# INNOVATE

## DATA EDITION

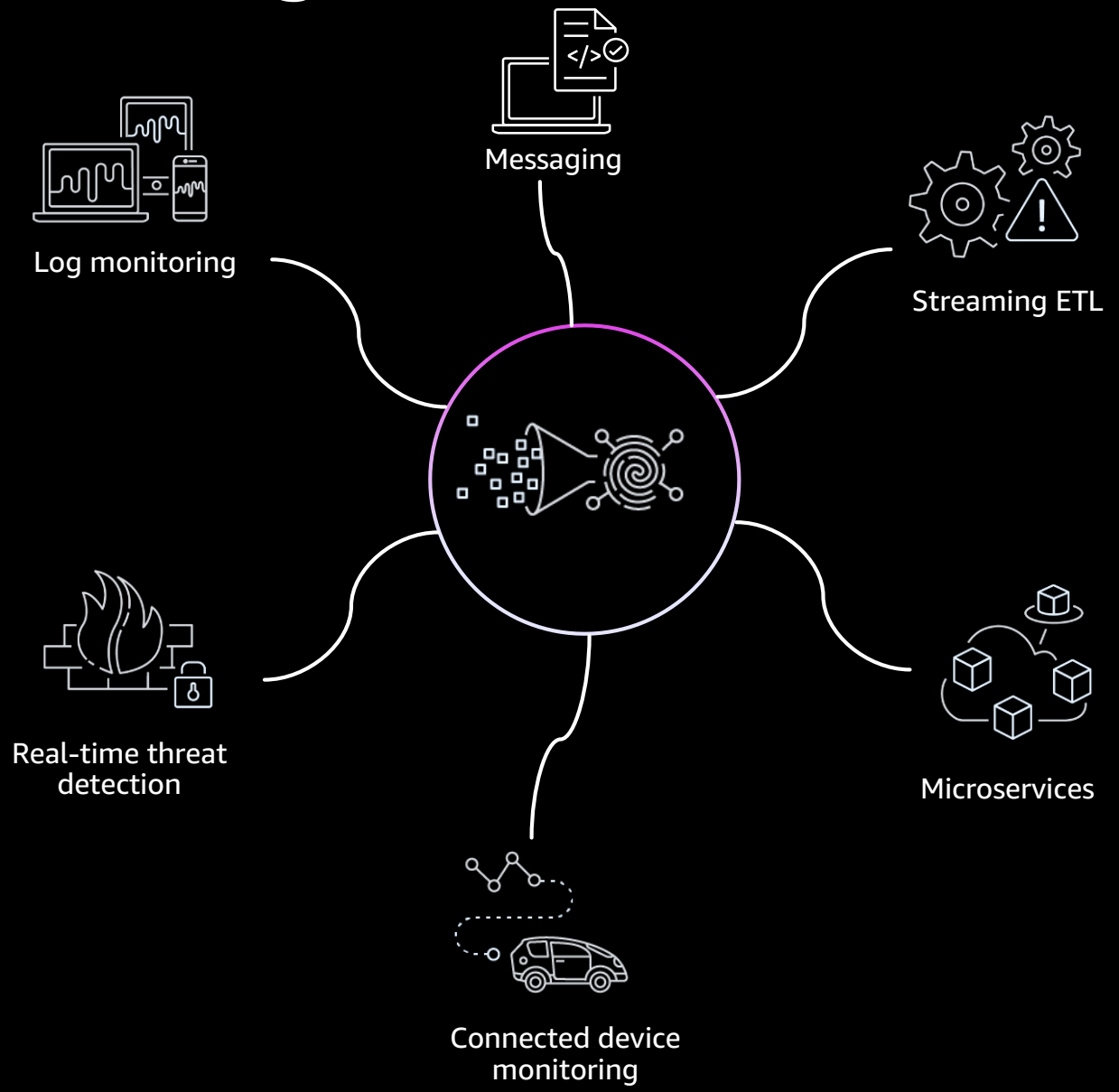
23 August, 2022

# SQL-based stream processing at scale with Amazon Managed Streaming for Apache Kafka and Amazon Kinesis Data Analytics for Apache Flink

