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

ANT302

What’s new with Amazon EMR

Neil Mukerje (he/him)

Principal Product Manager, Amazon EMR
AWS

Amazon EMR

EASILY RUN SPARK, HIVE, PRESTO, HBASE, FLINK, AND MORE BIG DATA APPS ON AWS

Latest versions

Updated with latest open source frameworks within 60 days

Support for popular OSS like Spark, Flink, Hudi, Iceberg etc.

Great performance at lower cost

Spark workloads run up to 3x faster compared to open source

50–80% reduction in costs with EC2 Spot, EC2 Savings Plan, Reserved Instances

Per-second billing for flexibility

Use S3 storage

Process data in S3 securely with high performance using the EMRFS connector

Scale Compute and Storage independent of each other

Multiple deployment options

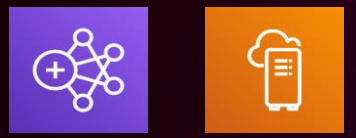
Fully managed, choose multiple deployment options based on your need

Flexible deployment options

Amazon EMR on
Amazon EC2

Amazon EMR on
Amazon EKS

Amazon EMR on
AWS Outposts



NEW!

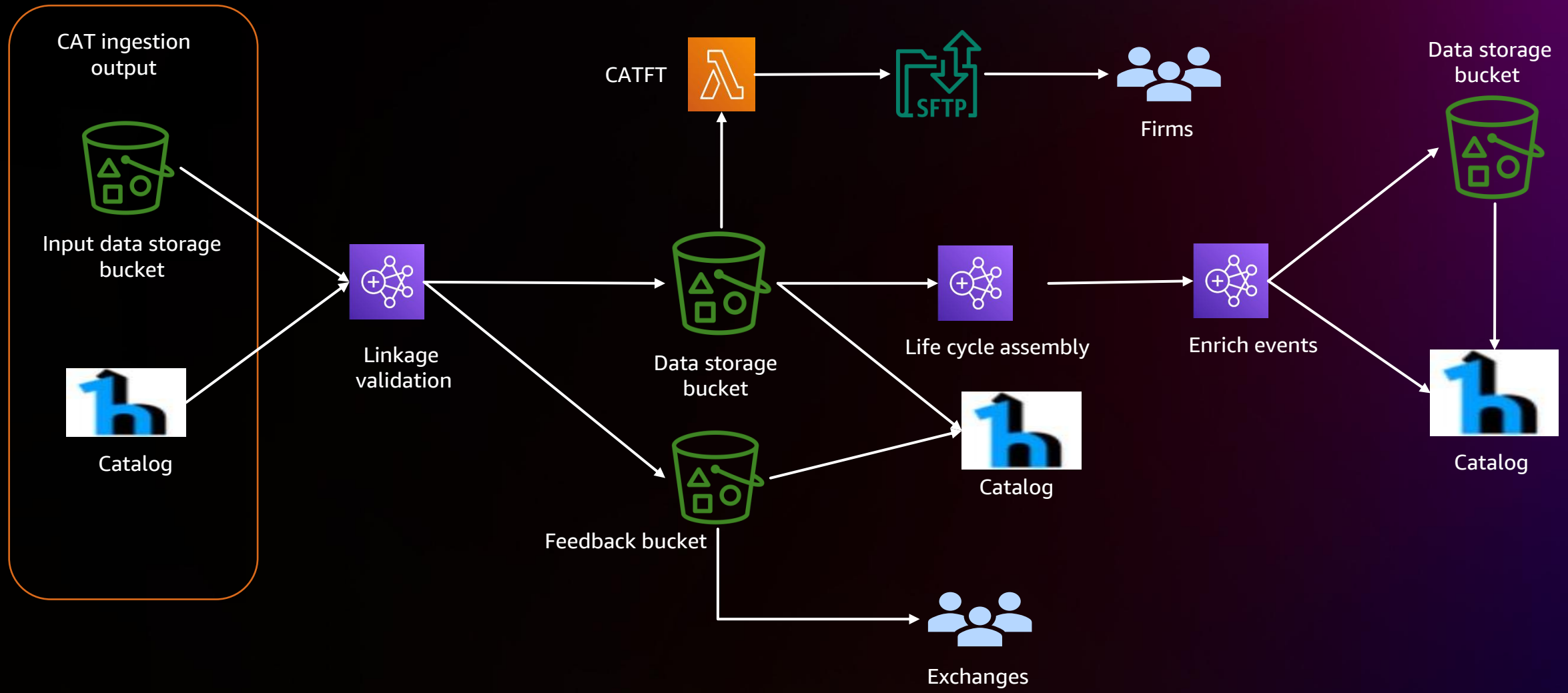
Amazon EMR
Serverless

“We’ve been using EMR to power our data mesh data processing needs for dozens of teams. Each team is free to decide the compute resources they need and scale appropriately for each of their use cases.”

