



Getting Started with Stream Processing and Confluent Cloud



This workshop is for developers who:

Want to develop a use case to evaluate whether
Confluent Cloud is suitable

Are new to Confluent Cloud and need some
guidance in setting it up



Today's Agenda

5 minutes

Intro

30 minutes

**Overview of Confluent Cloud,
Connect and ksqlDB**

45 minutes

Hands on workshop

3

15 minutes

Q&A



Intro to Stream Processing

Customers demand immediacy in every aspect of their lives through real-time applications

Transportation



Connect to a driver immediately after rideshare request

Retail



Instant notification upon package delivery

Technology



Real-time notification once a new patch is available

Banking



Automatic alert once fraudulent activity has been detected



These applications require reacting to events that happen in your business immediately

Events occur everywhere across an organization



A rider searches for an available rideshare



A valuable customer is beginning the checkout process



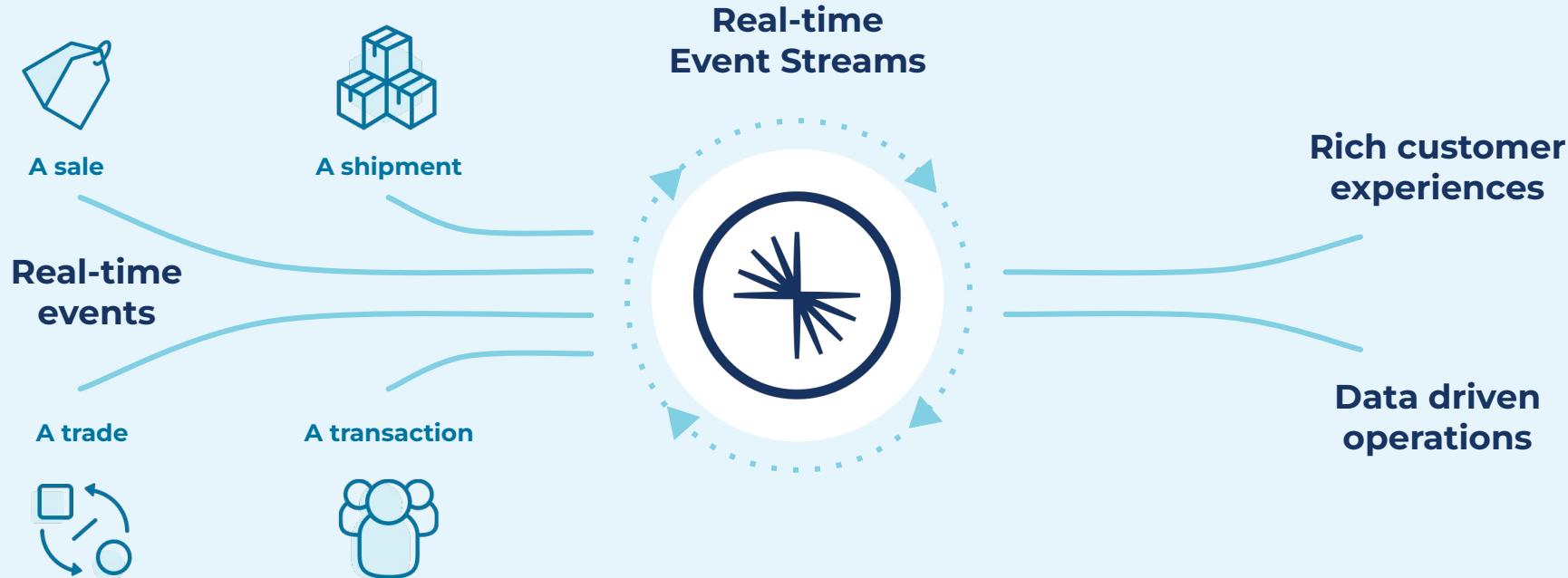
A new patch for bug fixes was released



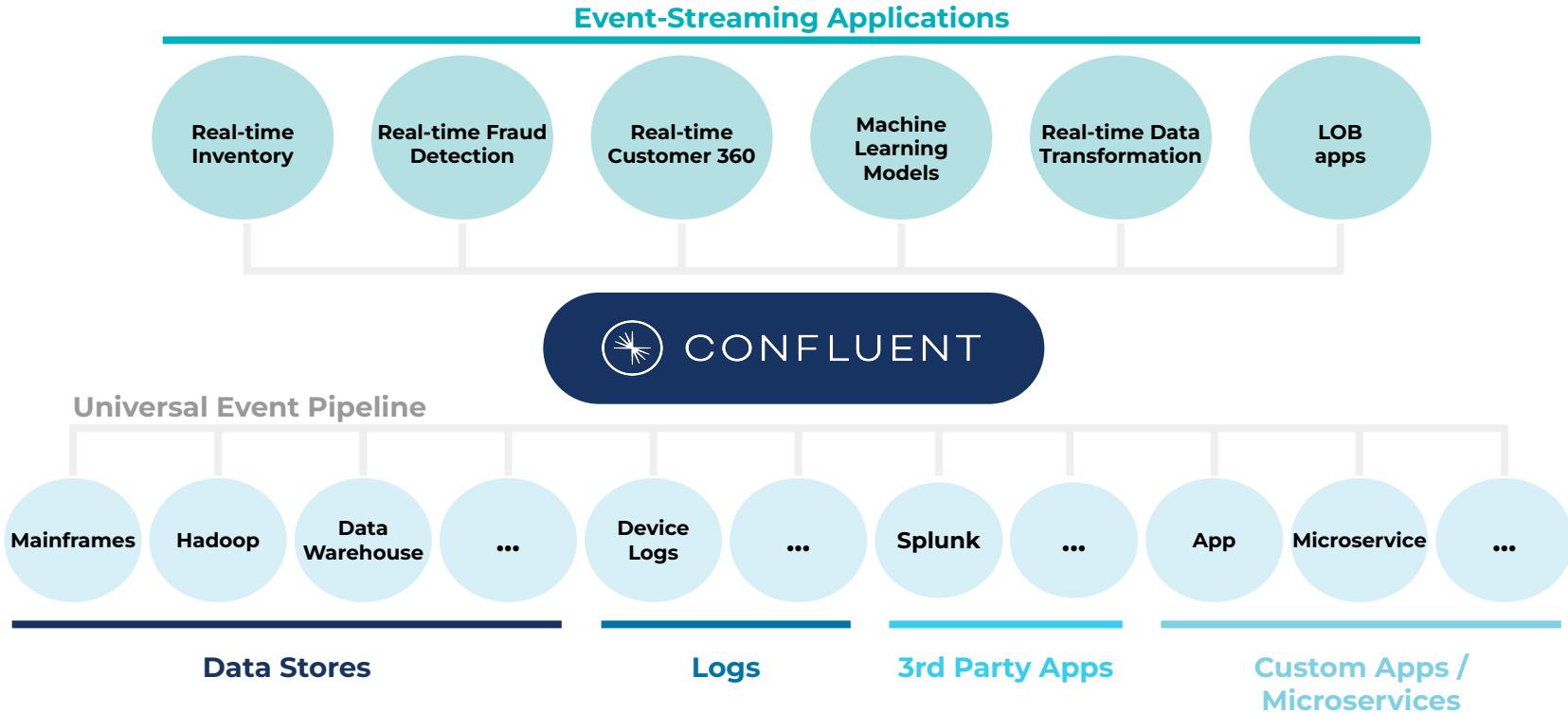
The same charge was rapidly made multiple times



This new paradigm shift, for data in motion, is centered around event streams



Confluent and Kafka for Event Streaming



The Power of Kafka and the Central Nervous System

>80% of Fortune 100 Companies are using Apache Kafka



intuit

Square

NETFLIX

Dropbox

GoPro

stripe

airbnb

coursera



LinkedIn

slack

salesforce

fitbit

box

yahoo!