Masudur Rahaman Sayem

Senior Analytics Solutions Architect
Amazon Web Services

# Common streaming data use-cases



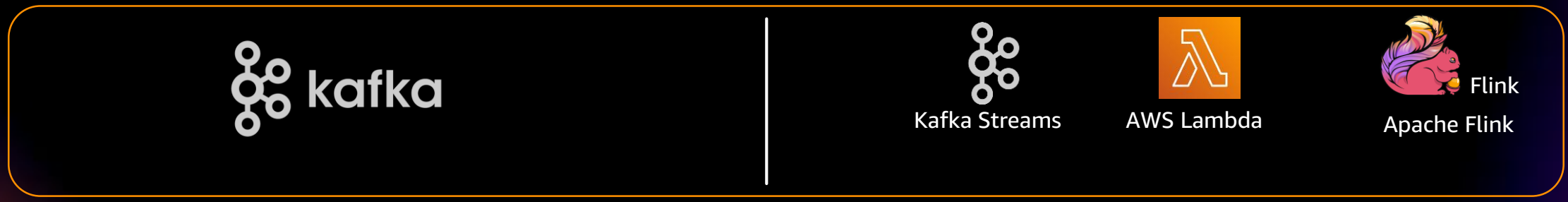
# Two key components of streaming data workload

**Stream storage**

- Data retention
- Distributed
- Record ordering
- Replayability

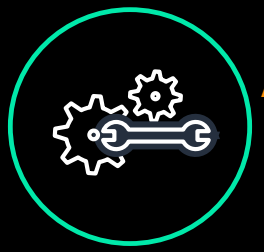
**Stream processing**

- Real-time analytics
- Streaming ETL
- Real-time alert



# Amazon Managed Streaming for Apache Kafka (Amazon MSK)

Securely stream data with a fully managed, highly available Apache Kafka service



## Automate provisioning, configuring, and tuning

Eliminate operational overhead—including the provisioning, configuration, and maintenance of Apache Kafka and Kafka Connect clusters

## Fully compatible with open source Apache Kafka

Use applications and tools built for Apache Kafka out of the box, with no application code changes required

## Highly Secure

Easily deploy secure, production-ready applications using native integrations to an Amazon Virtual Private Cloud (VPC) for authentication and authorization

## Lower Cost

Keep costs low with fully managed Apache Kafka, offered as low as 1/13th the cost of other providers

aws

# Amazon MSK Serverless

- Easily run Apache Kafka clusters without needing to right-size cluster capacity or worrying about overprovisioning
- Instantly scale I/O without needing to worry about scaling capacity up and down or reassigning partitions
- Pay for the data volume you stream and retain with throughput based pricing
- Cost effective for highly variable workloads

# Streaming data processing with Apache Kafka

AWS Lambda

Kafka Streams

Flink

Apache Flink

```
StreamsBuilder builder = new StreamsBuilder();
       KStream<String, String> textLines = builder.stream("TextLinesTopic");
       KTable<String, Long> wordCounts = textLines
           .flatMapValues(textLine ->
Arrays.asList(textLine.toLowerCase().split("\\W+")))
           .groupBy((key, word) -> word)
           .count(Materialized.<String, Long, KeyValueStore<Bytes,
byte[]>>as("counts-store"));
       wordCounts.toStream().to("WordsWithCountsTopic",
Produced.with(Serdes.String(), Serdes.Long()));

       KafkaStreams streams = new KafkaStreams(builder.build(), props);
       streams.start();
```

Sample Kafka Streams consumer

# Why Apache Flink for streaming data processing?

## Diverse use-cases

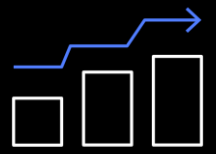
- Event-driven Applications
- Streaming Analytics & ETL
- Batch Analytics

