StreamSets and Spark: Analytic Insights In Retail

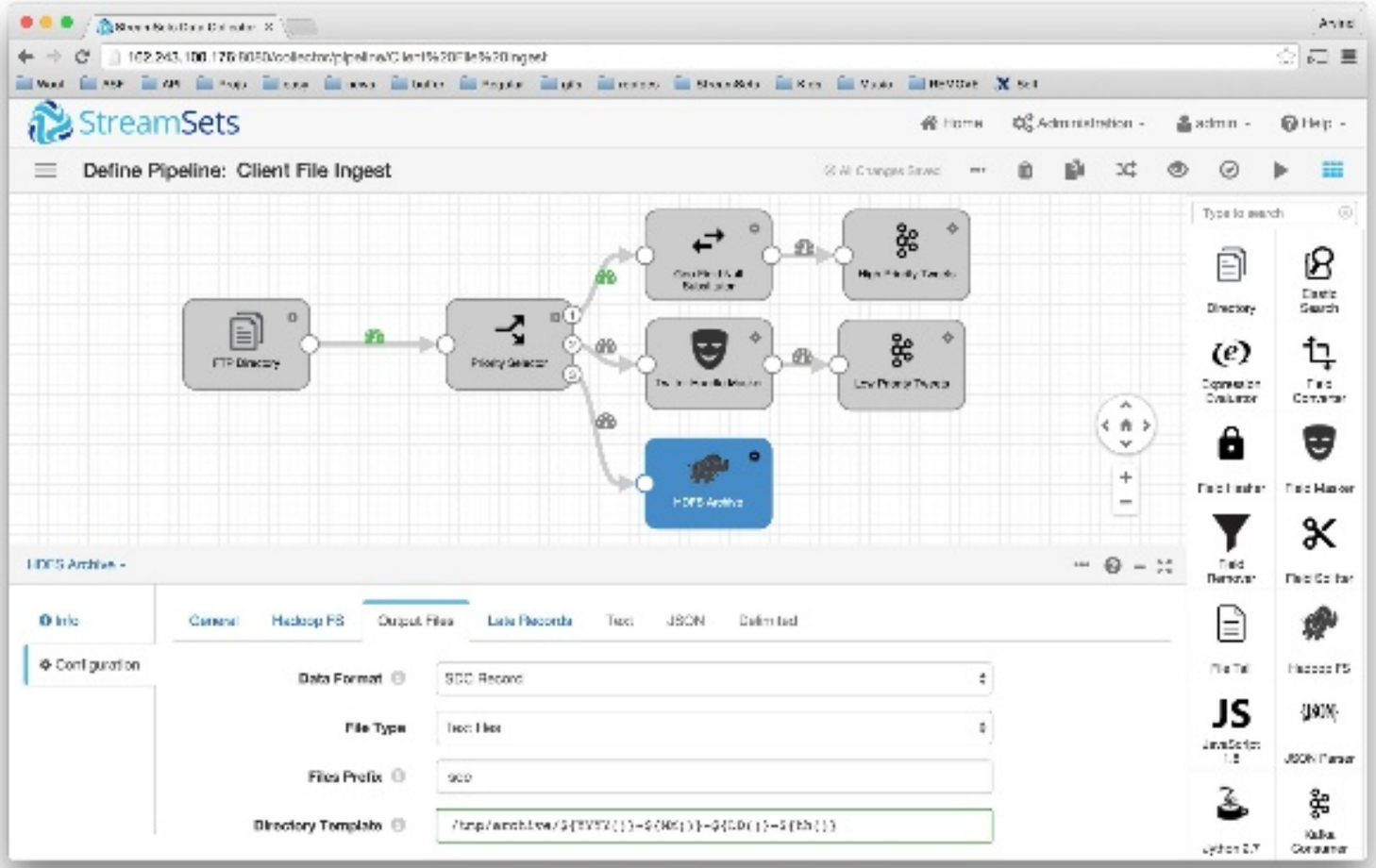
StreamSets

Hari Shreedharan
Software Engineer
@harisr1234
hshreedharan@streamsets.com

Who am I?

