
AWS
re:Invent
NOV. 28 – DEC. 2, 2022 | LAS VEGAS, NV

ANT339

How Samsung modernized architecture for real-time analytics

Jewon Lee

Head of SmartThings Cloud Engineering,
Samsung Electronics

Anusha Dasarakothapalli

Principal Software Engineer,
Amazon MSK, AWS

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Agenda

Samsung Electronics SmartThings

Before: problems faced with batch processing

Evolution

Now: modern stream processing platform

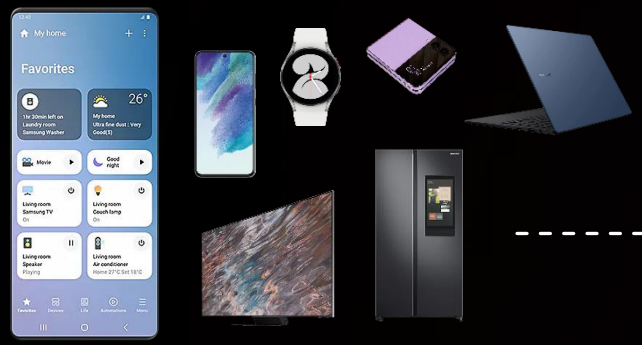
Further improvements: CI/CD and auto scaling

Impact & looking forward

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Samsung Electronics SmartThings

App



Cloud

SmartThings Cloud

Value-added services

- Video home monitoring
- Energy management
- Appliance services
- B2B customized

Hub



Appliances



Third-party cloud



aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

SmartThings data platform



Device history

History of device action and automation execution



Notification history

History of notifications delivered to users from the device



Usage statistics

Provides statistics such as device energy usage and usage time



Analytics & recommendation

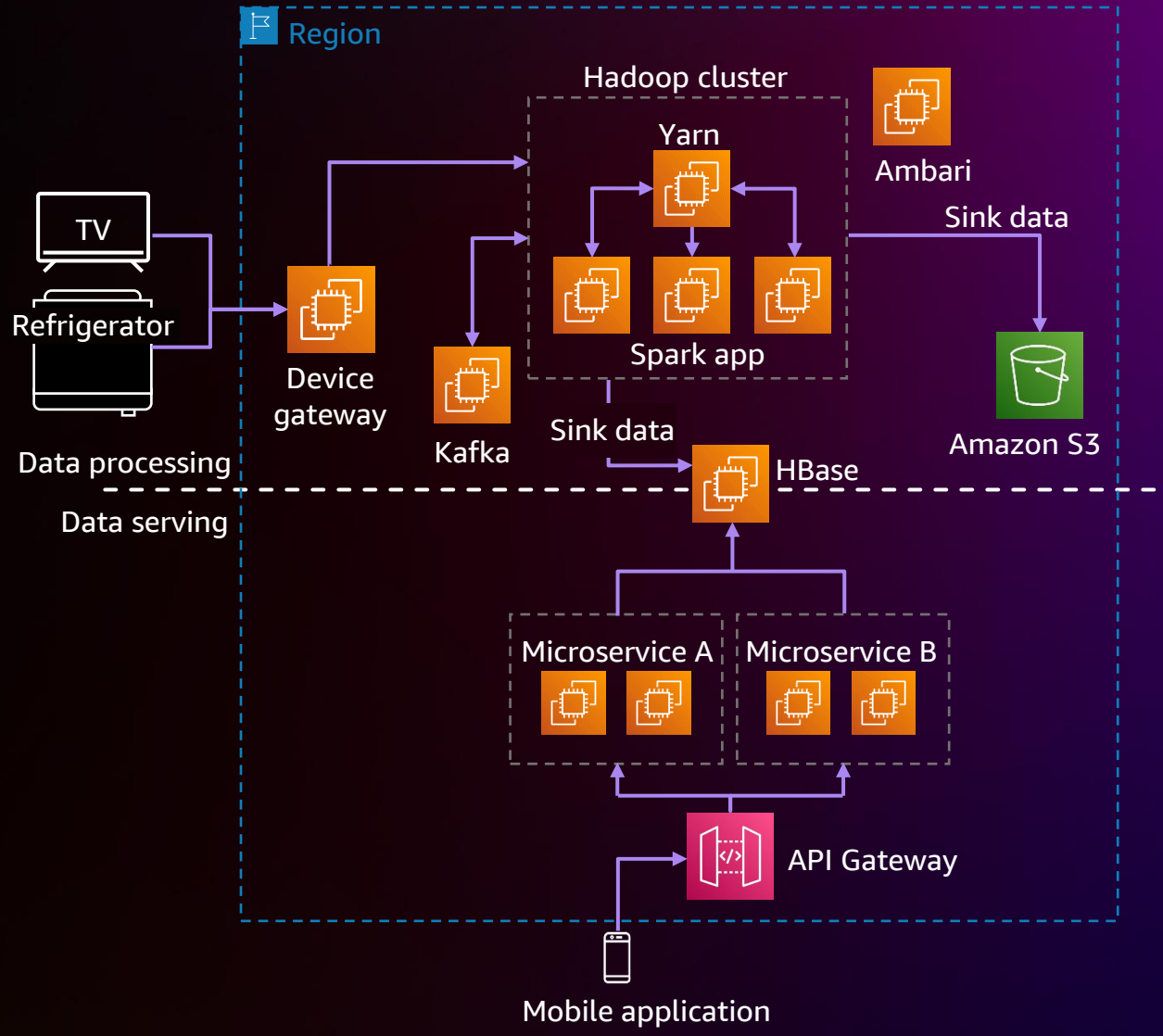
Analyze device usage patterns and recommend useful functions