Oliver Fenton

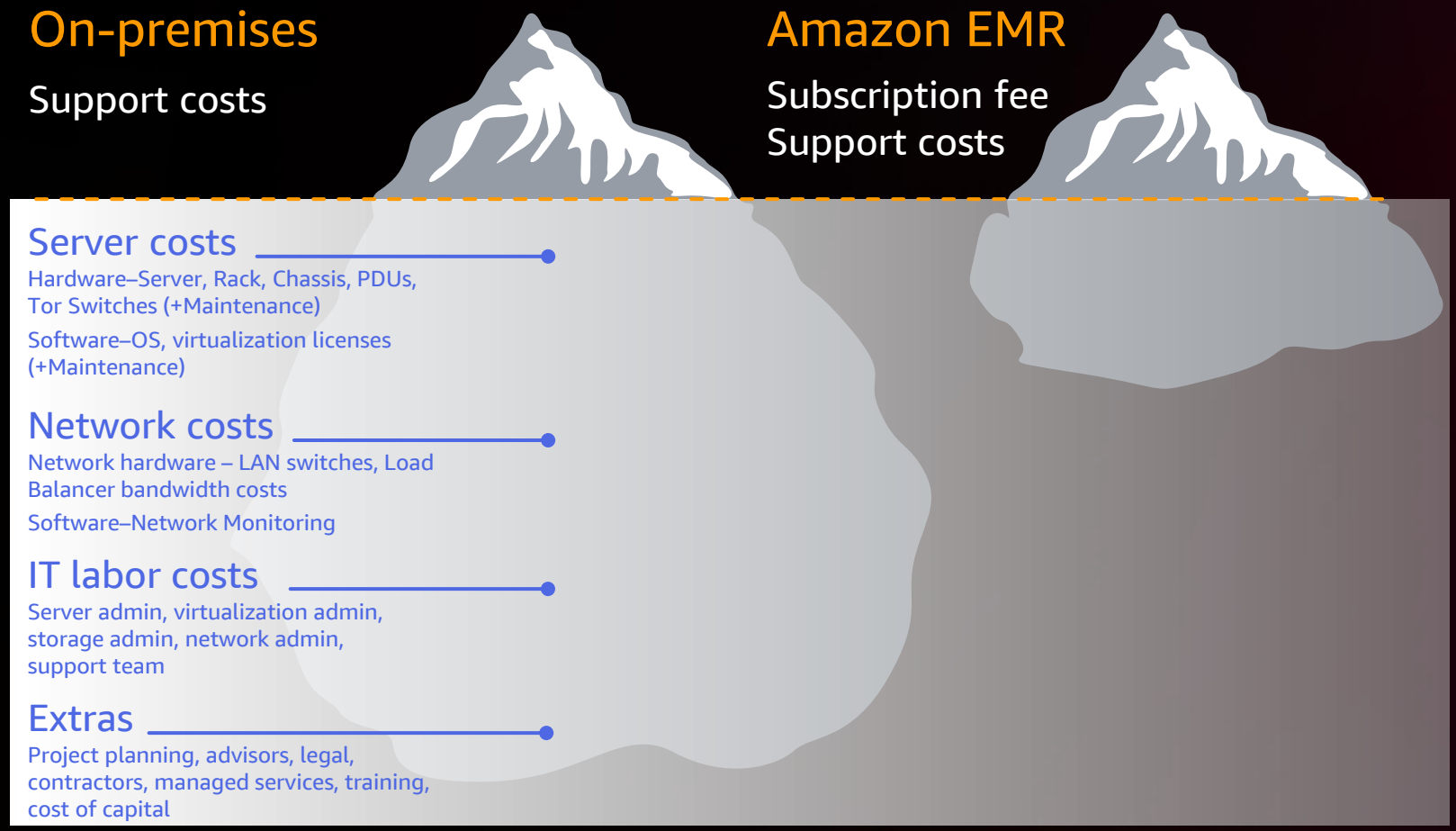
Director of Data and ML Platform, Glovo

FINRA CAT Processing Architecture



Lower costs

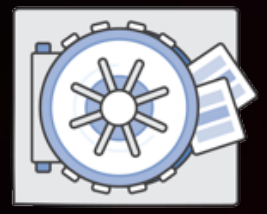
LOWER TCO



- Less admin time to manage, and support Hadoop clusters
- No up-front costs – hardware acquisition, installation
- Save on operating costs – data center space, power, cooling
- Business value: Cost of delays, risk premium, competitive abilities, governance, etc.

