Building Streaming Applications with Kafka and Upstash

Frank Kane

Sundog Education

upstash

What is streaming?

- How does new data get into your cluster? Especially if it's "Big data"?
  - New log entries from your web servers
  - New sensor data from your IoT system
  - New stock trades
- Streaming lets you publish this data, in real time, to your cluster.
  - And you can even process it in real time as it comes in!

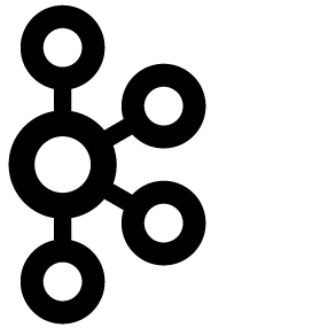
Two problems

How to get data from many different sources flowing into your cluster

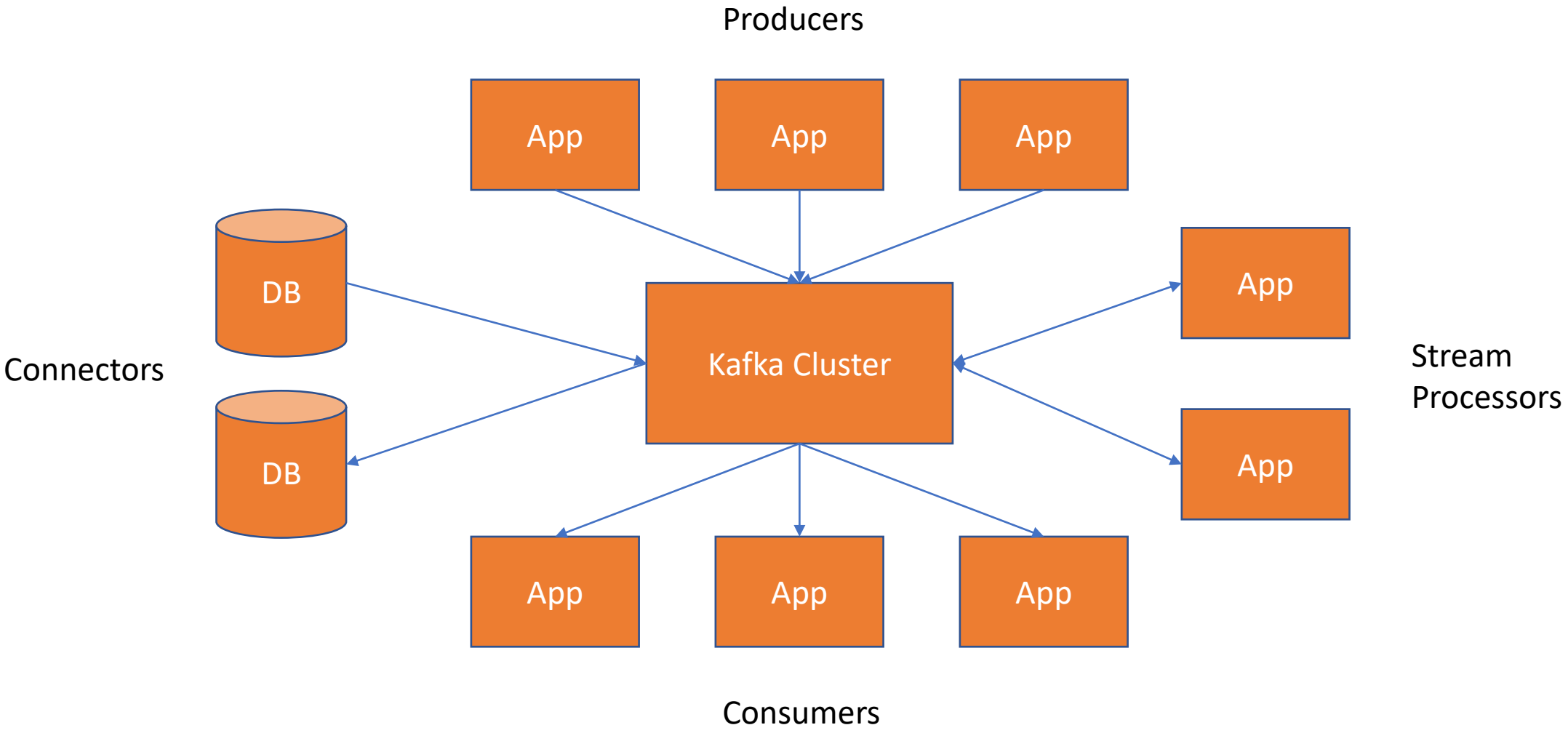
Processing it when it gets there

Enter Kafka

- Kafka is a general-purpose **publish/subscribe messaging system**
- Kafka servers store all incoming messages from *publishers* for some period of time, and *publishes* them to a stream of data called a *topic*.
- Kafka *consumers* subscribe to one or more topics, and receive data as it's published
- A stream / topic can have many different consumers, all with their own position in the stream maintained



