Build a Real-time Stream Processing Pipeline with Apache Flink on AWS

Dr. Steffen Hausmann, Solutions Architect, AWS

September 13, 2017

Stream Processing Challenges

Consistency and high availability

Low latency and high throughput

Rich forms of queries

Event time and out of order events

Apache Flink

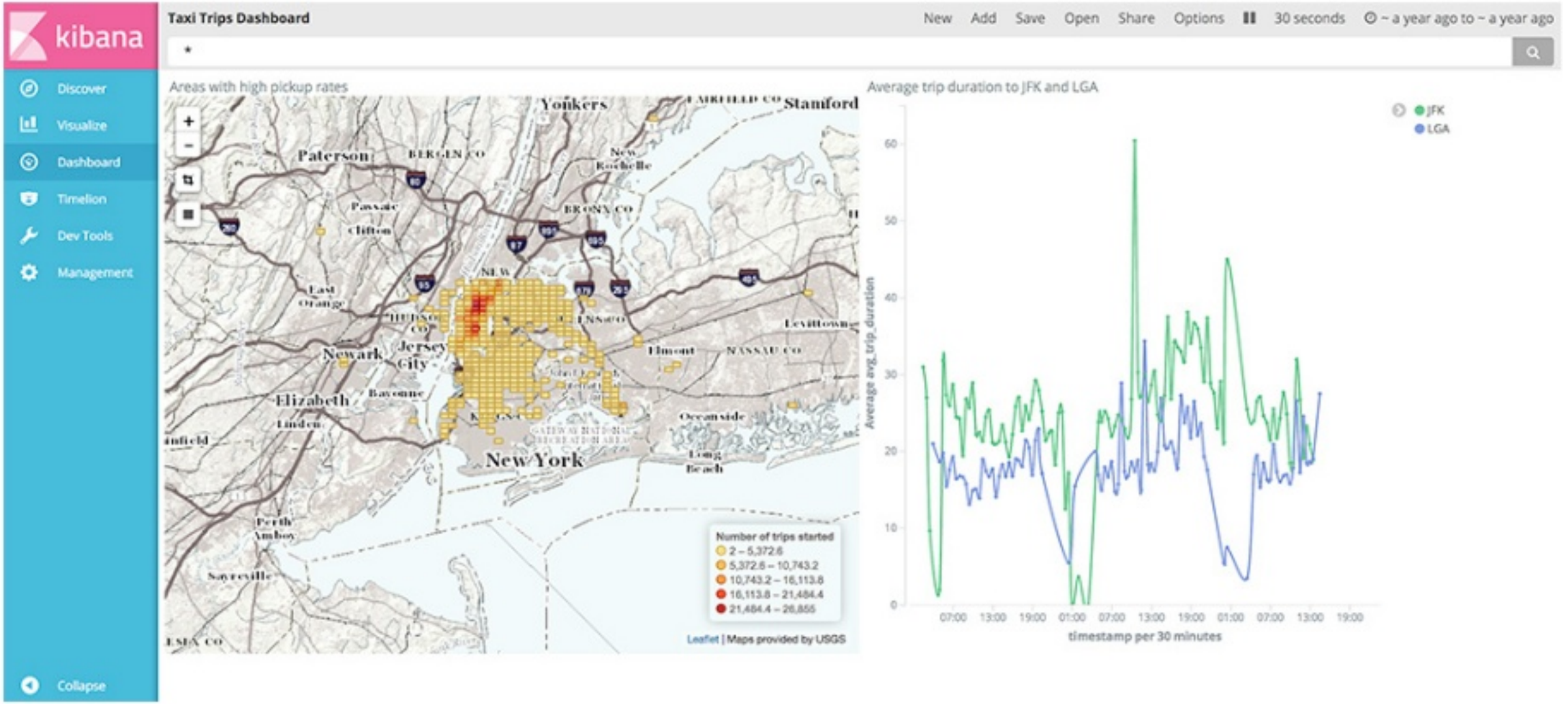
"**Apache Flink®** is an open source platform for distributed stream and batch data processing."