Cost optimization options

Transient clusters

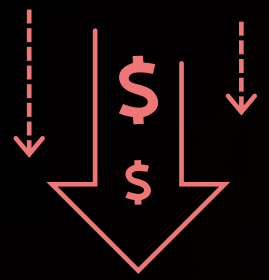


Savings plan and reserved instances

Spot instances & instance fleets

Amazon EC2 Spot instances

ACCELERATE COMPUTE FOR LESS



Low, predictable prices

Up to 90% discount over on-demand prices

Faster results

Increase throughput up to 10x while staying in budget

Easy to use

Launch through AWS services (e.g., ECS, EKS, Batch, Amazon EMR) or integrated third-parties

AWS Graviton2 instances have the best price performance for workloads in Amazon EC2

We compared M5 versus M6g using EMR 5.30.1 using TPC DS 3TB benchmark queries with data in Amazon S3

12%–16% improvement in performance
compared to M5 instance types

20% lower cost
vs same-sized comparable M5 instances

Up to 30% better price-performance

Managed scaling feature overview

COMPLETELY MANAGED ENVIRONMENT FOR AUTOMATICALLY RESIZING AMAZON EMR ON EC2 CLUSTERS

Amazon EMR managed algorithm that constantly improves, giving you a completely managed experience

High-resolution metrics enabled with managed scaling

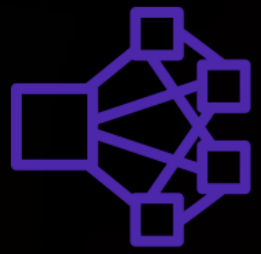
Only min/max cost constraints configurations required

More data points and faster reaction time than earlier autoscaling feature

Save 20-60% costs depending on your workload patterns

NEW!

Amazon EMR on EC2 enhancements



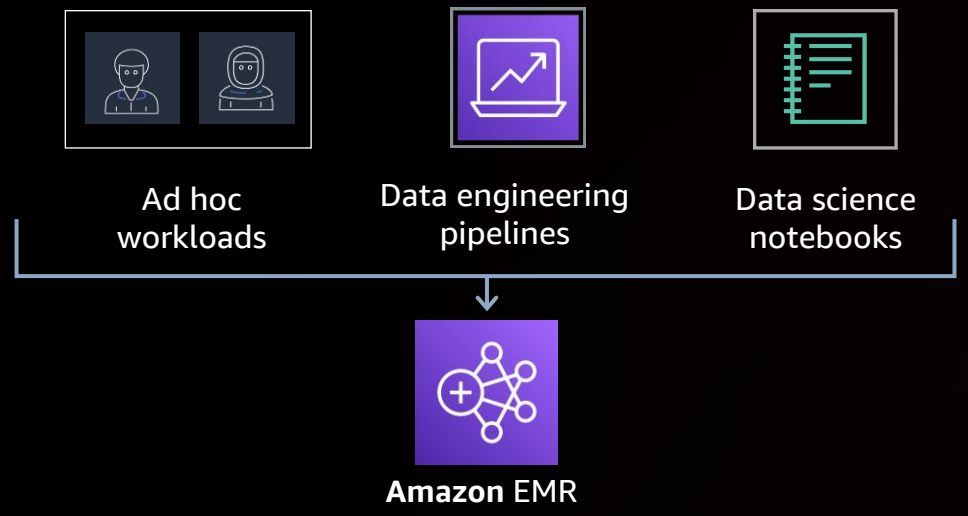
Reduce start-up time for Amazon EMR on EC2 cluster in a private subnet by up to 30%

Reduce task nodes start-up time for Amazon EMR on EC2 cluster by up to 30%

Better performance and lower costs with Spark shuffle awareness in managed scaling

Reduce costs and improve performance with EBS GP3 volumes

Amazon EMR on Amazon EKS



- Optimized runtime; runs 3.9x faster
- Consolidate infrastructure across organizations
- Manage resource limits by teams and workload
- Start jobs quickly; no cluster provisioning delays
- Run application on single AZ or across multiple AZs
- Choose serverless with AWS Fargate on Amazon EKS

“Migration to EMR on EKS from open-source Spark on Kubernetes helped us to consolidate on two fronts – multiple Spark versions on the same EKS cluster and Spark workloads alongside other workloads on the same EKS cluster.

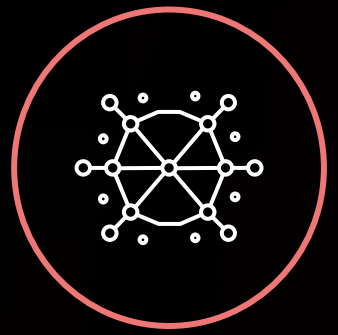
This consolidation led to 45–55% reduction in infrastructure cost while reducing job failures due to timeouts by 25% and job failures due to out-of-memory errors by 90%.”