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Before: batch data processing

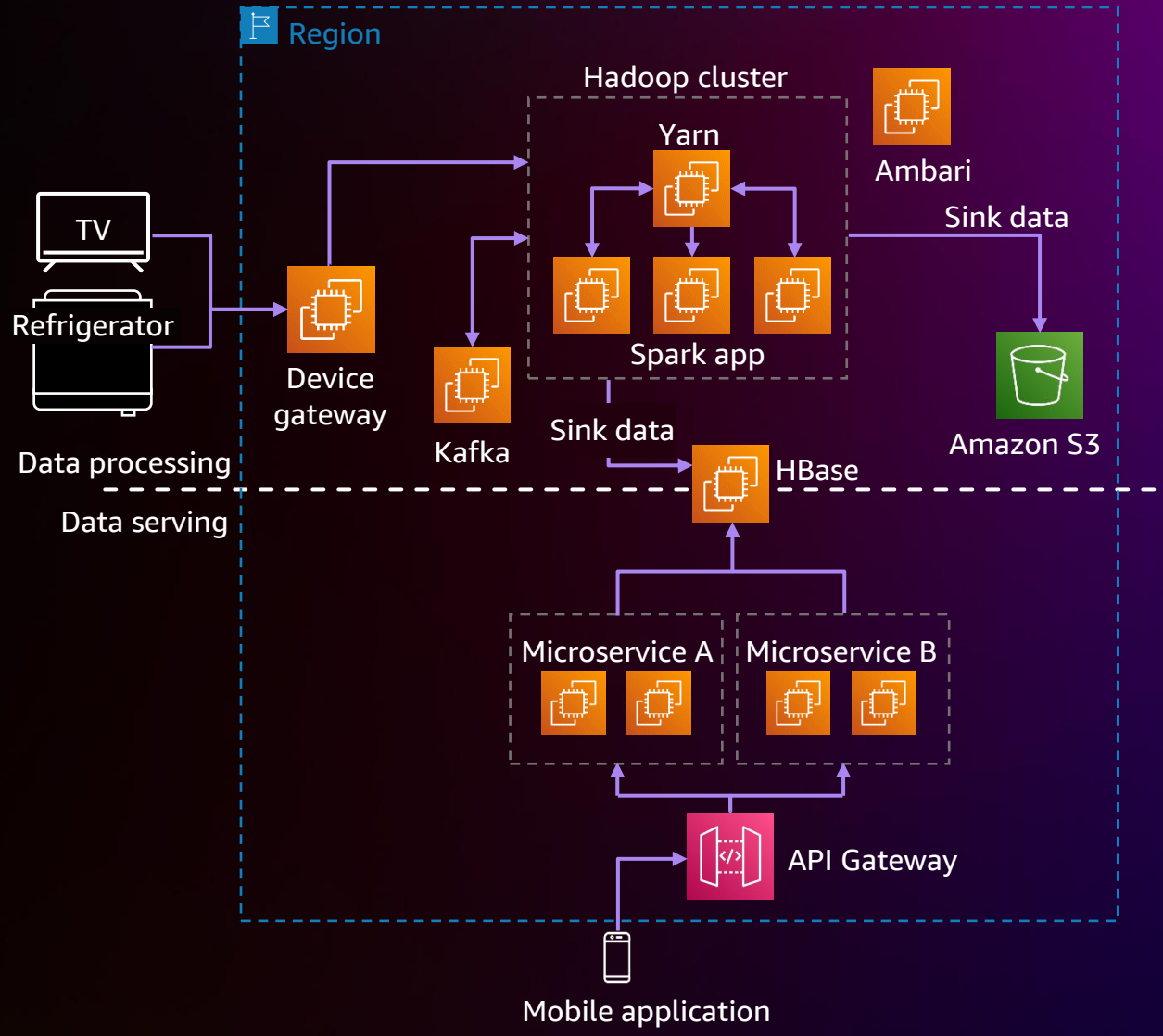
- Processes 30 TB+ a day
- Hadoop based clusters on Amazon EC2
- Several microservices on Amazon EC2