https://flink.apache.org/

http://data-artisans.com/why-apache-flink/

Flink

Analyzing NYC Taxi Rides in Real-time



Simple Pattern for Streaming Data

Data Producer

Continuously creates data

Continuously writes data to a stream

Can be almost anything

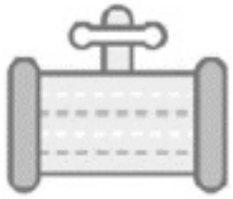
Mobile Client

Streaming Storage

Durably stores data

Provides temporary buffer

Supports very high-throughput



Amazon Kinesis

Data Consumer

Continuously processes data

Cleans, prepares, & aggregates

Transforms data to information

Apache Flink

Amazon Kinesis Streams

Create streams to capture and store streaming data

Replicates your streaming data across three facilities

Elastically add and remove shards to scale throughput

Secured via AWS IAM and server-side encryption

Amazon Elastic Map Reduce (EMR)

Easily provision and manage clusters for your big data needs

Hadoop, Flink, Spark, Presto, HBase, Tez, Hive, Pig, …

Dynamically scalable, persistent, or transient clusters

Tightly integrated with other AWS services, eg, for storage, encryption, and monitoring

Amazon Elasticsearch Service

Setup Elasticsearch cluster in minutes

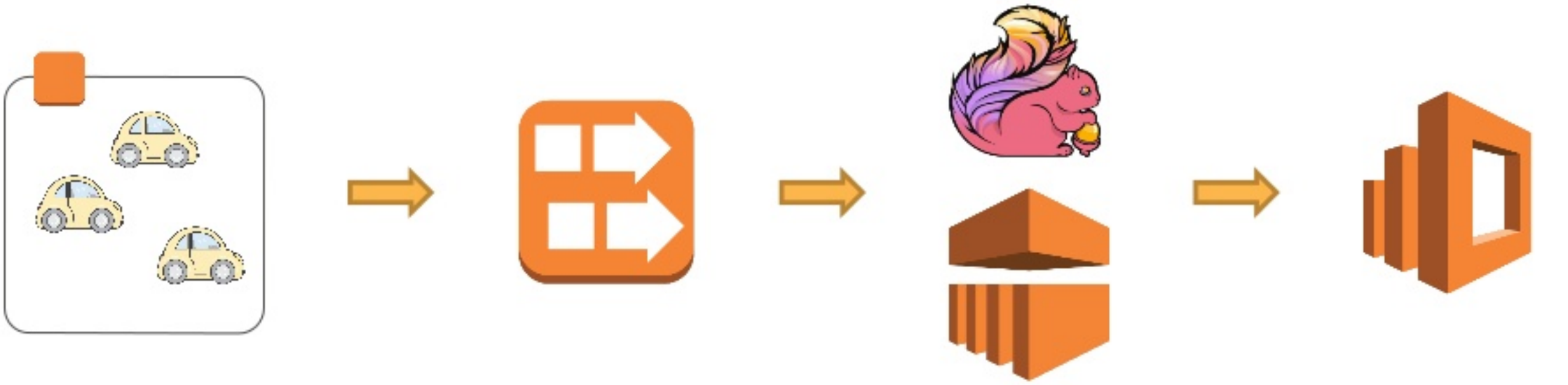
Integrated with Logstash and Kibana

Scale Elasticsearch clusters seamlessly

Highly available and reliable

Tightly integrated with other AWS services

Architecture for Analyzing Taxi Rides



Amazon Kinesis Streams

Apache Flink on Amazon EMR

Amazon ES

Let’s dive right in!

Lessons Learned and Best Practices

Building the Flink Kinesis Connector

The Flink Kinesis Connector binary is not available from Maven Central

Build the Connector with Maven 3.0.x, 3.1.x, or 3.2.x …

- mvn clean install -Pinclude-kinesis -DskipTests -Dhadoop-two.version=2.7.3

… or use CodeBuild to let it be build for you!