Ujjwal Sarin

Data Platform Engineer, Stitch Fix

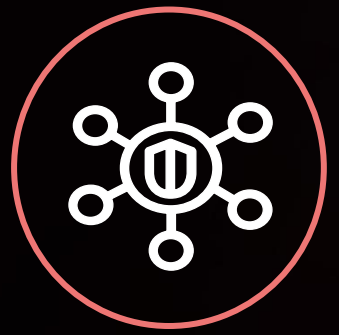
NEW!

Building applications on Amazon EMR on EKS



Job templates

Simplifies job authoring experience for data engineers and scientists by enforcing common parameters



Spark-SQL runner

Execute SQL scripts directly with API without writing any additional code

DynamoDB connector

Easily access data in DynamoDb

Enhanced job failure messages

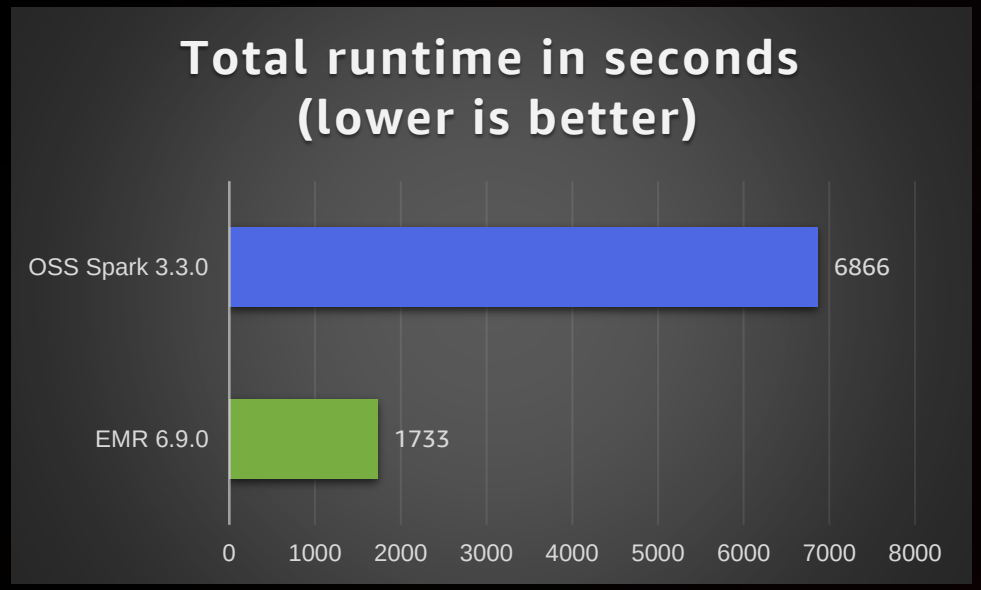
Show task failure messages in DescribeJobRun API and driver logs for job failures

Performance optimized runtimes

NEW!

Amazon EMR runtime vs. OSS Spark

UP TO 3.9X FASTER PERFORMANCE FOR APACHE SPARK 3.3.0



Spark 3.3.0 on EMR 6.9.0

*Based on TPC-DS 3TB Benchmarking running 6 node C5.9XL cluster and EMR 6.9.0 running Spark 3.3.0

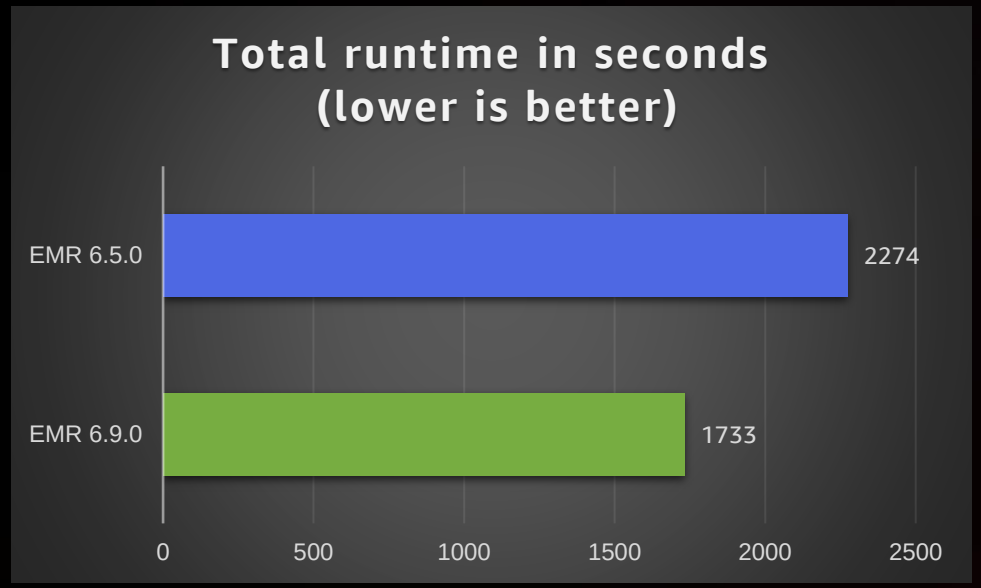
Runtime built on a optimized version of Apache Spark

Best performance

- **Up to 3.9x faster on total time**

NEW!