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

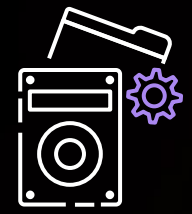
Before: batch data processing

- Processes 30 TB+ a day
- Hadoop based clusters on Amazon EC2
- Several microservices on Amazon EC2

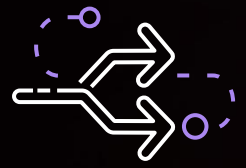


© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Before: Batch data processing – Challenges



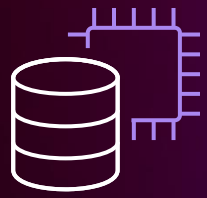
Resource contention



Low flexibility



Weak automatic recovery



CPU usage maxed out & lack of memory





Delayed data processing



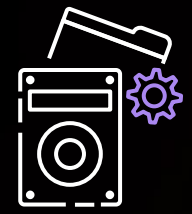
High operational overhead



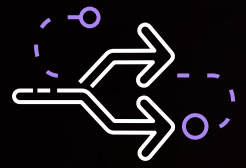
Inaccurate device operation history & energy consumption

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Before: Batch data processing – Challenges



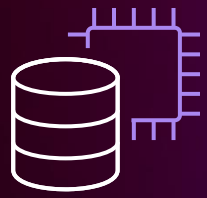
Resource contention



Low flexibility



Weak automatic recovery



CPU usage maxed out & lack of memory





Delayed data processing



High operational overhead



Inaccurate device operation history & energy consumption

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Operational pain with self-managed Spark

Number of failure recovery processes when operating 50 Spark applications

131

Data processing delay notification

28

App failure recovery

13

Manual cluster recovery

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

We knew we had to evolve the architecture

Needed



Reduce high operation cost



Improve low system stability



Enhance inefficient operating method

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

And we considered several options

Need

- Real-time large scale data analysis
- Cluster availability and maintainability
- Node and application scalability

Reviewed several options: Apache Spark vs. Apache Flink

- Spark on Amazon EKS
- Amazon Kinesis Data Analytics for Apache Flink

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

We chose Kinesis Data Analytics for Apache Flink

Advantages of using Apache Flink on Kinesis Data Analytics

- Serverless
- Managed job (i.e., applications are independent and no noisy neighbor issues)
- High availability

Concerns

- Migration cost from Spark to Flink

Apache Spark	Apache Flink
• Micro batch • Doesn't support recovery policy • Serverless with Amazon EKS	• Real time • State management with save points • Automatic recovery • Serverless with Kinesis Data Analytics

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

We chose Kinesis Data Analytics for Apache Flink

Advantages of using Apache Flink on Kinesis Data Analytics

- Serverless
- Managed job (i.e., applications are independent and no noisy neighbor issues)
- High availability