Kafka architecture



How Kafka scales

- Kafka itself may be distributed among many processes on many servers
  - Will distribute the storage of stream data as well
- Consumers may also be distributed
  - Consumers of the same group will have messages distributed amongst them
  - Consumers of different groups will get their own copy of each message

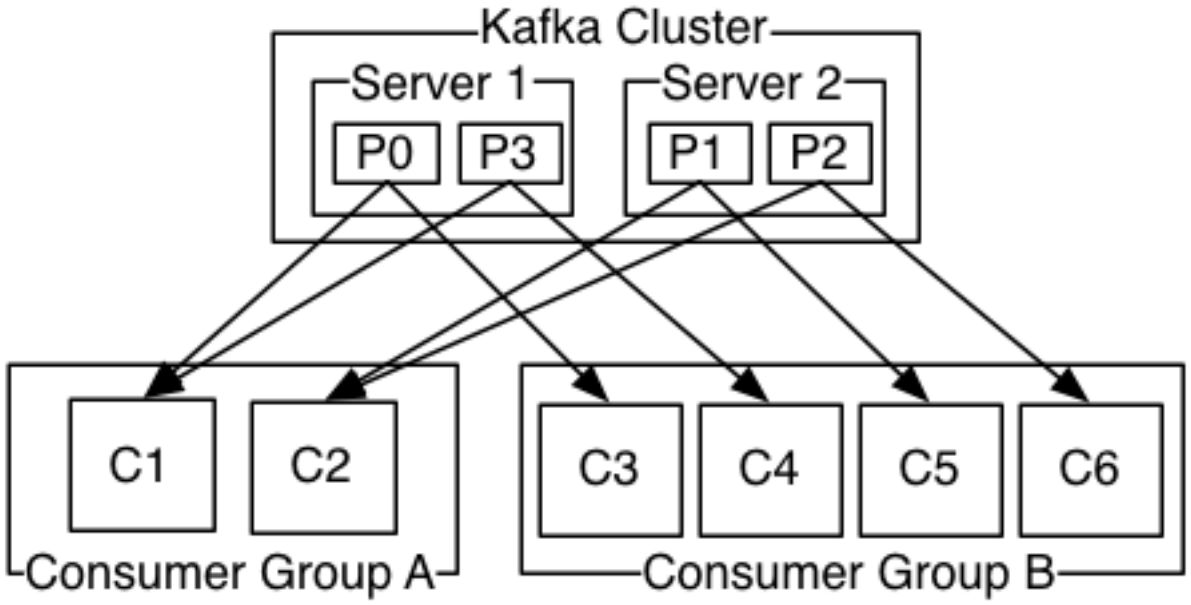


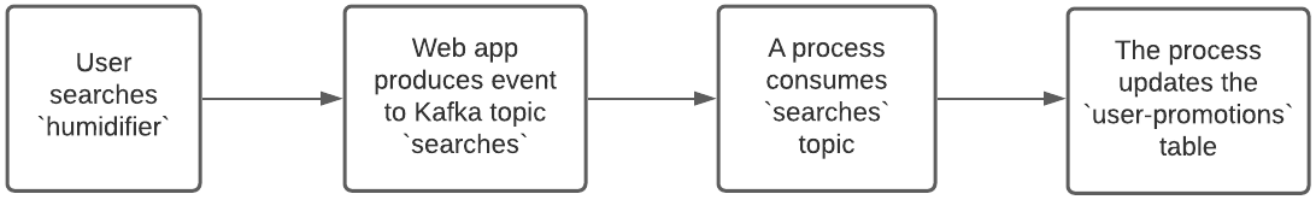
Image: kafka.apache.org

Sample Kafka Use-Cases

Web applications with Kafka and Upstash