- Software Engineer at Streamsets
  - Since March 2016
  - Work on:
    - Streamsets Data Collectector
    - Dataflow Performance Manager
- Software Engineer at Cloudera, Feb 2012 to March 2016
  - Worked on Apache Flume, PMC Chair
  - Worked on Apache Sqoop, Committer
  - Worked on Apache Spark, Contributor
- Software Engineer at Yahoo!, Jan 2010 to Feb 2012
  - Worked on Yahoo! Mail Metadata storage
- Author of O'Reilly's "*Using Flume*"

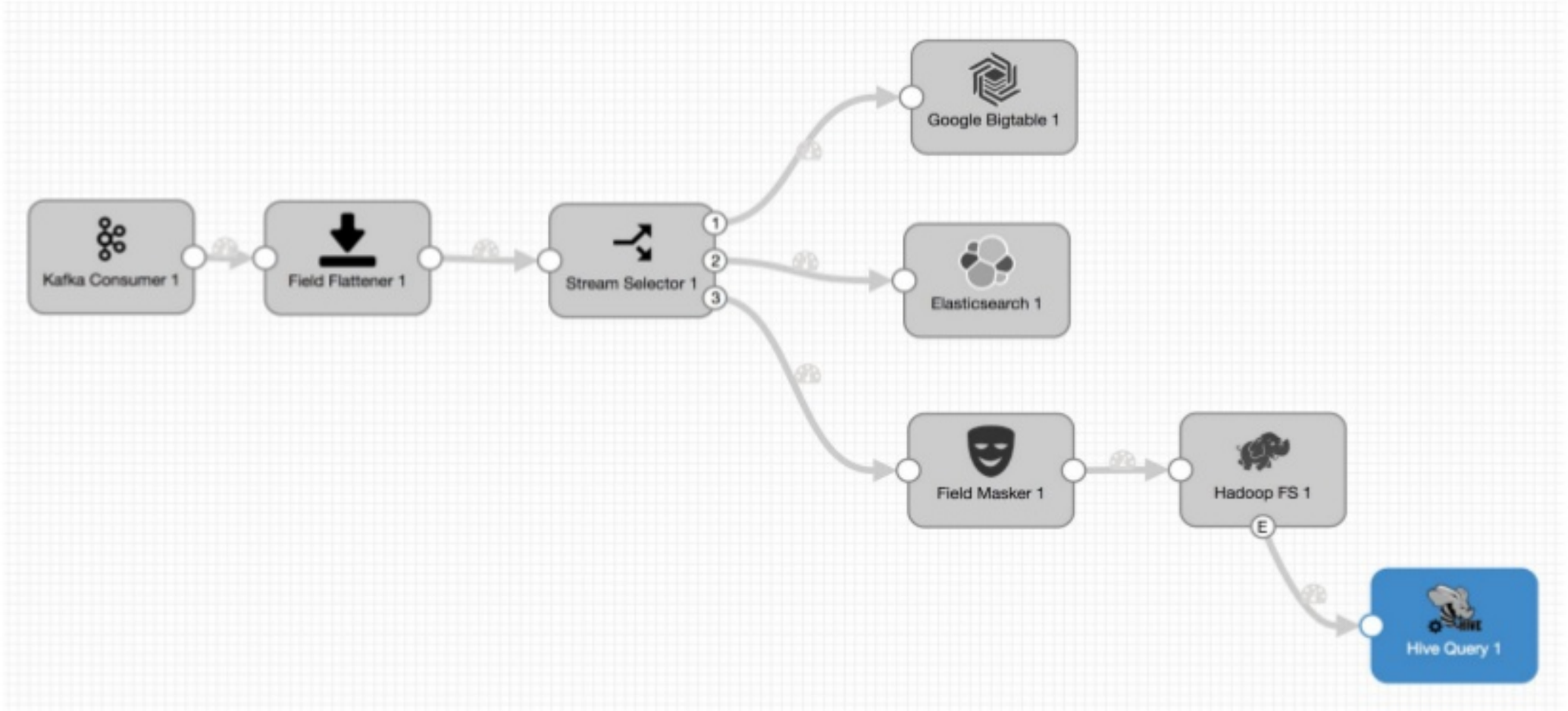
StreamSets Data Collector



Open source software for the rapid development and reliable operation of complex data flows.

- Efficiency
- Control
- Agility
- In-memory batches

Inside an SDC



Sensors and Data for Personalized Shopping

- Retail giant - hundreds of shoppers in each center at any time
- Lot of data available about each shopper:
  - Location
  - Existing user-profiles
  - Phone app data
- How to use this data to provide a personalized shopping experience for each of these shoppers?
- Sensors around the center track:
  - WiFi
  - Bluetooth
  - In some cases, cell phone usage

Sensors contd..