Concerns

- Migration cost from Spark to Flink

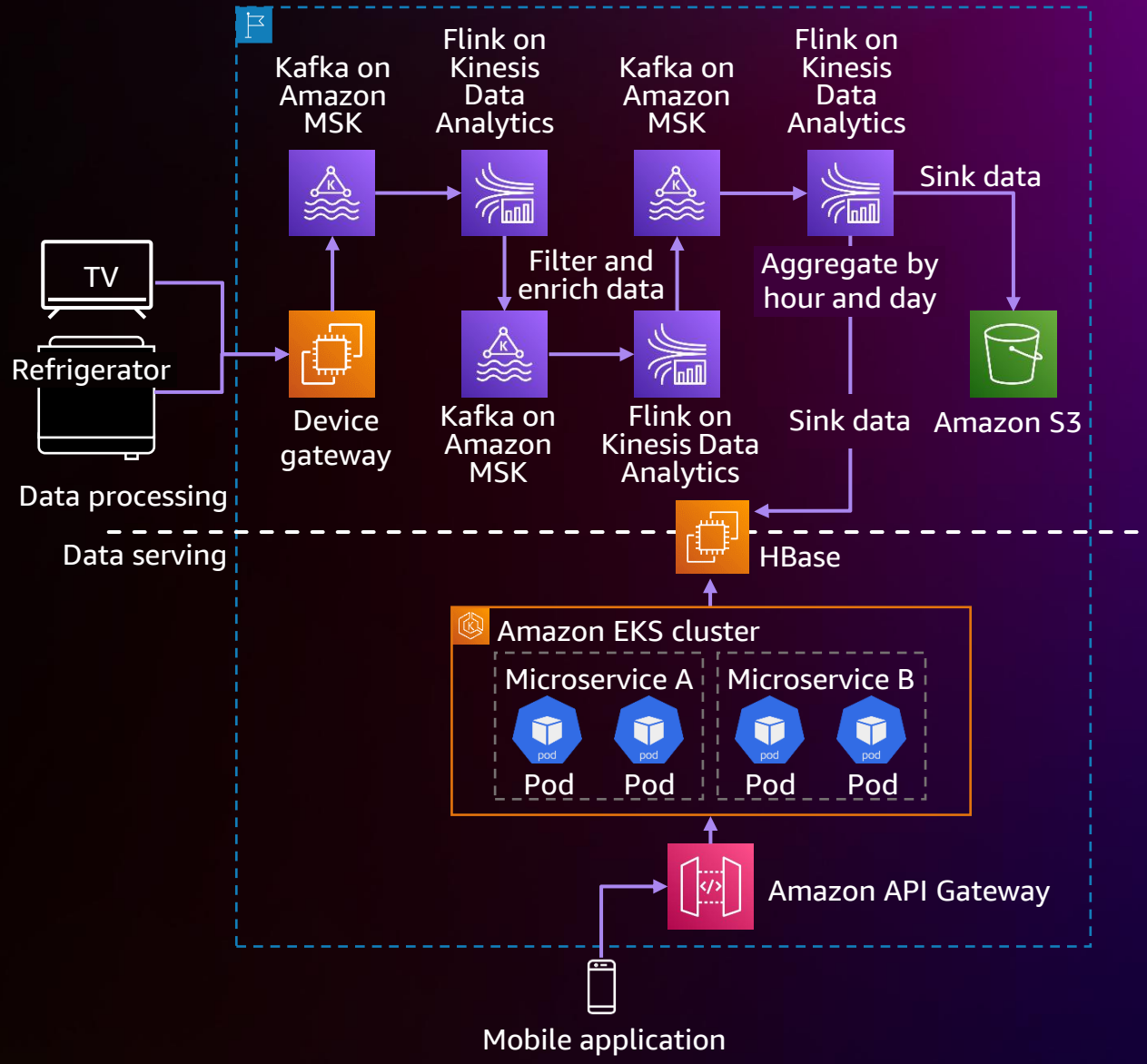
Apache Spark	Apache Flink
• Micro batch • Doesn't support recovery policy • Serverless with Amazon EKS	• Real time • State management with save points • Automatic recovery • Serverless with Kinesis Data Analytics

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Current architecture

- 30+ Flink apps on Kinesis Data Analytics
- Process data with managed Kafka
- Several microservices on Amazon EKS