Important Parameters of the Kinesis Connector

AWS_CREDENTIALS_PROVIDER

- determines how Flink obtains IAM credentials
- set to AUTO and use appropriate roles with the EMR cluster

SHARD_GETRECORDS_INTERVAL_MILLIS

- determines how often Flink polls events from Kinesis
- set to at least 1000 to facilitate multiple consumers

Connecting to the Flink Dashboard

Use dynamic port forwarding to the Master node

- ssh -D 8157 hadoop@...

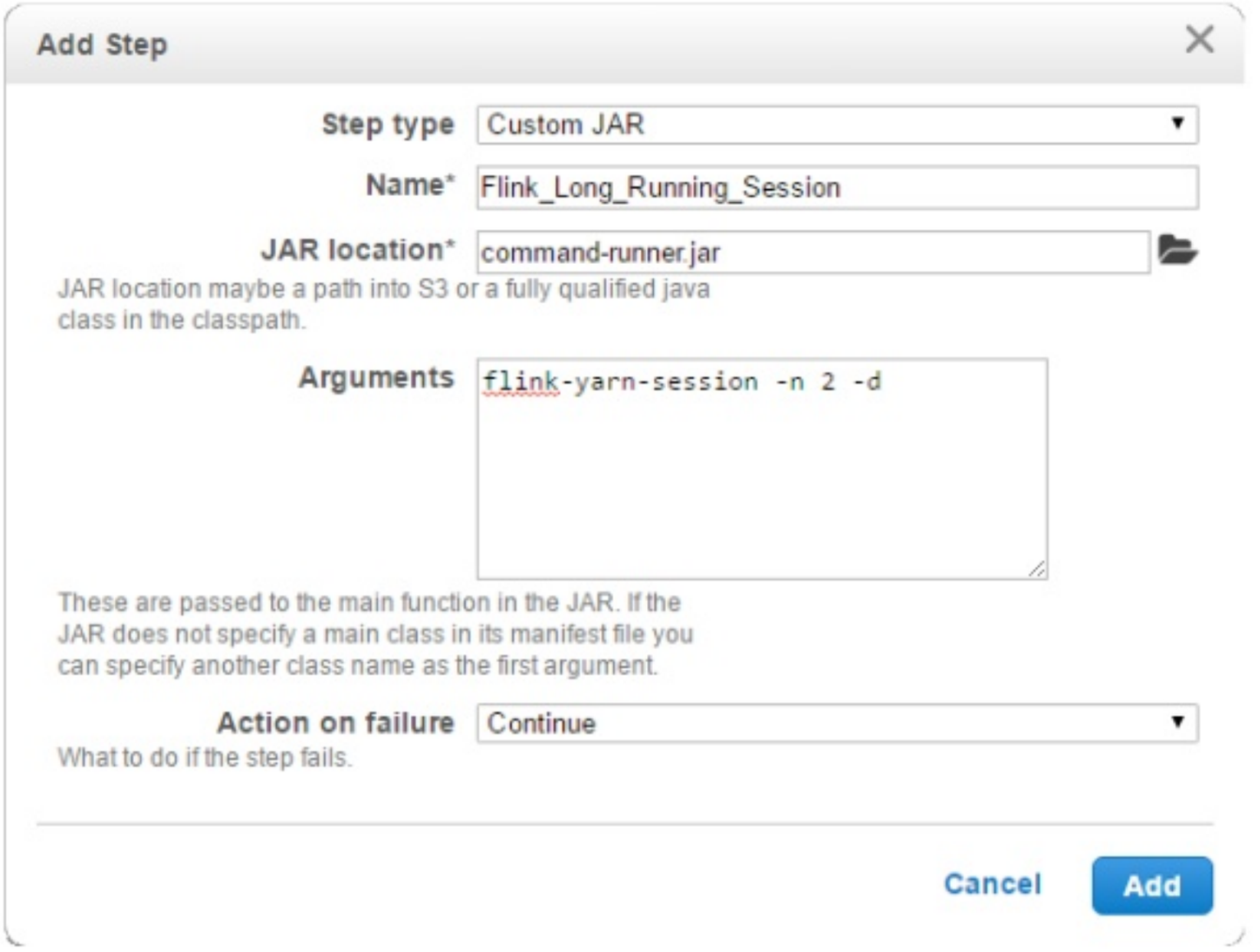
Use FoxyProxy to redirect URLs to localhost