StreamSets

- Partial WiFi handshake
  - Get unique id for each device
- Multiple sensors can be used to track an approximately accurate location
- Foot traffic pattern calculated using this information
- Unique user's approximate time spent in individual stores and parts of the center can be calculated and improved over recurring visits
- App provides additional information
- Specific promotions and updates can be sent to their devices (via app) on recurring visits

Getting the data from sensors

StreamSets

- Data pulled in from sensors via Streamsets Data Collector
- Various IoT origins:
  - CoAP
  - HTTP
  - TCP
  - UDP
  - WebSockets
  - MQTT
  - OPC-UA
- Add more nodes as required on (multi)cloud or on-prem or both (hybrid)

Formatting the data

StreamSets

- SDC transparently converts several formats into SDC Record:
  - JSON
  - Avro
  - Byte[]
  - Delimited
  - Protobuf
  - Text
  - XML
  - Apache Log format
- Wow! That's a lot of formats!
- Process SDC records and then write it out in any format, not necessarily the one you got the data in!
- We also support ingesting Binary data

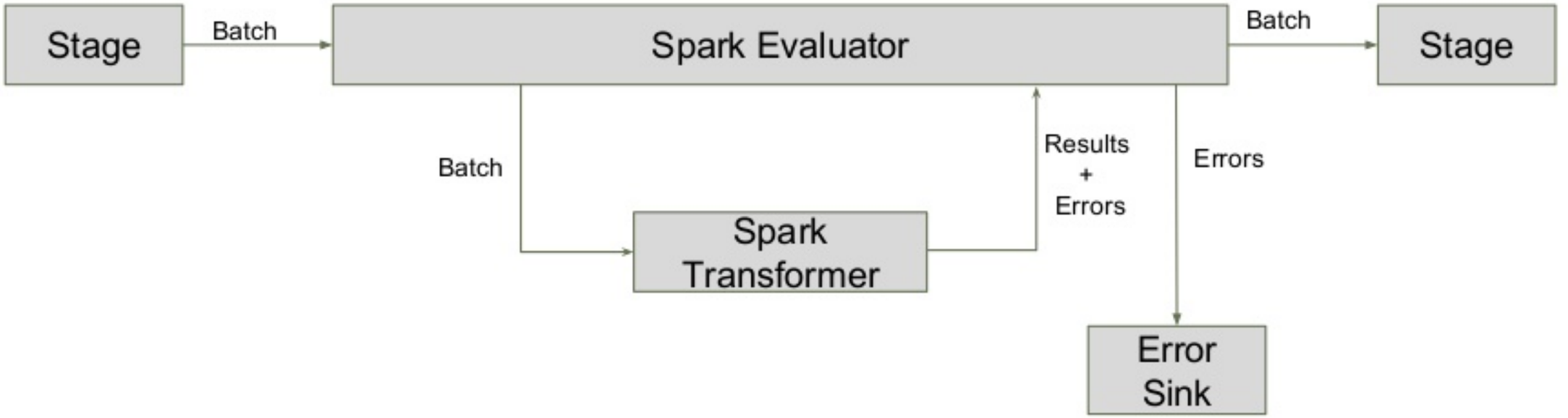
Processing the data

StreamSets

- In-stream processing using processors
- Most processors are `map` operations on individual records
- Scripting processors process batches of records in python, groovy, javascript
- Spark Evaluators:
  - Perform arbitrary Spark operations on each batch of records
  - Each batch is an RDD[Record]
  - Stash batches or maintain state!
  - Perform reduces/shuffles.
  - Cluster mode pipelines
    - Each pipeline in the cluster generates an RDD batch
    - RDD can see all data coming in through the cluster

Spark Evaluator

- Long running SparkContext, passed to user-code during pipeline start
- Processor that runs each batch through user provided "application" - *SparkTransformer*
- Each batch of records passed in as an RDD to the transformer
- Use MLLib, existing Spark-based algorithms



Spark Evaluator

- Transformer returns:
  - Result records that need to go to the next stage
  - Error records that can't be processed
    - Written out to error stream
- Results are passed through to the rest of the pipeline
- Already available for CDH, MapR

Using Spark Evaluator

StreamSets

- In-stream processing useful for various real-time processing
  - Anomaly Detection
  - Sentiment Analysis
  - NLP
- Uses for retail giant:
  - Fraud Detection
  - Real time feedback/promotions