Website activity tracking

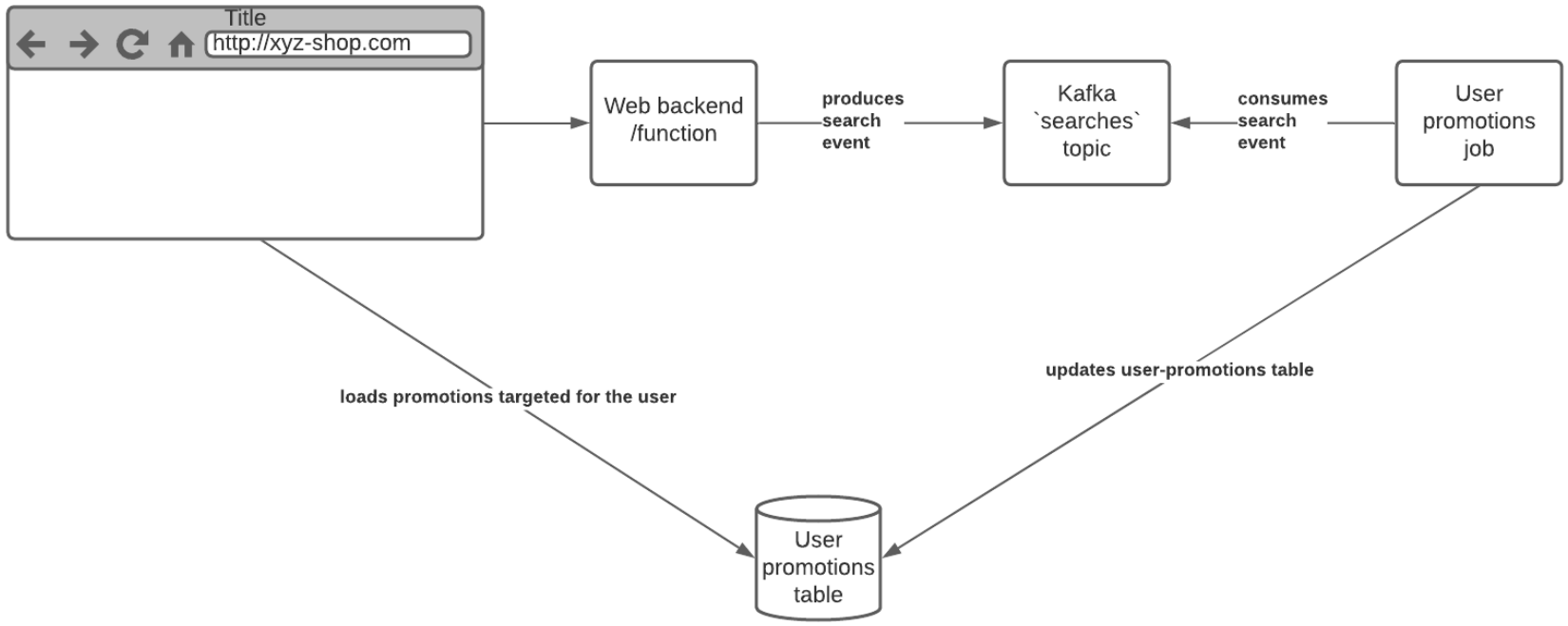
- Send clicks, searches, scrolls, form submissions etc. to different Kafka topics
- Example: Dynamic promotion on an e-commerce site
  - User searches for “humidifier,” you want to show deals for humidifiers on the home page to that user



Sample message to ‘searches’ topic:

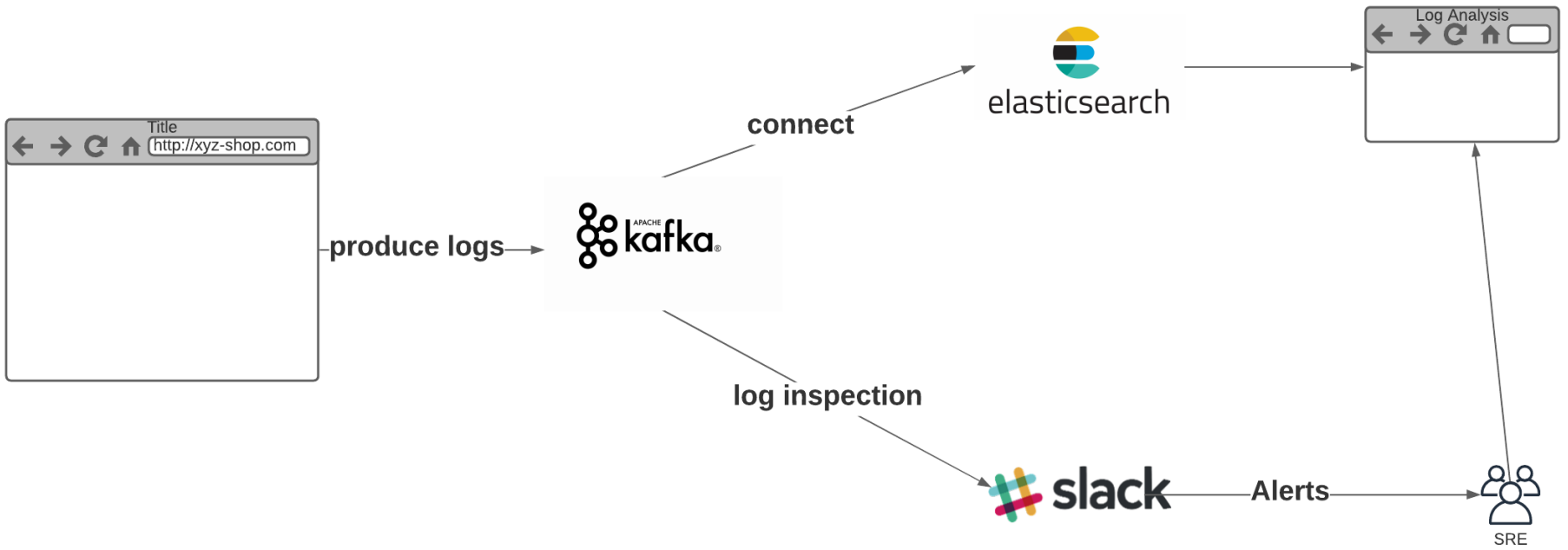

```
{
“user_id” : “[abcde@gmail.com](mailto:abcde@gmail.com)”,
“page_url” : “/deals/”
“search_keyword” : “humidifier”,
“date” : 1640912388090
}
```