Amazon EMR runtime for Apache Spark: Performance improvements - 2022



Spark 3.3.0 on EMR 6.9.0

*Based on TPC-DS 3TB Benchmarking running 6 node C5.9XL cluster and Amazon EMR 6.5.0 running Spark 3.0

Runtime built on a optimized version of Apache Spark

Best performance

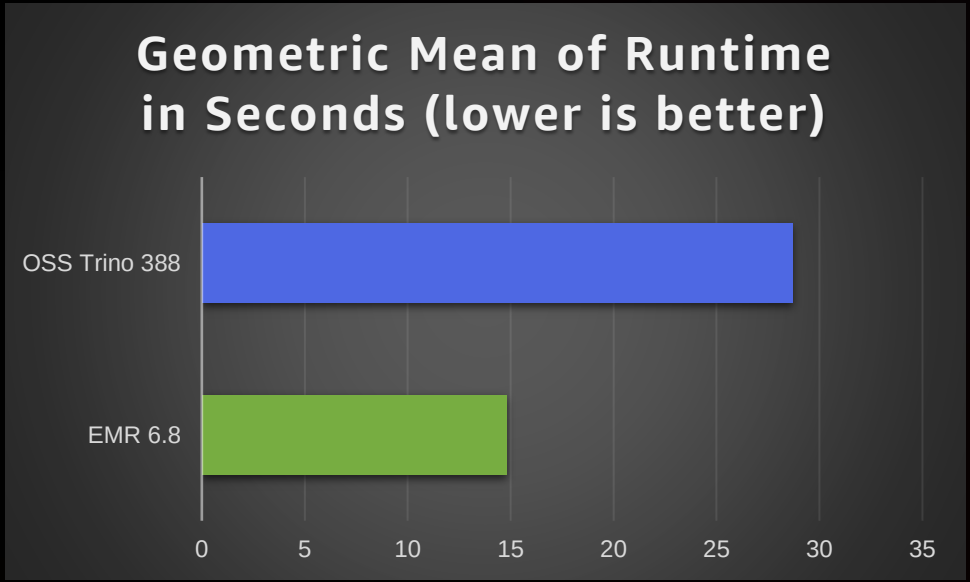
- **Up to 1.3X faster on total time**

100% compliant with Apache Spark APIs

NEW!

Amazon EMR runtime vs. OSS Trino

UP TO 3.1X FASTER PERFORMANCE FOR APACHE TRINO 388



Runtime built on a optimized version of Trino

Best performance

- **Up to 3.1x faster on geometric mean**
- **Up to 4.2x faster for total time**

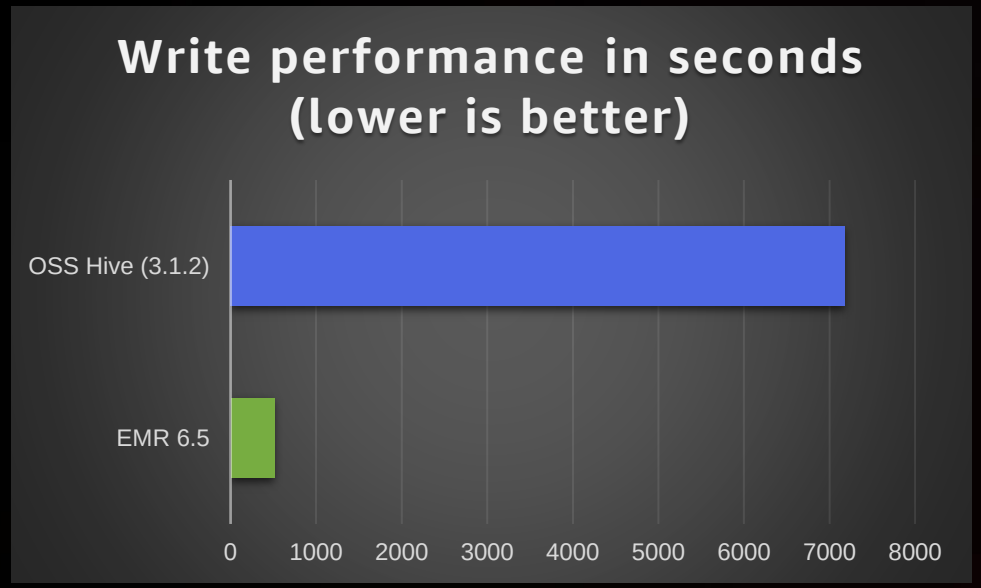
Trino 388 on EMR 6.9.0

*Based on TPC-DS 3TB Benchmarking running 6 node C5.9XL cluster and EMR 6.9.0 running Trino 388

NEW!

