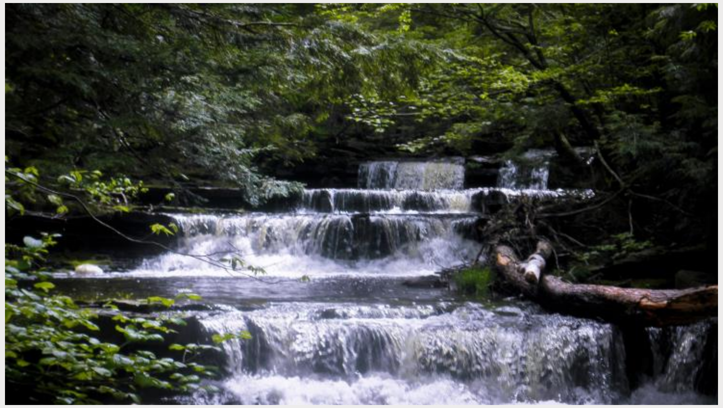
STREAMING WITH KAFKA

Publish/Subscribe Messaging with Kafka

What is streaming?



- So far we’ve really just talked about processing historical, existing big data
  - *Sitting on HDFS*
  - *Sitting in a database*
- But 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!*

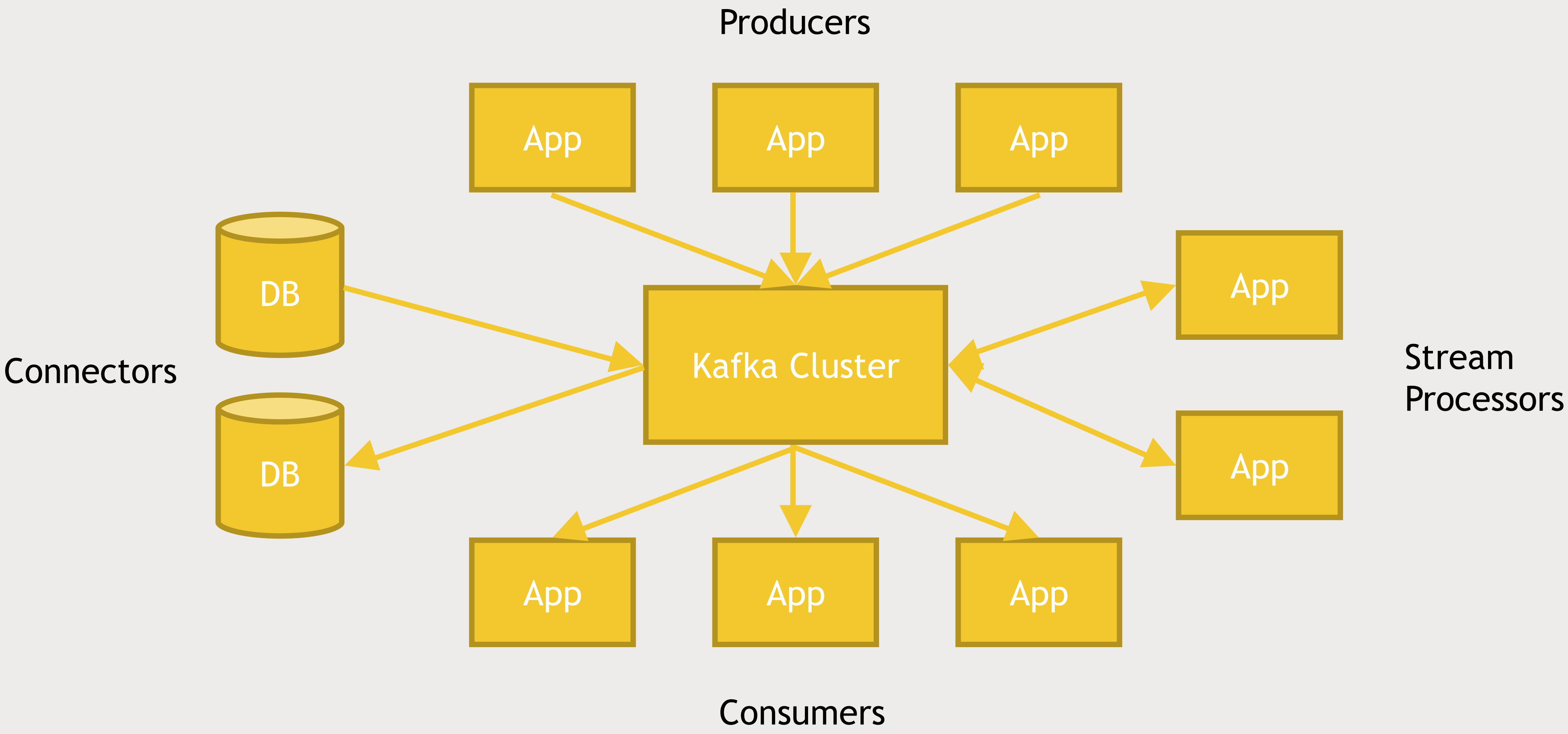
Two problems

- How to get data from many different sources flowing into your cluster
- Processing it when it gets there
- First, let’s focus on the first problem

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
- It's not just for Hadoop

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*

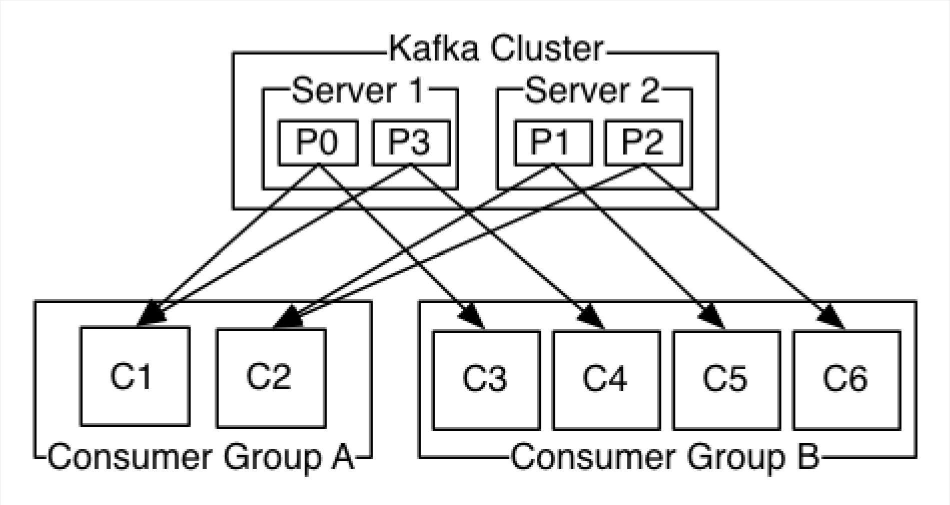


Image: kafka.apache.org

Let’s play

- Start Kafka on our sandbox
- Set up a topic
  - *Publish some data to it, and watch it get consumed*
- Set up a file connector
  - *Monitor a log file and publish additions to it*