Website activity tracking



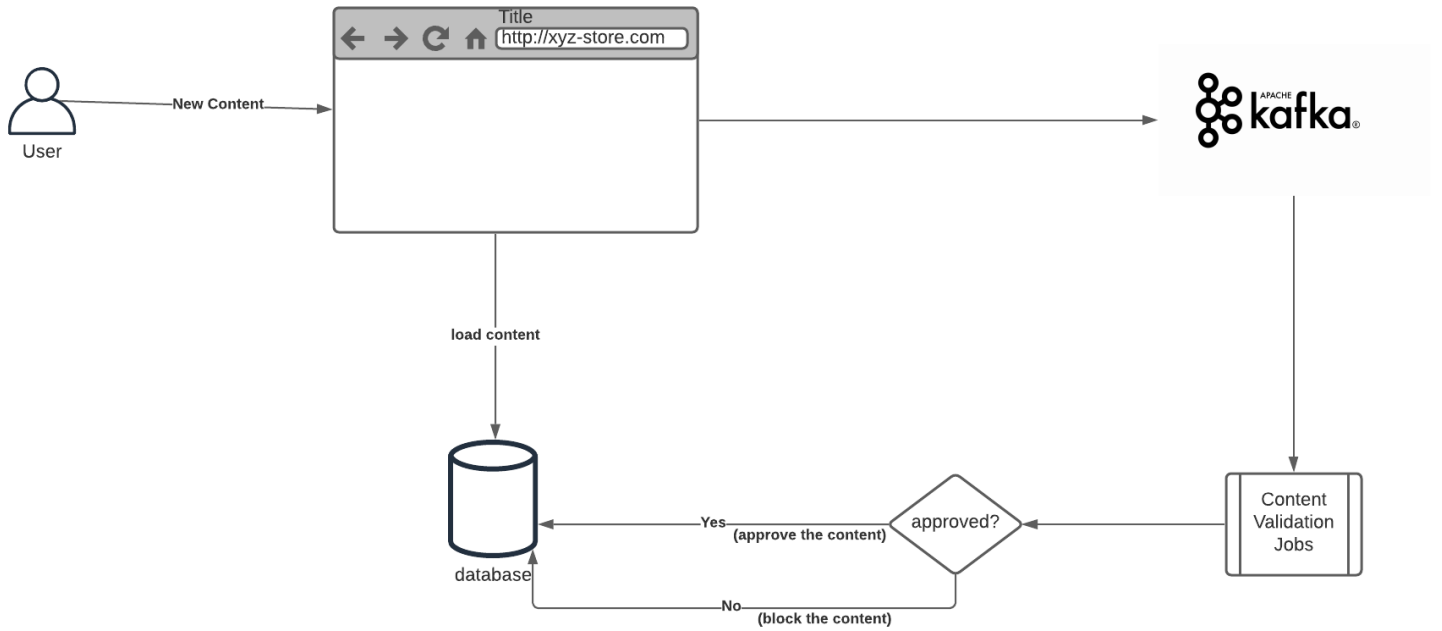
Log aggregation

- Central place for web app's logs
- Process them for alerts
- Move them to analysis tools for reporting
  - Or run processes to trigger alerts
- Storing logs in Kafka makes it easy to switch analysis tools later