Amazon EMR runtime vs. OSS Hive

UP TO 15X FASTER PERFORMANCE FOR APACHE HIVE WRITES USING EMRFS S3-OPTIMIZED COMMITTER



Hive 3.1.2 on EMR 6.5.0

*Based on load phase of TPCx-BB 1 TB Benchmarking running 1 m5d.8xlarge master node, 20 m5d.8xlarge core nodes cluster on EMR 6.5.0 running Hive 3.1.2

Hive write performance with the Amazon EMR Hive zero-rename feature

Best performance:

- **Up to 15x faster writes**

Engine improvements

NEW!

Presto/Trino improvements

Spot loss handling

Improved join reordering, selection and common subexpression removal

Enforce fine-grained lake formation policies (Presto)

Iceberg and Delta support (Trino)

HDFS for checkpointing (Trino)

Hive improvements

NEW!

Zero-rename Writes

MSCK Optimization

Parquet Modular Encryption

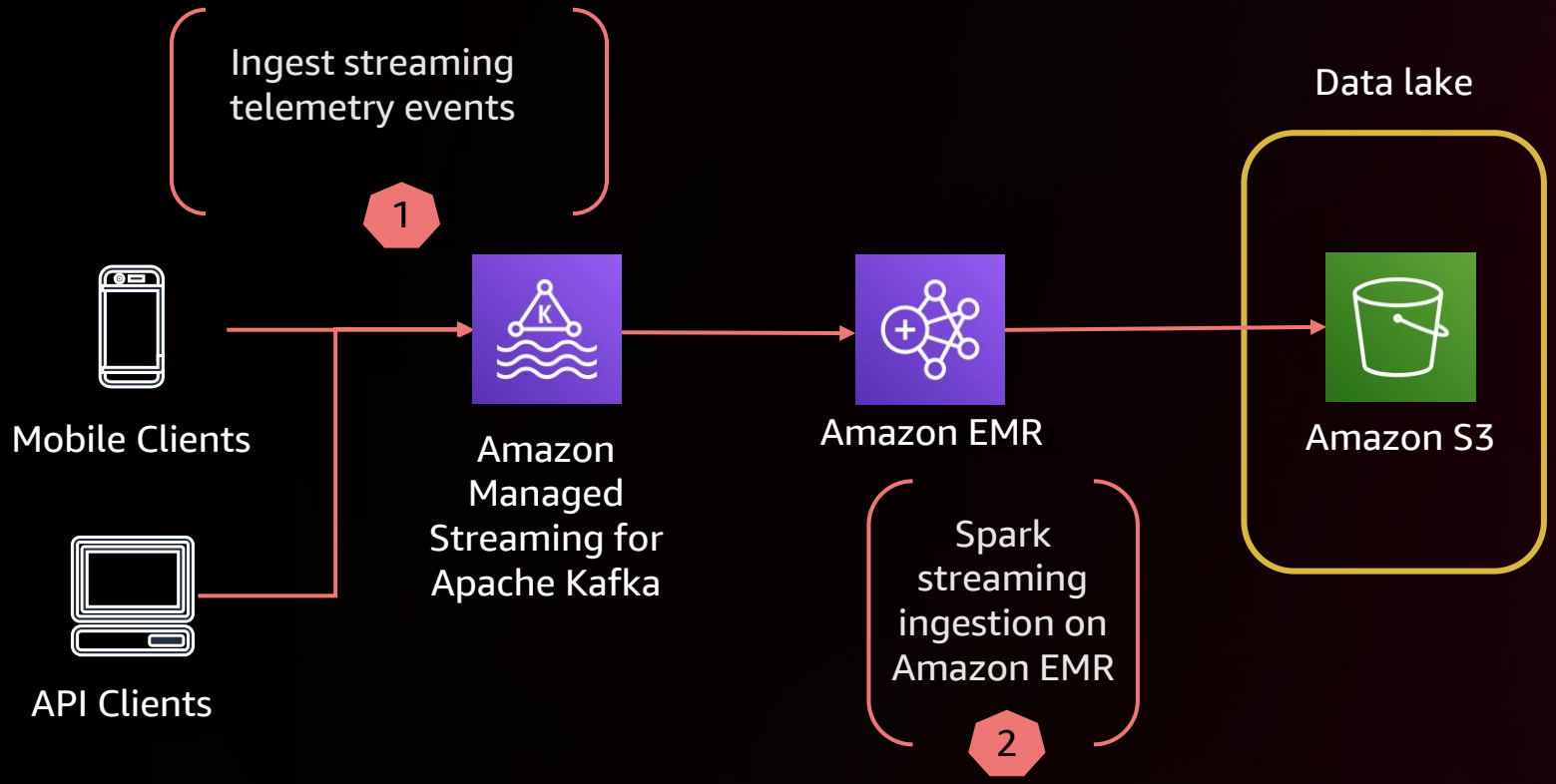
Fine-grained lake formation based policies