PayPal



BLIZZARD
ENTERTAINMENT

WIKIPEDIA

Pinterest

Uber

Confluent is so much more than Apache Kafka



∞ **Complete:** Go beyond Kafka with all the essential tools for a complete data streaming platform

DEVELOPER

Unrestricted Developer Productivity

Multi-language Development
Non-Java Clients | REST Proxy | MQTT Proxy

Stream Processing & Integration
Connectors | ksqlDB | Stream Designer

OPERATOR

Efficient Operations at Scale

Management & Monitoring
Cloud UI | Metrics API | Control Center | Health+

Flexible DevOps Automation
Admin REST APIs | Terraform APIs | Confluent for K8s | Ansible Playbooks

ARCHITECT

Production-stage Prerequisites

Enterprise-grade Security
RBAC | Audit Logs | Encryption | BYOK | Private Networking

Stream Governance
Schema Registry | Schema Validation | Stream Lineage | Stream Catalog

EXECUTIVE

Partnership for Business Success

Complete Engagement Model
Data in Motion Blueprint

Business Case Justification
TCO | ROI | Risk



Cloud Native: Apache Kafka®, fully managed and re-architected to harness the power of the Cloud

Elastic Scalability

Expand | Shrink | Self-Balancing Clusters

Infinite Storage

Infinite Storage | Tiered Storage

High Availability

99.99% SLA | Multi-AZ Clusters | Multi-Region Clusters



Everywhere: Connect your data in real time with a platform that spans from on-prem to cloud and across clouds

Fully managed cloud service

AWS | Azure | GCP

Self-managed software

Kubernetes | VMs | Bare Metal

Hybrid and Multicloud

Cluster Linking | Replicator



Enterprise Support



Professional Services

Committer-driven
Expertise



Training



Partners



Traditional ETL = Legacy Data Integration

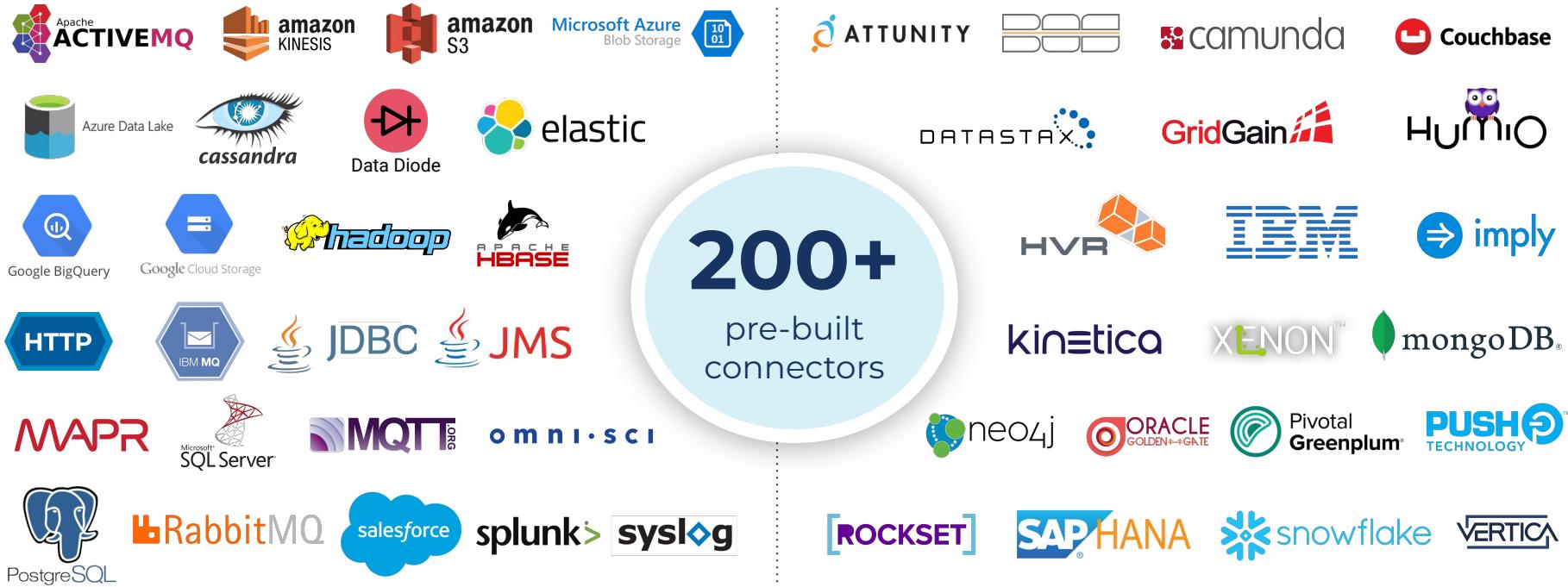
Daily movement of data from operational data stores to a data warehouse