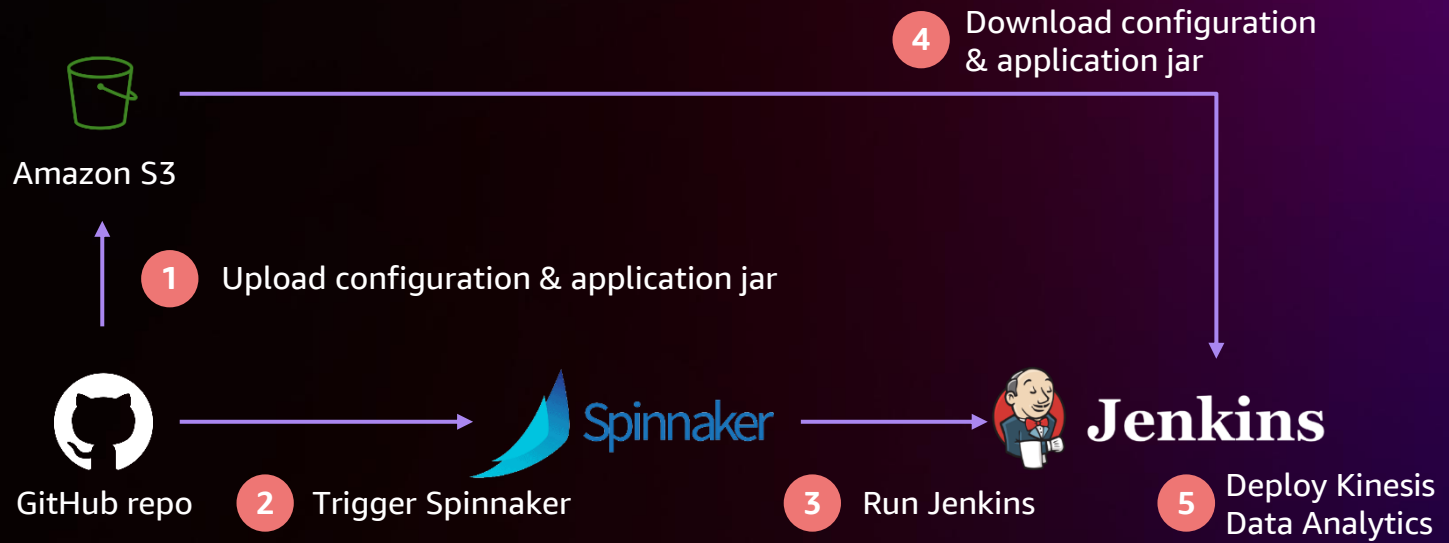


© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Further improvements: CI/CD

Benefits

- Apply configuration for multiple apps
- Continuous delivery with existing Spinnaker
- Maintain deployment history



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Jenkins logo by Frontside/CC BY-SA 3.0

We overcame three challenges with migration

Three major challenges

- Auto scaling that changes rapidly according to CPU usage
- Differences in performance and optimizations between Spark and Flink
  - Async I/O
  - DB batch read/write/delete
- Managing streaming data offsets
  - Kafka checkpoint on HDFS → Kafka/Flink save point
  - Kinesis checkpoint on DynamoDB → Flink save point

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Challenge 1

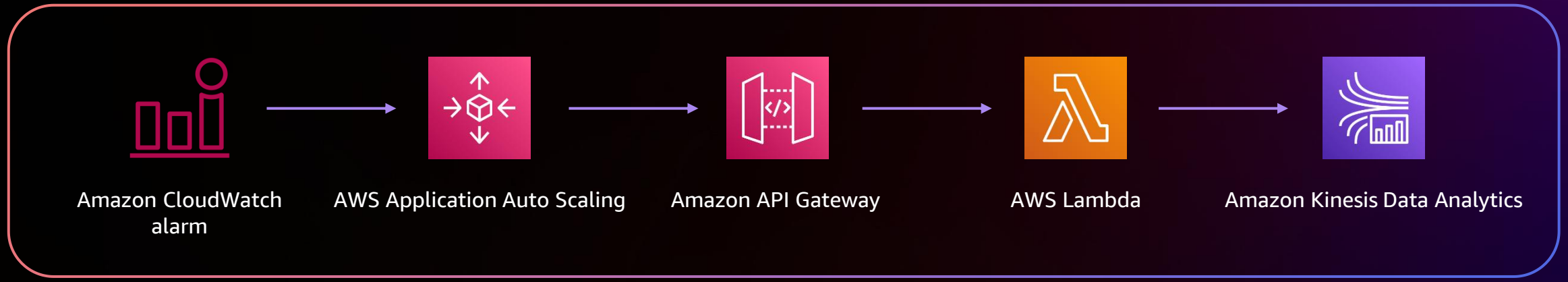
Auto scaling that changes rapidly according to CPU usage