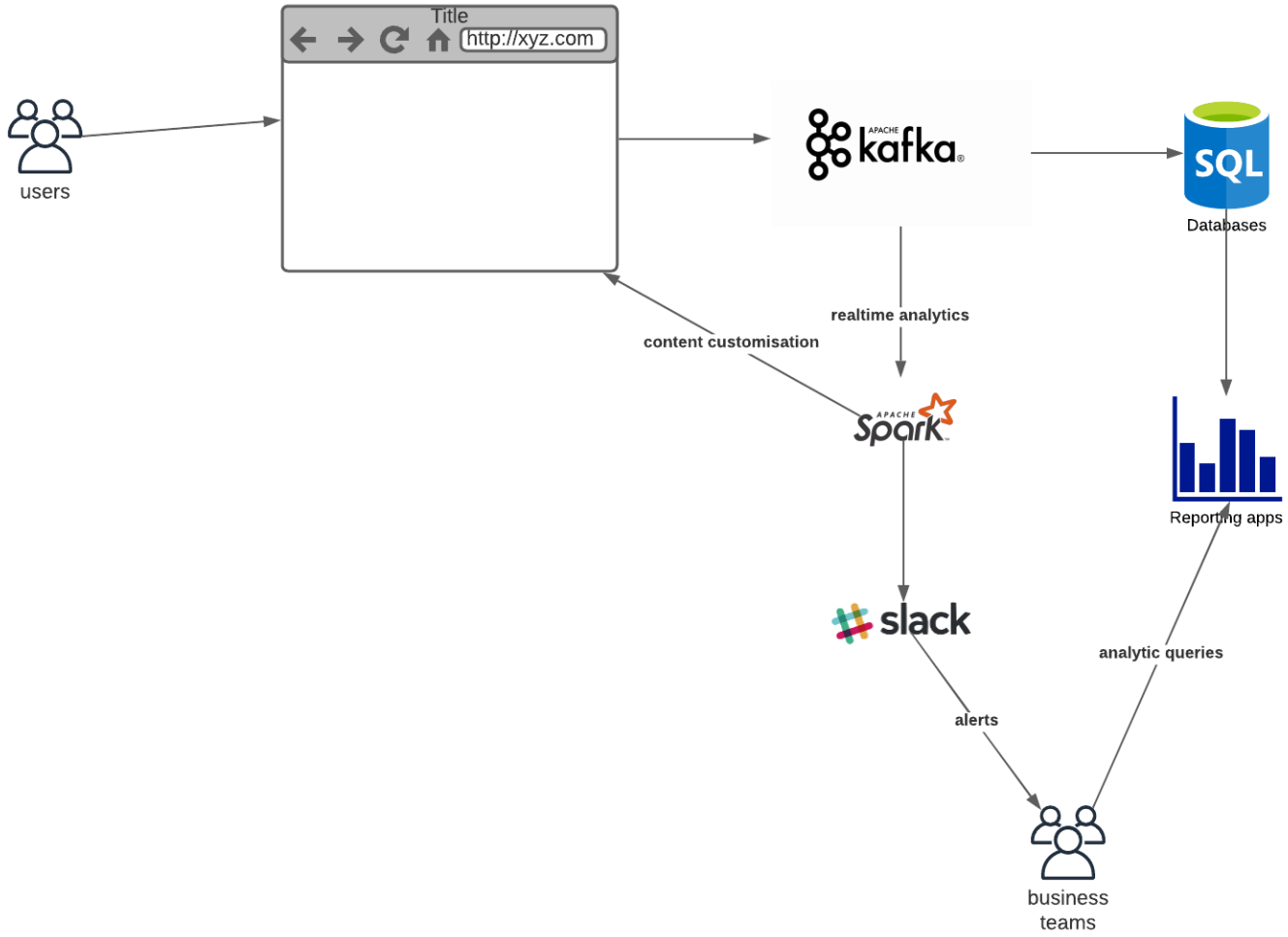
Content Moderation

- For example, product reviews
  - Need to axe fake reviews or reviews that violate policies
- Decouple moderation system from the web app
  - Makes it easier to switch moderation systems
  - But must be able to retroactively validate past content when switching
- Review start in “pending” state and visible only to author until approved



Website analytics

- Similar to activity tracking
- Behavior is stored to a database for analytical processing and reporting
- Kafka also allows real-time processing on the same data



Coming up: Hands-On Activities!

- Serverless URL shortener app
- Content moderation
- Message streaming
- Application log streaming
- Webhooks (pushing Stripe events into Kafka)