Incorporated 300+ improvements, critical fixes into Amazon EMR Hive

Iceberg support in Amazon EMR 6.9

Transactional data lakes: Transactions and record-level updates/deletes to data lakes

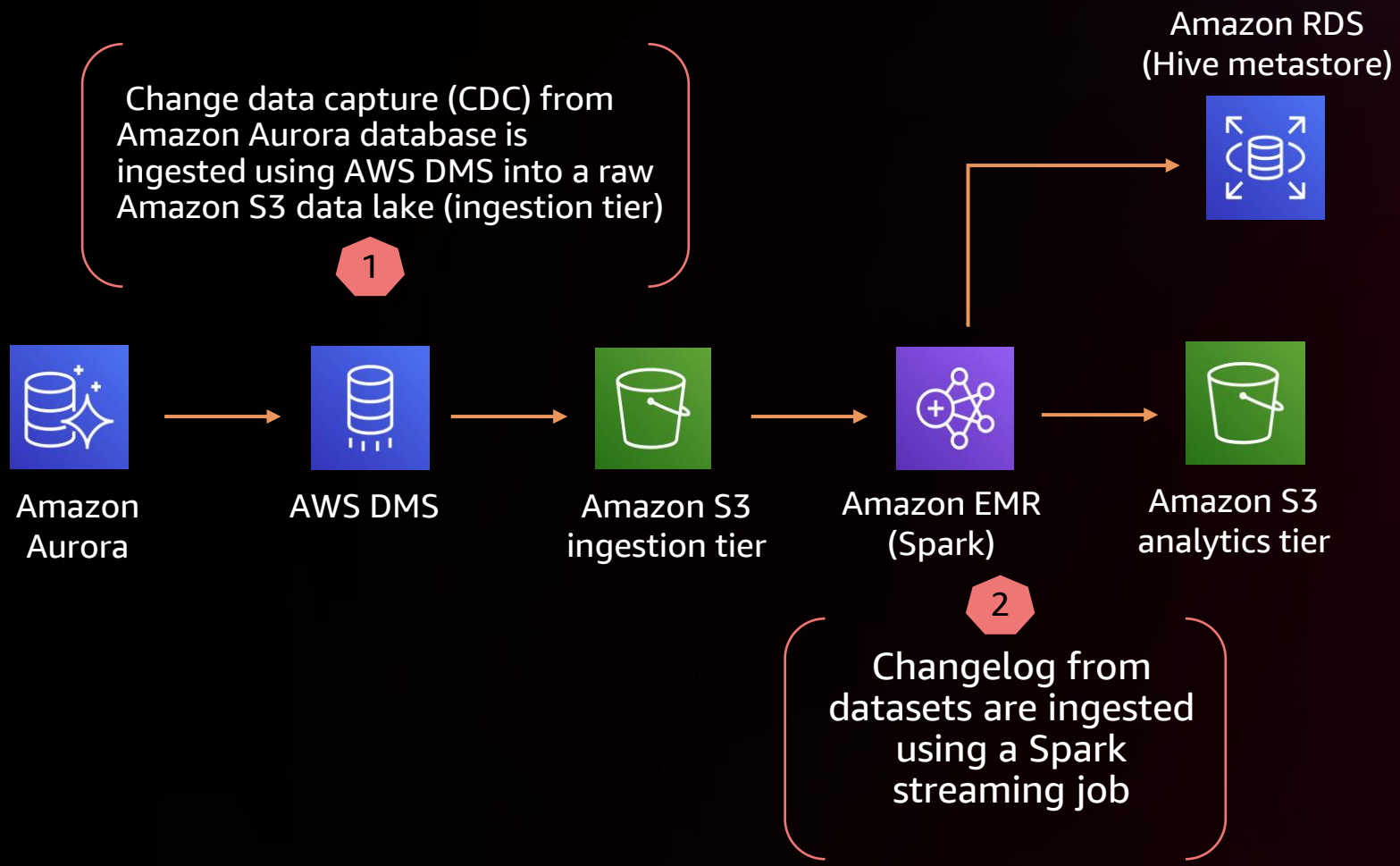
Streaming ingestion pipeline



Challenges

- Make atomic changes
- Reader–writer isolation
- High throughput ingestion
- Small file compactions
- Row-level upserts and deletes