Default Kinesis Data Analytics auto scaling

- Scale out: CPU usage 75% or more for 15 minutes (maximum statistic)
- Scale in: CPU usage less than 10% for 6 hours (maximum statistic)

Custom auto scaler

- Target tracking of Application Auto Scaling

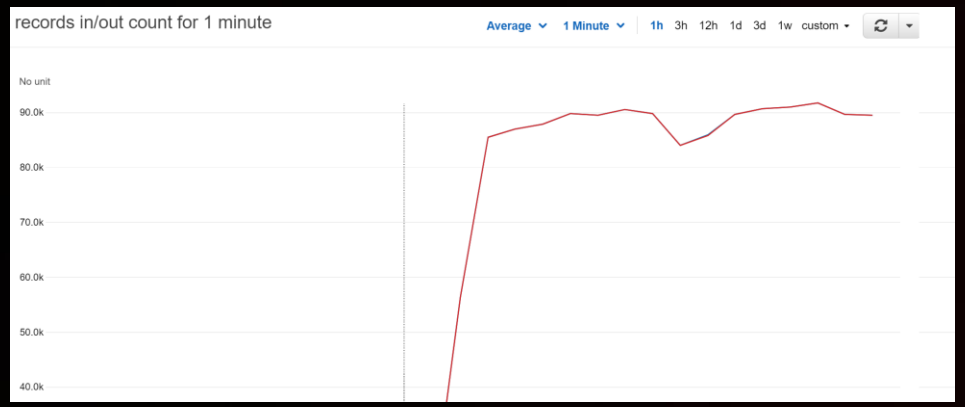
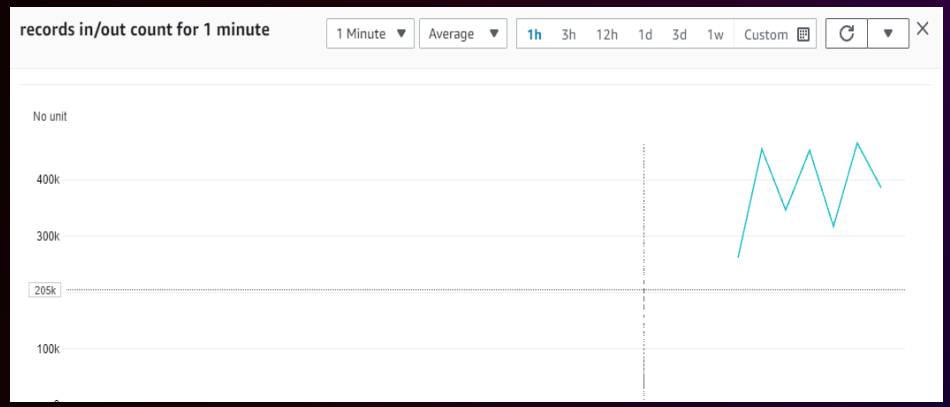


© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Challenge 2

Differences in performance and optimizations between Spark and Flink

- Async I/O
- DB batch read/write
  - HBase needs batch
  - Composition of time and count window trigger