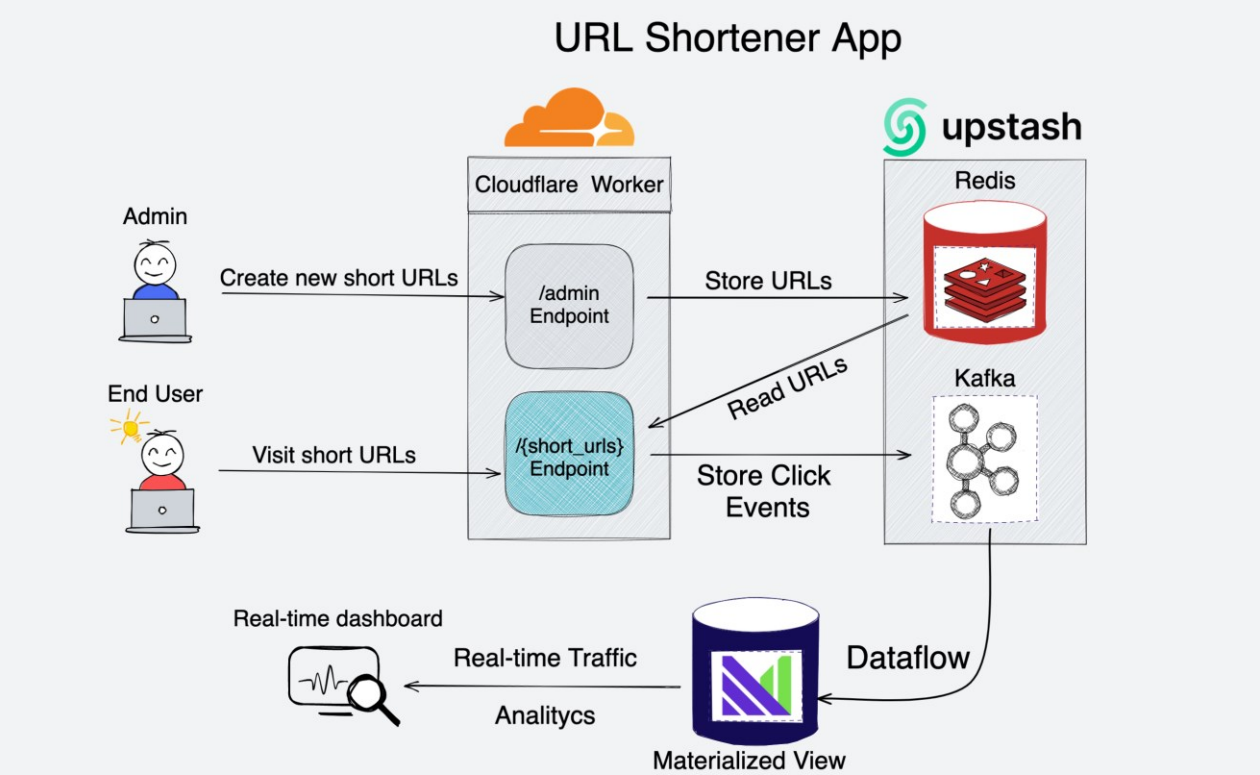
Serverless Kafka with Upstash

- In these activities, we'll create a Kafka cluster using a free account with Upstash
- Upstash offers serverless Kafka (and Redis)
  - WAY easier than setting up your own Kafka cluster from scratch
  - REST API, SDK for wide range of integrations
  - Generous free tier (10,000 messages/day, 256MB retention)
  - Pay-as-you-go for production usage (based on # of messages and storage) with a price cap
    - See docs.upstash.com/kafka/pricing
  - Truly serverless – price scales to 0
  - Enterprise plans available
    - Higher throughput, security
- Let's set up an account and a cluster