## Expressive APIs

- SQL
- Table API
- DataStream API
- Stateful Functions

## Processing guarantees

- Exactly-once state consistency
- Event-time processing
- Late data handling



## Scale-out architecture

- Adapt to desired throughput
- Support for TBs of state

## Community

- Vibrant open source community
- Broad set of connectors

# Deep dive on Flink SQL

# Using Flink SQL for streaming data processing

- Unified API for batch and stream processing
- Develop app faster
- Declarative query language
- Industry standard
- Mature
- Simple and easy
- Open source

# Nature of Flink SQL

- SQL queries change over time
- Queries operate on top of tables
- Queries operate continuously over external tables
- Flink data processing pipelines begin with source tables
- Differs from a traditional database
- Sink tables to emit to an external system
- Source and sink could be Kafka topics, Kinesis streams, databases, filesystems, and many others
- Auto query optimization

# Flink SQL supported connectors

- Filesystem
- OpenSearch
- Apache Kafka
- Amazon Kinesis Data Streams
- JDBC
- Apache HBase
- Apache Hive

# Connectors example

## Kafka

```
CREATE TABLE MyKafkaTable (
`user` BIGINT,
  `message` STRING,
  `rowtime` TIMESTAMP(3) METADATA FROM 'timestamp',
  `proctime` AS PROCTIME(),
  WATERMARK FOR `rowtime` AS `rowtime` - INTERVAL '5'
SECOND
) WITH (
'connector' = 'Kafka',
  'topic' = 'topic_name',
  'scan.startup.mode' = 'earliest-offset',
  'properties.bootstrap.servers' = b-
3.mskkda….:9092,b-2.mskkda….amazonaws.com:9092,b-
1.mskkda….amazonaws.com:9092,
  'format' = 'json'
)
```

## OpenSearch

```
CREATE TABLE mySearchTable (
  user_id STRING,
  user_name STRING,
  uv BIGINT,
  pv BIGINT,
  PRIMARY KEY (user_id) NOT ENFORCED
) WITH (
  'connector' = 'elasticsearch-7',
  'hosts' = 'https://search-abc-xyz.ap-
southeast-2.es.amazonaws.com',
  'index' = 'users'
);
```