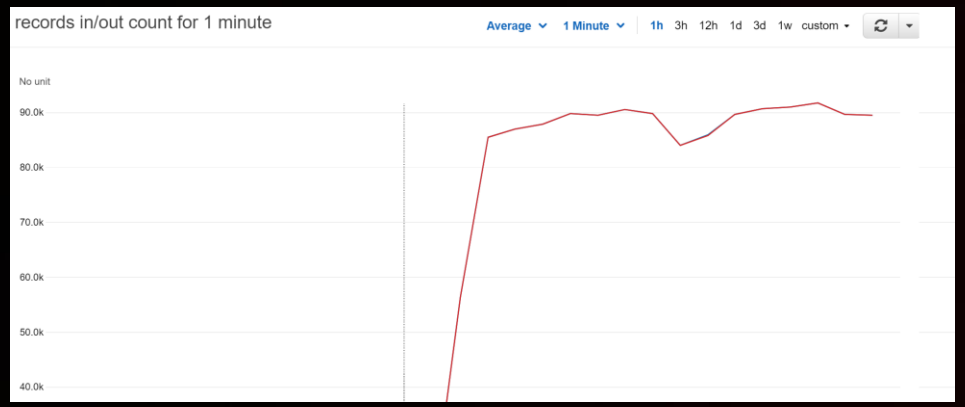
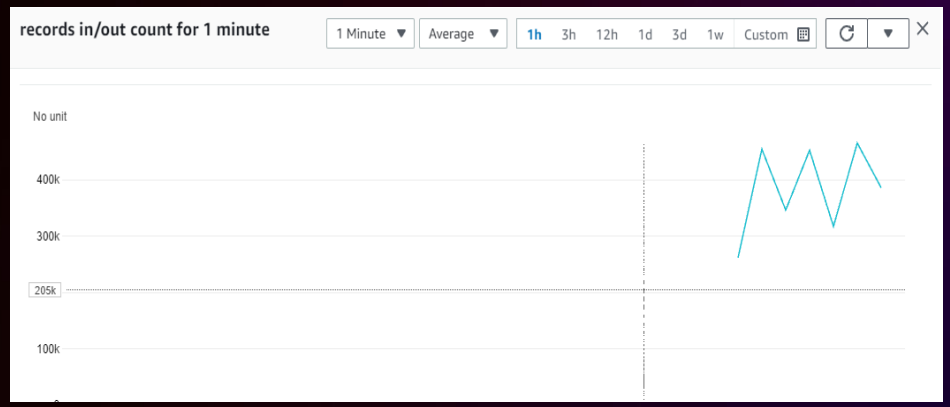
Insert each record (90K/min)	Insert batch record (400K/min)
	

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Challenge 2

Differences in performance and optimizations between Spark and Flink

- Async I/O
- DB batch read/write
  - HBase needs batch
  - Composition of time and count window trigger

Insert each record (90K/min)	Insert batch record (400K/min)
	

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Challenge 3

Managing streaming data offsets

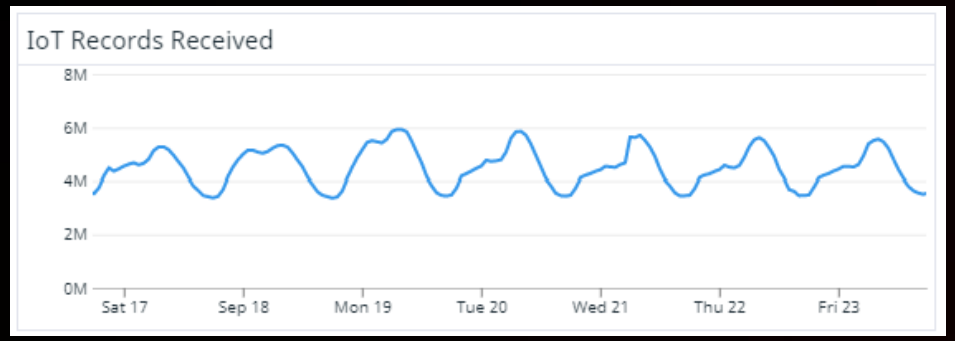
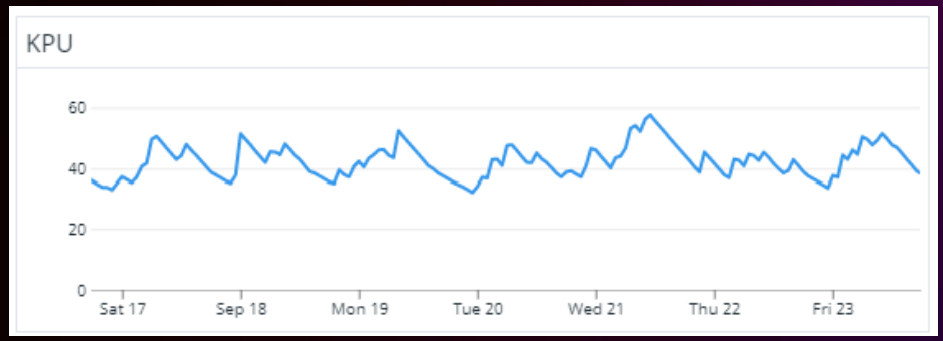
	Kafka	Kinesis
Apache Spark	• HBase table • HDFS • Amazon S3 • Kafka topic (__consumer_offsets)	• Amazon DynamoDB
Kinesis Data Analytics for Apache Flink	• Kafka topic (__consumer_offsets) • Save point	• Save point