Real Time Stream Processing



Connect Popular Data Sources & Sinks



200+

pre-built
connectors

90+ Confluent Supported

70+ Partner Supported, Confluent Verified



Easily Build Real-Time Applications

ksqldb at a Glance

ksqldb is a database for building real-time applications that leverage stream processing

Aggregations

Joins

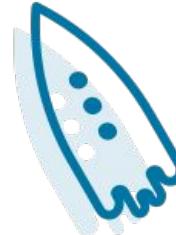
Filters

User-Defined Functions

Push & Pull Query Support

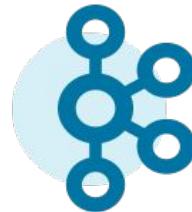
Embedded Connectors

ksqldb



Compute

Kafka

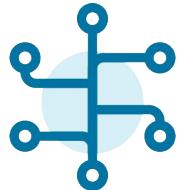


Storage

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

Build a complete real-time application with just a few SQL statements

ksqlDB Example Use Cases



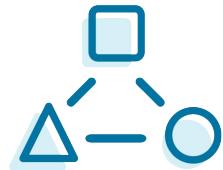
Data exploration



Data enrichment



Streaming ETL



Filter, cleanse, mask



Real-time monitoring



Anomaly detection

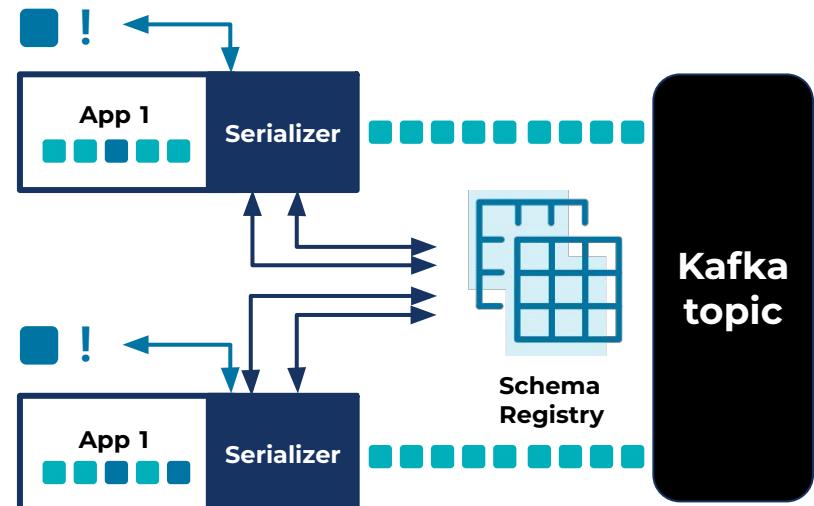


Schema Registry



Schema Registry

- **Enable App Dev Compatibility**
- **Develop using standard schemas**
- **Store and share** a versioned history of all standard schemas
- **Validate data compatibility** at the client or connector level
- **Reduce operational complexity**
- **Avoid time-consuming** coordination among developers to standardize on schemas