Building a Serverless URL Shortener App

With Upstash Kafka, Node.js, Cloudflare, and Materialize

App structure

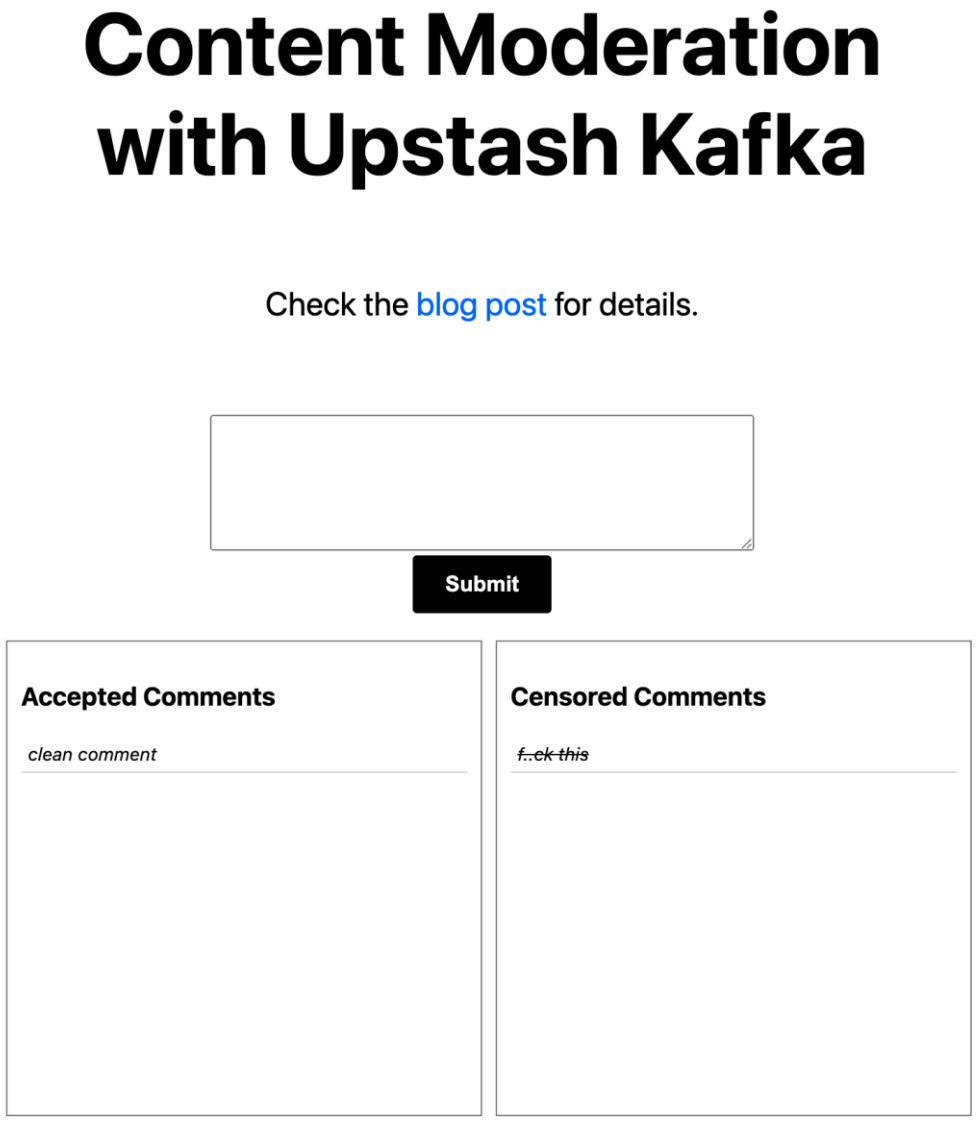


- Cloudflare worker accepts new short-links and redirects them
- Data stored in Upstash Redis (serverless)
- Redirects trigger events stored in Upstash Kafka
- Materialize receives data from Kafka for real-time analysis

Content Moderation

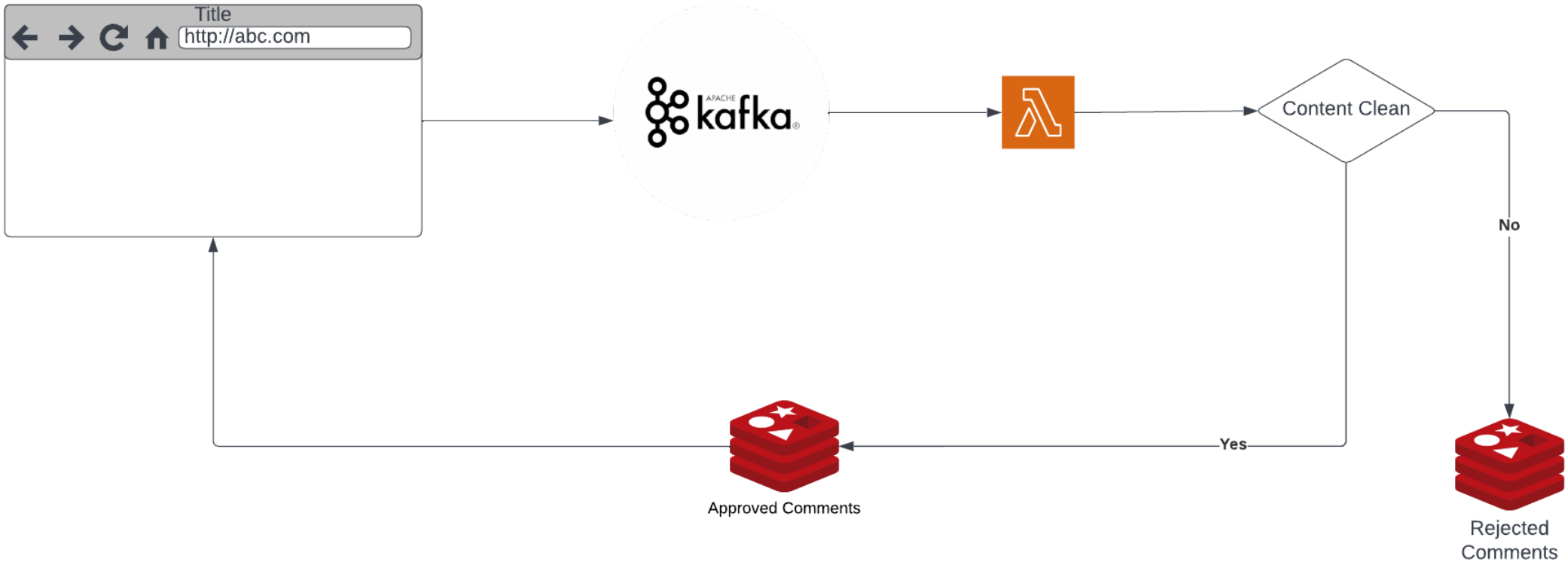
With Upstash Kafka & Redis, AWS Lambda, Next.js, and Sightengine