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Impact 1: Scalability

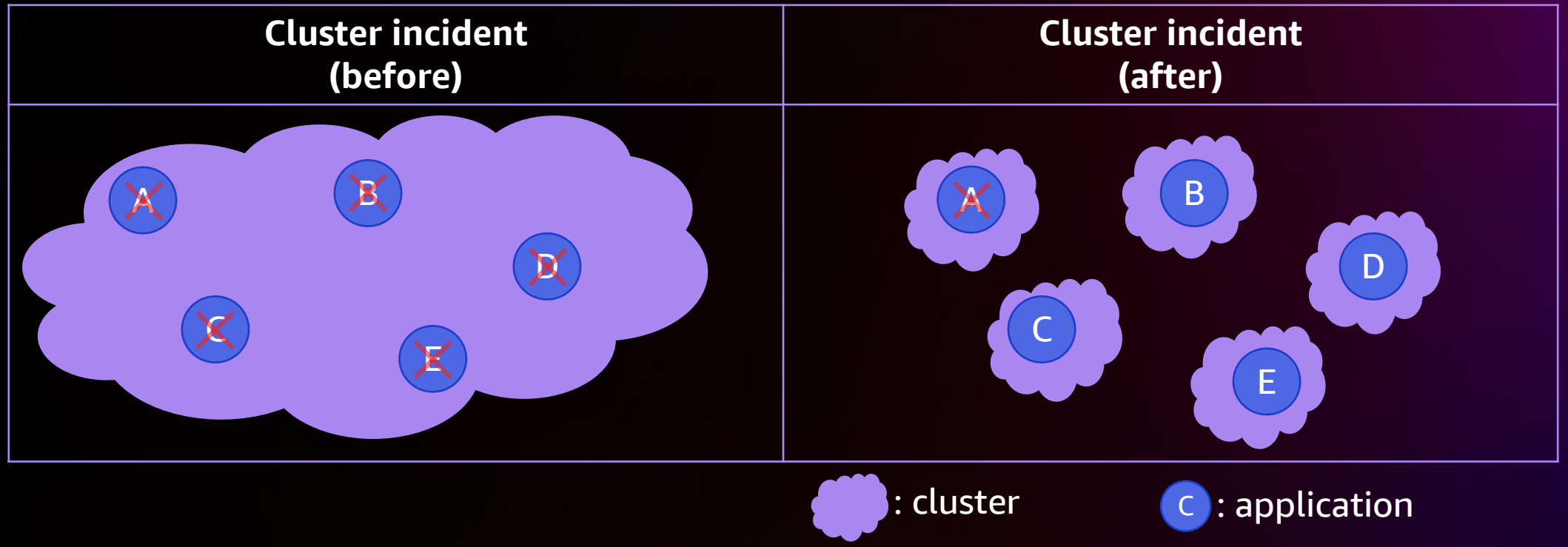
Scalable system

Events pattern	Application KPU usage pattern
	

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Impact 2: Reliability and availability

Improved reliability and availability



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Impact 3: Efficiency

Operational efficiency improvements: reduction in tickets, human intervention

131 → **53**

Data processing delay notification

28 → **0**

App failure recovery

13 → **0**

Manual cluster recovery

Reduction in TCO

500 → **600**

Application resource usage (KPU)

Spark: KPU estimation

Estimating KPU based on the resource set in Spark Livy

106 → **26**

Operation resource (M/H)

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Looking forward

Time required for deployment

- Before (Spark): under 1 minute
- After (Kinesis Data Analytics): about 2–4 minutes

Monitoring factors

- Log level per logger
- KPU metrics

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Thank you!



Please complete the session survey in the **mobile app**

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.