- *ec2*.amazonaws.com*
- *.compute.internal*

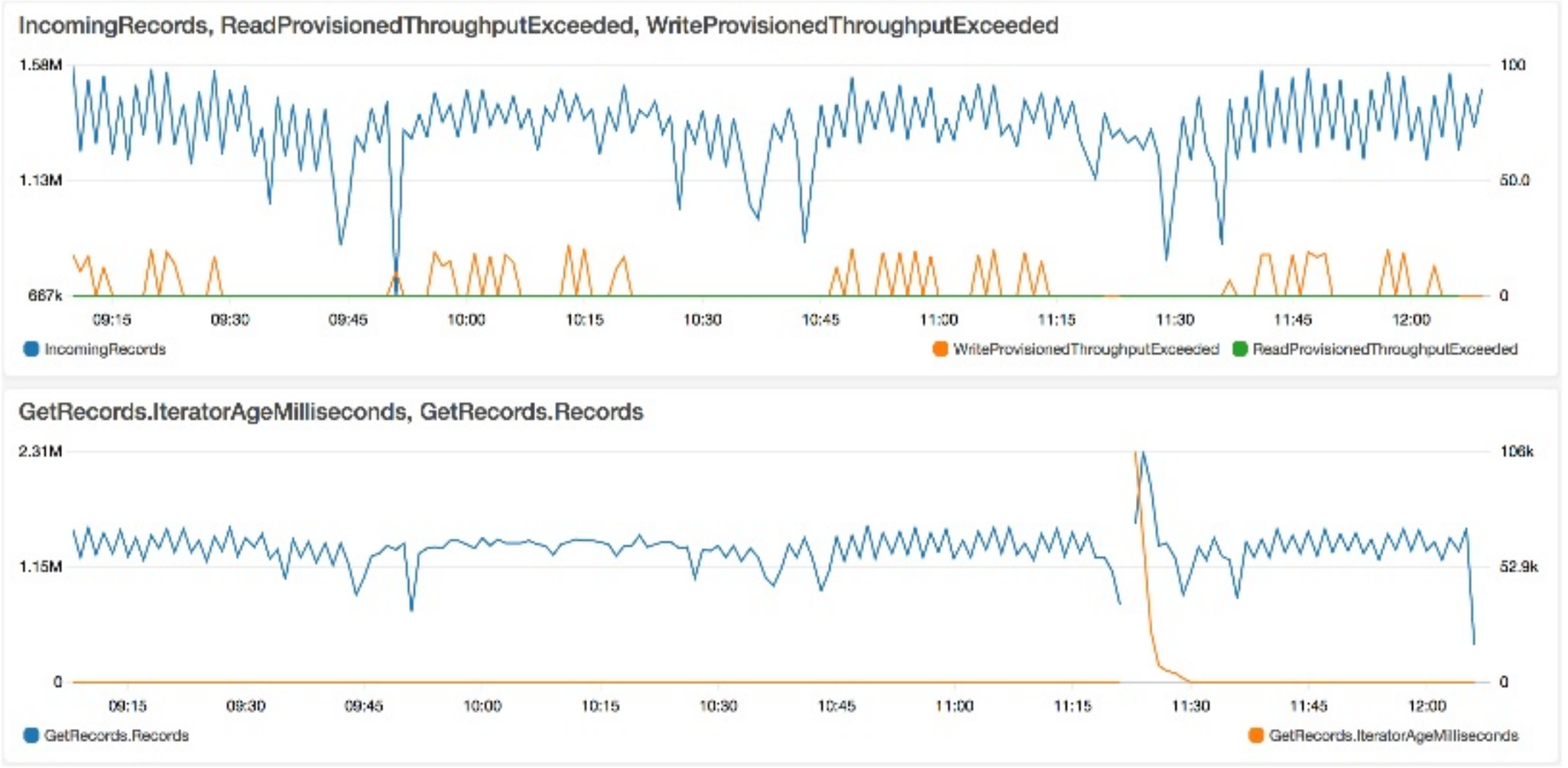
Connect through the EMR console

- navigate to the YARN Resource Manager
- select the Flink ApplicationMaster

Starting Flink and Submitting Jobs



Important Kinesis Streams Metrics

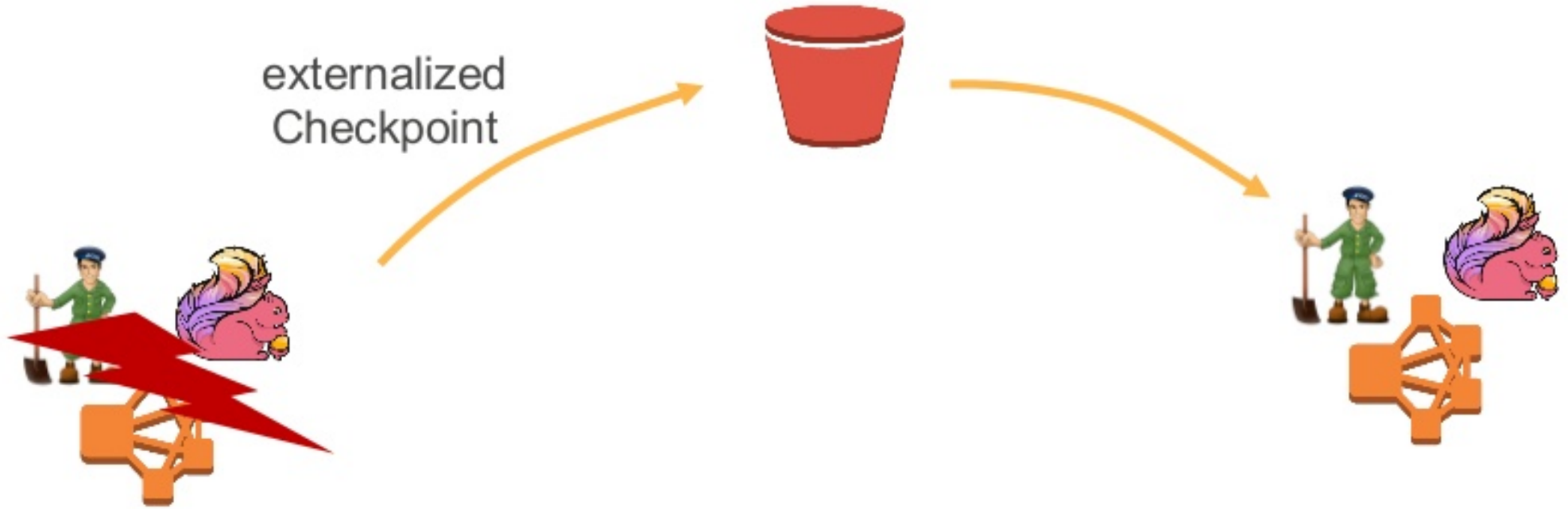


Checkpointing and High Availability

Zookeeper can be bootstrapped on EMR

Overprovision the EMR cluster for fast failovers

Use externalized checkpoints and store them on Amazon S3



Build a Stream Processing Pipeline Yourself

Many examples with sample code are on the AWS Big Data Blog. Follow the blog!

Build a Real-time Stream Processing Pipeline with Apache Flink on AWS

https://github.com/awslabs/flink-stream-processing-refarch/

Thank you!

shausma@amazon.de