Hands-On Workshop

Sign Up for a Confluent Cloud Account

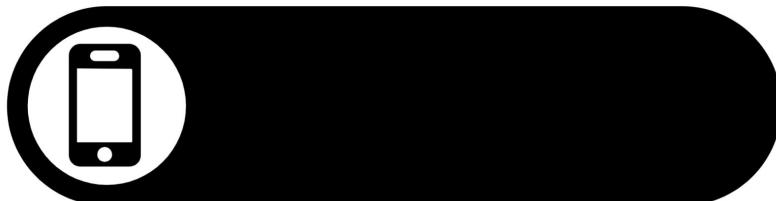


- Sign up (**Link** is provided in the chat box)
- When you sign up for a new Confluent Cloud account, you will get up to \$400 in credits to use during the first 30 days
- Note: You will create resources during this workshop that will incur costs. The \$400 credit will cover the cost of resources created during the workshop. Make sure to tear down your infrastructure so you don't accrue extra charges.

Lab Guide: <https://shorturl.at/nsDP1>

Lab Guide

Lab Guide: <https://shorturl.at/nsDP1>

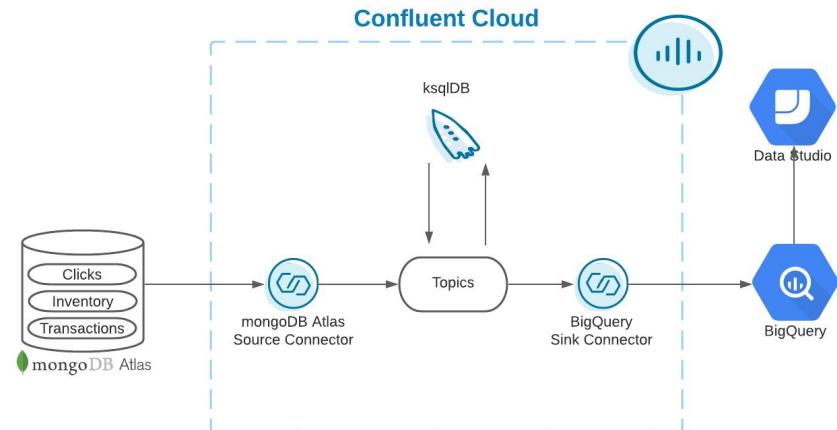


Workshop Flow

1. Log Into Confluent Cloud
2. Create an Environment and Cluster
3. Created Schema Registry Instance
4. Setup ksqlDB
5. Create Topic with Confluent Cloud UI
6. Create an API Key Pair
7. Connect mongoDB Atlas source to Confluent Cloud
8. Cloud Dashboard Walkthrough
9. Create Streams and Tables using ksqlDB
10. Stream Processing with ksqlDB
11. Connect BigQuery sink to Confluent Cloud
12. Consume Data from a topic (Python App)*
13. Clean Up Resources
14. Confluent Resources and Further Testing