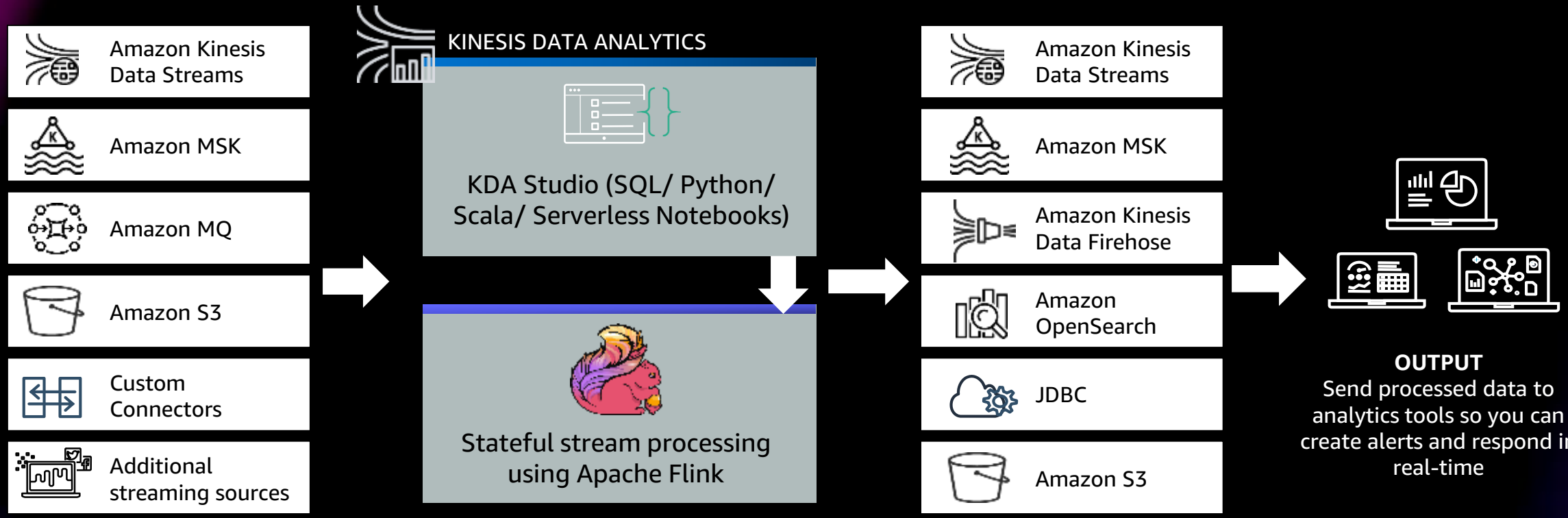
# Flink SQL built-in functions

- Comparison functions (value1 = value2)
- Logical functions (example: boolean1 OR boolean2)
- Arithmetic functions (example: numeric1 + numeric2)
- String functions (Upper, lower, trim etc.)
- Temporal functions (date, time, now etc.)
- Conditional functions
- Type Conversion functions {CAST (value AS type)}
- Aggregate functions (Count, Sum etc.)
- Pattern recognition

# Challenges of running Apache Flink

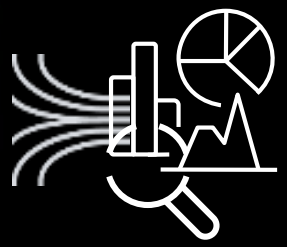
- Self-managed cluster stretched to its limits
- Not easily scalable
- Cost to upgrade unviable
- Infrastructure managed by internal team
- Real-time data enrichment is challenging

# Amazon Kinesis Data Analytics

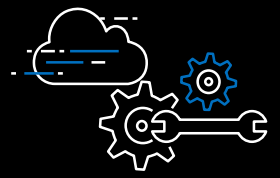



- Interact with streaming data in real-time using SQL or integrated Apache Flink applications
- Build fully managed and elastic stream processing applications