What we're building



How it works

- Comments sent to Kafka topic
- Kafka message triggers Lambda function
- Lambda calls Sightengine for moderation
- Sends message to Redis
  - “comments” list if appropriate
  - “rejected-comments” if not

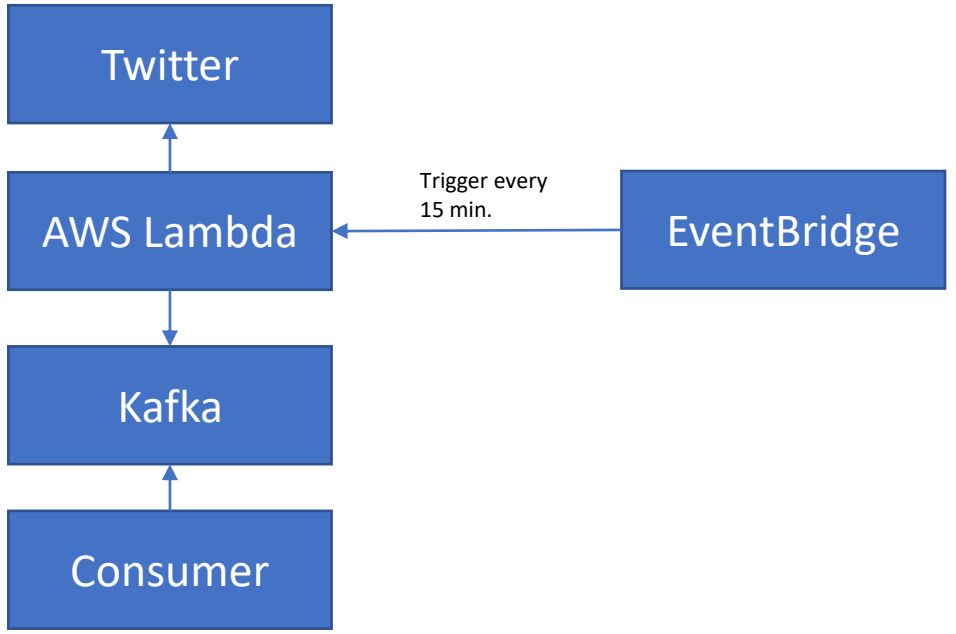


Message Streaming

With Twitter, Upstash Kafka, and AWS Lambda

How it works

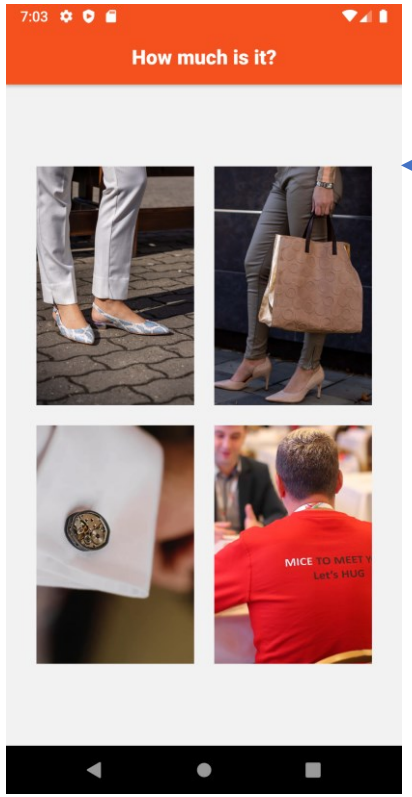
- AWS Lambda function runs every 15 minutes
- Tweets containing some keyword fed into Kafka producer
- Kafka consumer receives the tweets



Application Log Streaming

With Upstash Kafka & Redis, and React Native

How it works



Prices

Redis

Log info

Kafka

React Native app

Webhooks

With Stripe and Upstash Kafka's Webook API

Connecting Stripe and Kafka: Use cases

- Process payment events to notify business or sales
  - Flink, Spark, etc.
- Trigger sending Slack messages or email when a payment fails
- Use Kafka connector to move data into a database or data warehouse
  - Reporting & analytics
- Feed activity data to CRM system

How it works