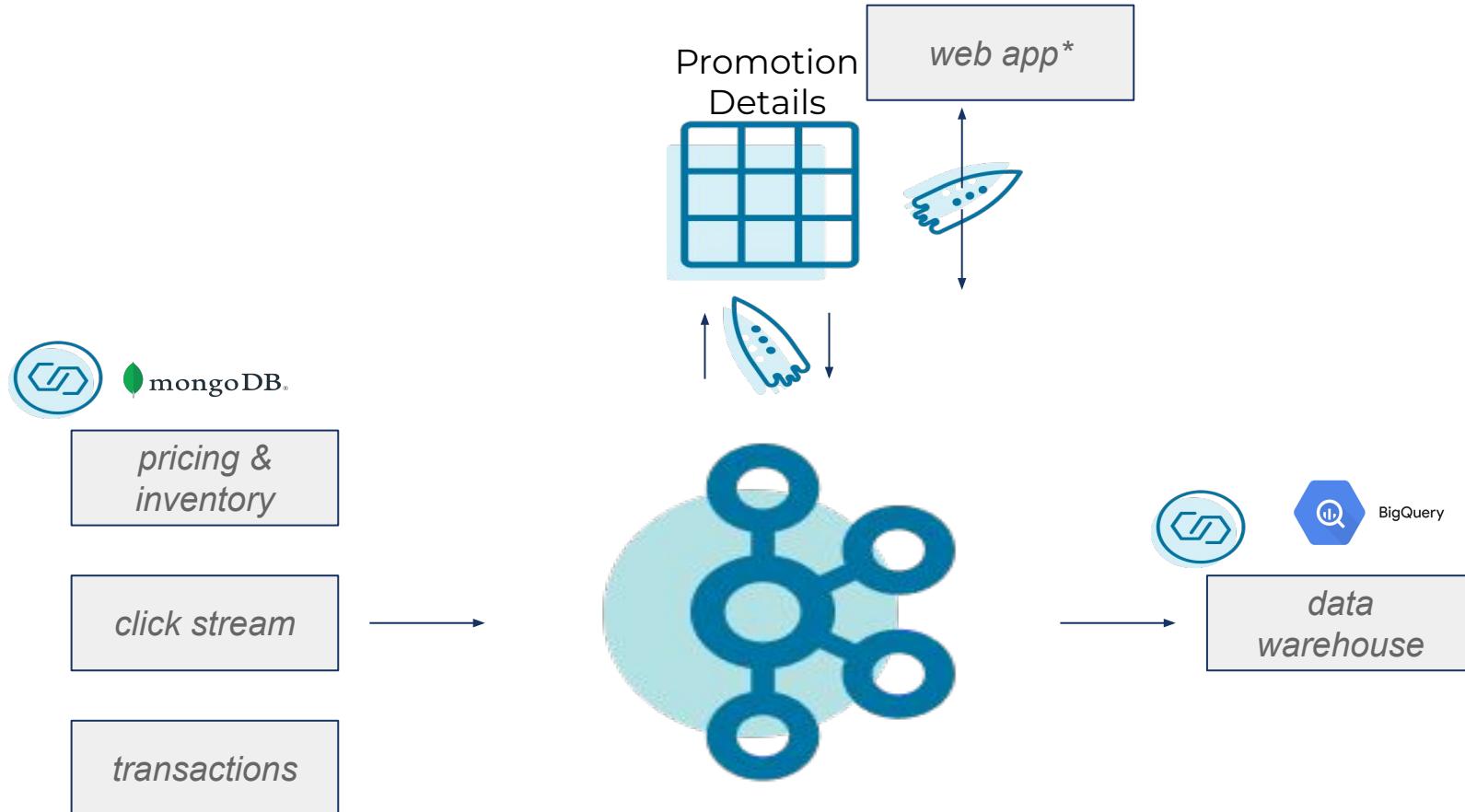


Architecture Diagram





Which Products to create Promotions for?





ksqldb Constructs



Streams & Tables

Stream

- Unbounded, continuously updating data set
- Useful for representing a historical series of events
- “Data in motion”

Table

- Represent the latest version of each value per key
- Serve queries to applications

Queries



Persistent Queries

- Execute continuous computations over unbounded streams of events
- Derive new collections or materialized views that are incrementally updated

Pull Queries

- Tells you point in time value
- Good fit for request/response flows

Push Queries

- Tells you all change values
- Emit refinements that quickly react to new information
- Perfect fit for asynchronous applications flows



Customer Case Studies



Challenge

Challenged with legacy, slow-moving and fragmented systems, the engineering teams at Lululemon needed to transform from a technology perspective to keep up with the business' unprecedented growth and global expansion.

Solution

Leveraging event streaming with Confluent Cloud + AWS to move from slow, manual processes to a business driven by real-time information.

Results

- Lululemon has streamlined and automated its costing projects with partners and manufacturers, which were previously being done in Excel spreadsheets.
- Implemented an app called FLOW, which optimizes merchandising.
- Working towards real-time inventory availability, which will be reduced from 35-minute latency to less than 1 minute across all channels.





Challenge

Maximize customer satisfaction and revenue growth by creating a hyper-personalized online retail experience, turning each customer visit into a one-on-one marketing opportunity.

Solution

Use Confluent to combine historical customer data with real-time digital signals from customers, generating hyper-personalized content – for example, targeted special offers – which is inserted in real time back into the customer's session.