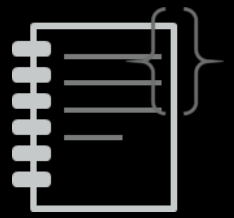
# Amazon Kinesis Data Analytics Studio



Stream inspection & visualization



Simple build and run environment



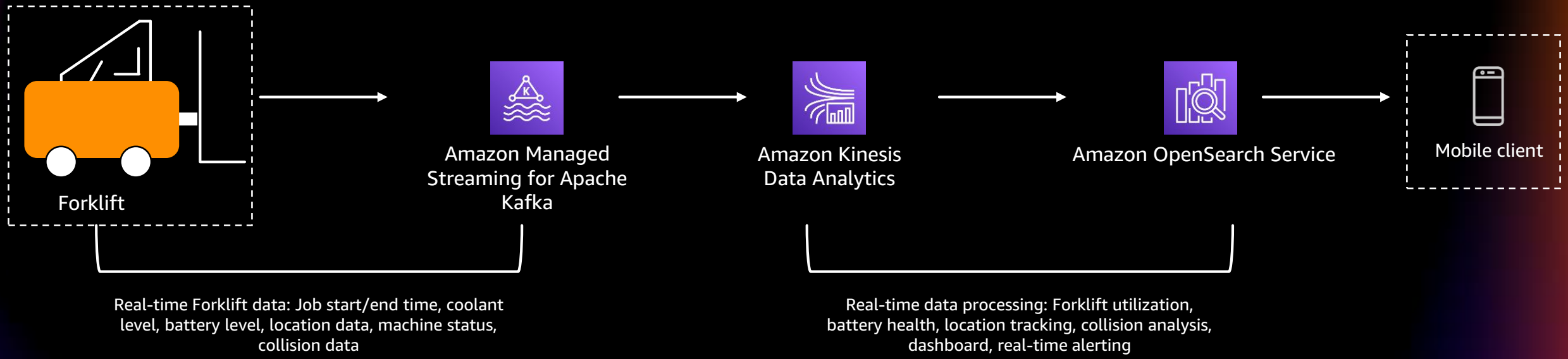
Process using SQL, Python, or Scala

Open Source

Integrates with AWS Glue Data Catalog

Serverless, rapid development of stream processing applications

# Demo – real-time fleet monitoring



# Key takeaways

Run Apache Kafka in provisioned mode or serverless with Amazon MSK

With Flink SQL focus on your application logic, not framework implementation

Flink SQL for unified batch and streaming data processing

Build faster with Amazon Kinesis Data Analytics Studio

Pattern recognition is easy with built-in MATCH_RECOGNIZE function

Run your streaming application in a serverless mode with Amazon Kinesis Data Analytics

# Other resources

- Getting started using Amazon MSK
  https://docs.aws.amazon.com/msk/latest/developerguide/getting-started.html

- Developer Guide - Amazon Kinesis Data Analytics for Apache Flink
  https://docs.aws.amazon.com/kinesisanalytics/latest/java/what-is.html

- Developer Guide - Using a Studio notebook with Kinesis Data Analytics for Apache Flink
  https://docs.aws.amazon.com/kinesisanalytics/latest/java/how-notebook.html

- Query your Amazon MSK topics interactively using Amazon Kinesis Data Analytics Studio
  https://aws.amazon.com/blogs/big-data/query-your-amazon-msk-topics-interactively-using-amazon-kinesis-data-analytics-studio/

- Amazon MSK Serverless
  https://aws.amazon.com/msk/features/msk-serverless/

# Visit the AWS Data resource hub

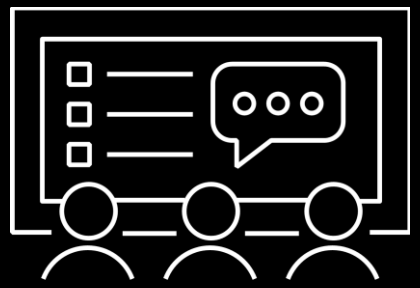
A modern data strategy can help you manage, act on, and react to your data so you can make better decisions, respond faster, and uncover new opportunities. Dive deeper with these resources today.

- Harness data to reinvent your organization
- In unpredictable times, a data strategy is key
- Make data a strategic asset
- Rewiring your culture to be data-driven
- Put your data to work with a modern analytics approach
- … and more!

https://tinyurl.com/data-hub-aws

Visit resource hub

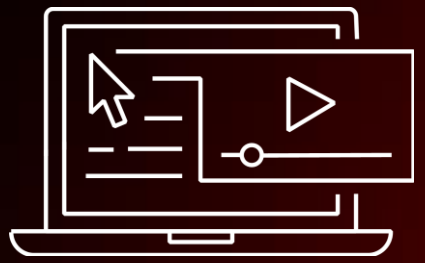
# AWS Training and Certification for Data and Analytics



## AWS Data & Analytics FREE Training Resources

Discover how to harness data, one of the world's most valuable resources, and innovate at scale.

https://bit.ly/3Ntlhy7



## AWS Data Analytics Learning Plan

This learning plan expose you to the fastest way to get answers from all your data to all your users. It can also help prepare you for the AWS Certified Data Analytics - Specialty certification exam.

https://bit.ly/3wBVjD1

## AWS Certified Data Analytics - Specialty

Earning AWS Certified Data Analytics – Specialty validates expertise in using AWS data lakes and analytics services.

https://go.aws/3lwF0RR

# Thank you for attending AWS Innovate – Data Edition

We hope you found it interesting! A kind reminder to **complete the survey.**
Let us know what you thought of today's event and how we can improve the event experience for you in the future.


aws-apj-marketing@amazon.com


twitter.com/AWSCloud


facebook.com/AmazonWebServices


youtube.com/user/AmazonWebServices


slideshare.net/AmazonWebServices


twitch.tv/aws

# Thank you!