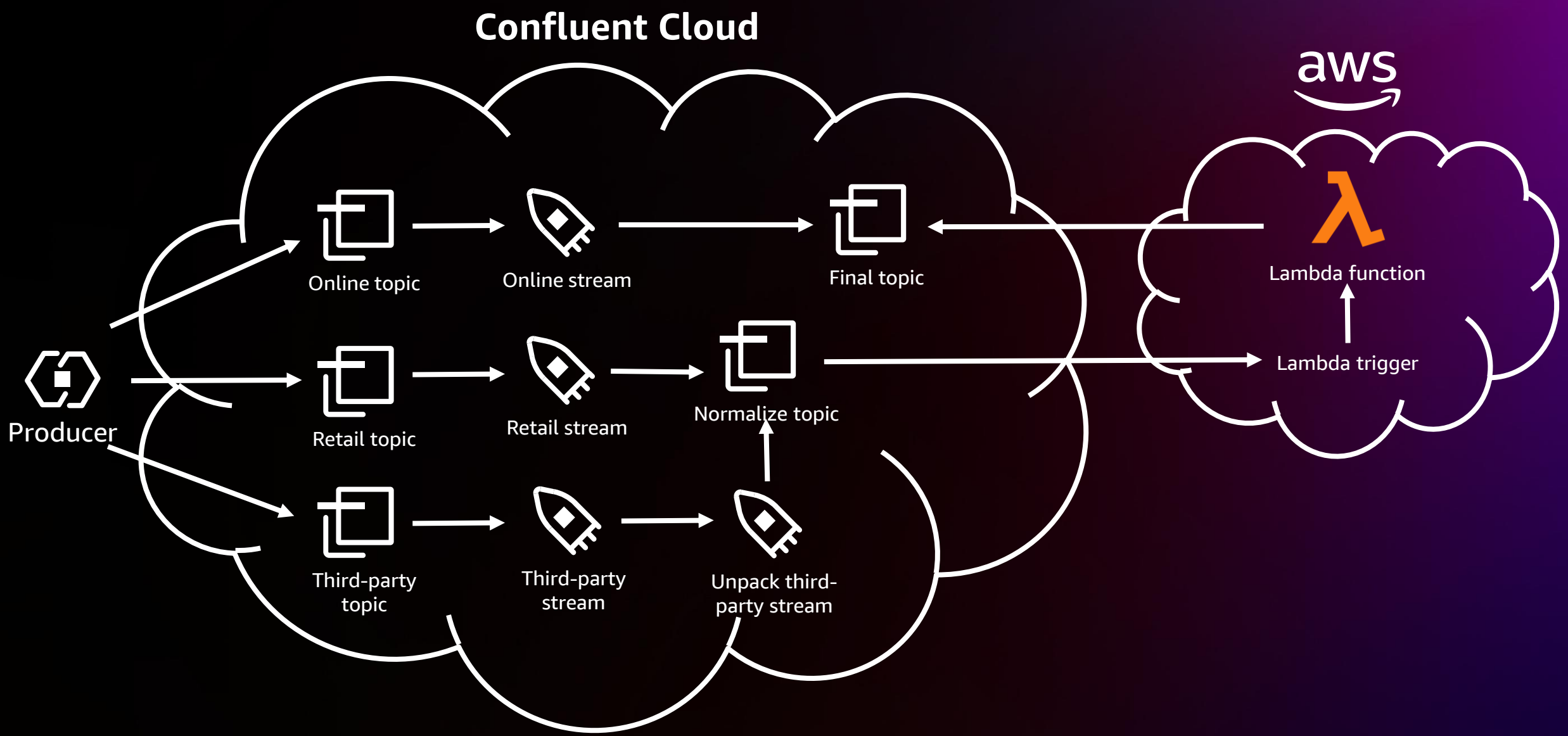
CONFLUENT OUTCOMES ON AWS



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Demo

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.



aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Want to learn more?

CONFLUENT AND SERVERLESS RESOURCES

- Confluent Kafka AWS workshops
  - confluent.awsworkshop.io
- Customer Use Case Webinar
  - Extend – Confluent + Lambda
  - Jan 21st 2023
- Confluent/AWS GameDays and Immersion Days
- **awsteam@confluent.io**
- serverlessland.com
  - Serverless examples
  - Blogs and video tutorials
  - Learning paths to help you get started





© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Thank you!



Please complete the session survey in the **mobile app**

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.