SDC on Spark - Connectivity

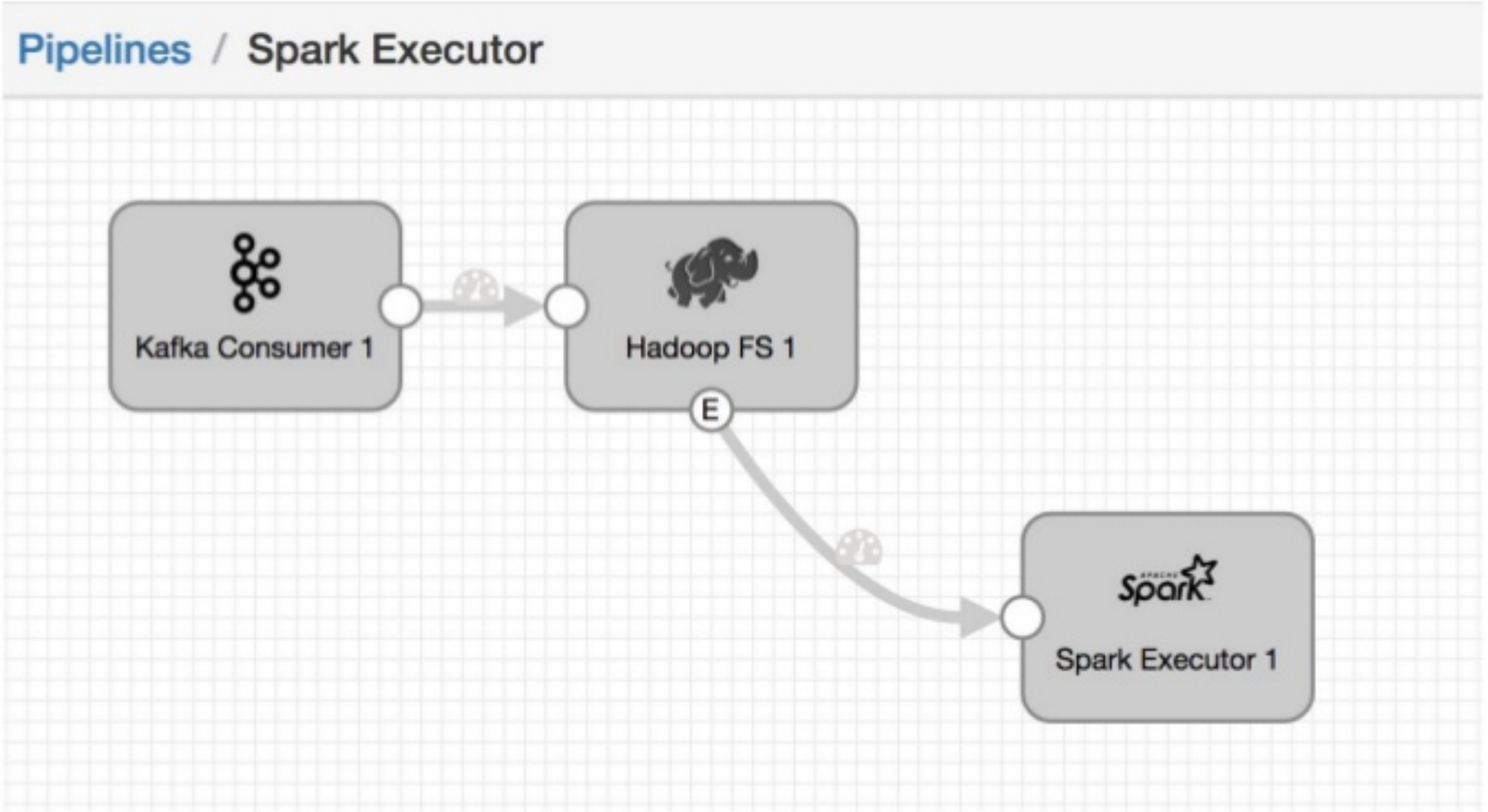
Sources

- Kafka

Destinations

- HDFS
- HBase
- S3
- Kudu
- MapR DB
- Cassandra
- ElasticSearch
- Kafka
- MapR Streams
- Kinesis
- etc, etc, etc!

Spark Executor



Spark Executor

- When an event is received, kick off a Spark application
- Ability to provide an application jar, and specific configuration
- Supports YARN and Databricks cloud support.
- YARN
  - Client and Cluster mode
  - Parameters can be based on the event data like file name
- Databricks Cloud
  - Define job beforehand
  - Jar or Notebook Job
  - Kick off the job on event
  - Parameters can be based on the event data like file name

databricks

cloudera

MAPR

Using Spark Executor

StreamSets

- Simplify operationalization of jobs on Databricks
- On File close on S3:
  - Kick of Databricks job
  - Object name passed in as a parameter
  - File data read in the Spark job
  - Can be a notebook job
    - `Dbutils.widgets` can be set by the executor to pass in parameters to notebooks
- Notebook integration can be used for easy experimentation:
  - Trial and error
  - Improvement of current job in real time
  - Change job between file closes
- Useful for model creation, classification algorithms and other batch jobs

Conclusion

StreamSets

- Spark is excellent for in-stream and end of batch processing
- SDC makes in-stream processing with Spark easy
- SDC + Databricks cloud makes ingest followed by batch processing a breeze!
- SDC + Notebooks makes it easy to incrementally improve applications and experiment with parameters.

Questions?