- Schema Registry, KSQL, Lambda Connector

Results

- Real-time hyper-personalization of the customer experience
- Increased customer conversions
- Accelerated innovation
- Confluent Cloud frees up developers' time

“Our hyper-personalized approach is delivering measurable results. In our A/B testing, we've seen a significant increase in customer conversion rates. That's proof that our decision to adopt a real-time event streaming approach was the right one. I expect even bigger benefits as we continue to grow our capabilities.” — **Jon Vines**,

Software Development Team Lead at AO.com

A background image showing a delivery truck for AO.com. The truck is green and white, with the AO logo and the word 'Unbeatable price' visible. A delivery person in a dark polo shirt is holding a cardboard box with the AO logo on it. The scene is outdoors with trees in the background.



Hands-on Workshop

Lab Guide

- [Lab Guide](#)



Enable Stream Governance Essentials



Enable Stream Governance Essentials

Select the cloud provider and region where you want the environment Schema Registry and Stream Catalog to run and metadata to be stored. [Learn more ↗](#)

The cloud provider and region cannot be changed once you enable the environment package.



aws



Google Cloud

Region*

Sydney (australia-sout... ▾



Microsoft
Azure



MongoDB Atlas Database Details

MongoDB Atlas database details

The MongoDB hostname address must provide a service record (SRV). A standard connection string will not work.

Connection host* —

abc.r3iy3qr.mongodb.net

The connection user must have privileged action "find" to query the MongoDB database.

Connection user* —

dbUser

Connection password* ⓘ —

.....

Database name ⓘ —

abc

Database details

Collection name ⓘ —

Topics After MongoDB connector is Active



Topics

Topics								
Topic name	Tags	Partitions	Production	Consumption	Retained bytes	Consumers	Schema	Action
abc.clicks	--	1	--	--	--	--	Set a schema	
abc.inventory	--	1	--	--	0B	--	Set a schema	
abc.transactions	--	1	--	--	--	--	Set a schema	
pksqlc-dow3xo-processing-log	--	8	--	--	0B	--	Set a schema	



Create Streams

Editor Flow Streams Tables Persistent queries Performance Settings CLI instructions

```
1 CREATE STREAM clicks(
2     ip VARCHAR,
3     userid INT,
4     prod_id INT,
5     bytes BIGINT,
6     referrer VARCHAR,
7     agent VARCHAR,
8     click_ts BIGINT
9 )
10 WITH (
11     KAFKA_TOPIC='abc.clicks',
12     VALUE_FORMAT='JSON',
13     TIMESTAMP='click_ts'
14 );
```

● Add query properties

auto.offset.reset = Earliest

+Add another field

Stop Run query

```
1 v {
2     "@type": "currentStatus",
3     "statementText": "CREATE STREAM CLICKS (IP STRING, USERID INTEGER, PROD_ID INTEGER, BYTES BIGINT, REFFERER STRING, AGENT STRING, CLICK_TS BIGINT) WITH (CLEANUP_POLICY='delete',
4     KAFKA_TOPIC='abc.clicks', KEY_FORMAT='KAFKA', TIMESTAMP='click_ts', VALUE_FORMAT='JSON');",
5     "commandId": "stream/CLICKS/create",
6     "commandStatus": {
7         "status": "SUCCESS",
8         "message": "Stream created",
9         "queryId": null
10    },
11     "commandSequenceNumber": 2,
12     "warnings": []
13 }
```

CONFIDENTIAL



Q&A



Additional Resources

- [Getting started with Event Streaming](#)
- [Confluent Cloud Documentation](#)
- [Quickstart](#) with Confluent Cloud
- [Quickstart](#) with ksqlDB in Confluent Cloud
- [Confluent Hub](#) - All fully managed and self managed connectors
- [KSQL Recipes Guide](#) - ksqlDB examples and tutorials
- [Best Practices for Developing Apache Kafka Applications on Confluent Cloud](#) - Get started with connecting your applications to Confluent Cloud



CONFLUENT