CDC ingestion pipeline



Challenges

- Make atomic changes
- Reader–writer isolation
- High throughput ingestion
- Small file compactions
- Row-level upserts and deletes
- Clustering by secondary keys

Transactional data lakes

CHOICE OF FRAMEWORK FOR EACH WORKLOAD

EMR 6.9 includes:

Apache Hudi 0.12

Apache Iceberg 0.14.1

OSS Delta Lake 2.1.0

Transactional data lakes features

TRANSACTIONS, RECORD-LEVEL UPDATES/DELETES, AND CHANGE STREAMS TO DATA LAKES

Ingestion

- Transactions (ACID) - Reader and writer isolation
- Transactions (ACID) - Concurrent write support
- Record level upserts and deletes
- High throughput streaming ingestion
- Spark, Flink, and Java Writer Support
- Automatic compaction of small files
- SQL DML support

Query

- Spark, PrestoDB/Trino, Flink, Hive Support
- Efficient queries across partitions and files
- Incremental query support
- Time travel query support

Transactional data lakes features

TRANSACTIONS, RECORD-LEVEL UPDATES/DELETES AND CHANGE STREAMS TO DATA LAKES!

Administration

- Async background compaction of files
- Async background sorting and clustering of keys
- Automatically clean up files beyond retention period
- Metrics for past commits or rollbacks

Apache Hudi

RICH PLATFORM TO BUILD STREAMING DATA LAKES WITH INCREMENTAL DATA PIPELINES

Apache Hudi 0.12

EMR 6.9 includes Hudi 0.12.

Key new features include:

- Multi-modal indexes: Improve the lookup performance in file index and query latency with data skipping
- Async indexer service: Index columns in the background without affecting writes
- Schema-on-read for Spark: Improved Schema evolution support

NEW!

Apache Iceberg

OPEN TABLE FORMAT FOR HUGE ANALYTIC DATASETS

Apache Iceberg 0.14.1 is packaged as a library for Spark3 Runtime, Trino, Flink, and Hive in EMR 6.9.0.

Key new features include:

Apache Iceberg

- Time travel support with Spark SQL and Trino SQL
- Merge on Read (MoR) support
- Optimistic concurrency with AWS Glue Data Catalog
- Disaster recovery with S3 access points
- Flink and Hive integration (EMR 6.9.0)

NEW!

OSS Delta Lake

OPEN-SOURCE STORAGE FRAMEWORK THAT ENABLES BUILDING A LAKEHOUSE ARCHITECTURE

OSS Delta Lake 2.1.0

OSS Delta Lake 2.1.0 is packaged as a library in EMR 6.9.0

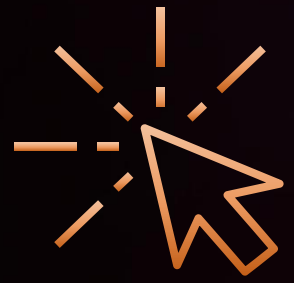
Engines supported : Spark3 and Trino

To learn more :

https://docs.aws.amazon.com/emr/latest/ReleaseGuide/emr-delta.html

Amazon EMR Serverless

All the benefits of EMR without managing clusters and servers



Run frameworks more easily; just pick a version and run

Automatically scale; don't guess cluster sizes

Optimize cost; Automatic and fine-grained scaling reduces cost

Performance-optimized version delivers 2x better performance

Multi-AZ resiliency from day 1

Integration with familiar tools like Apache Airflow

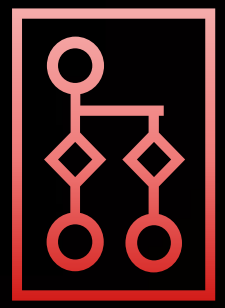
“We recently migrated over 50 Hive and Spark workloads to EMR Serverless from a third-party platform. The migration has been smooth with minimal source code change, native integration with Apache Airflow, and we were able to leverage our existing AWS Glue Data Catalog as metastore.

Finally, we launched our first EMR Serverless application with just a few clicks on the EMR Studio console.”

Sathwik Shirsat

Engineering Manager, Data & Analytics, , Malwarebytes

Jobs



- Run jobs on applications
- Can run multiple jobs on an application
- Can control authorization using per-job execution role

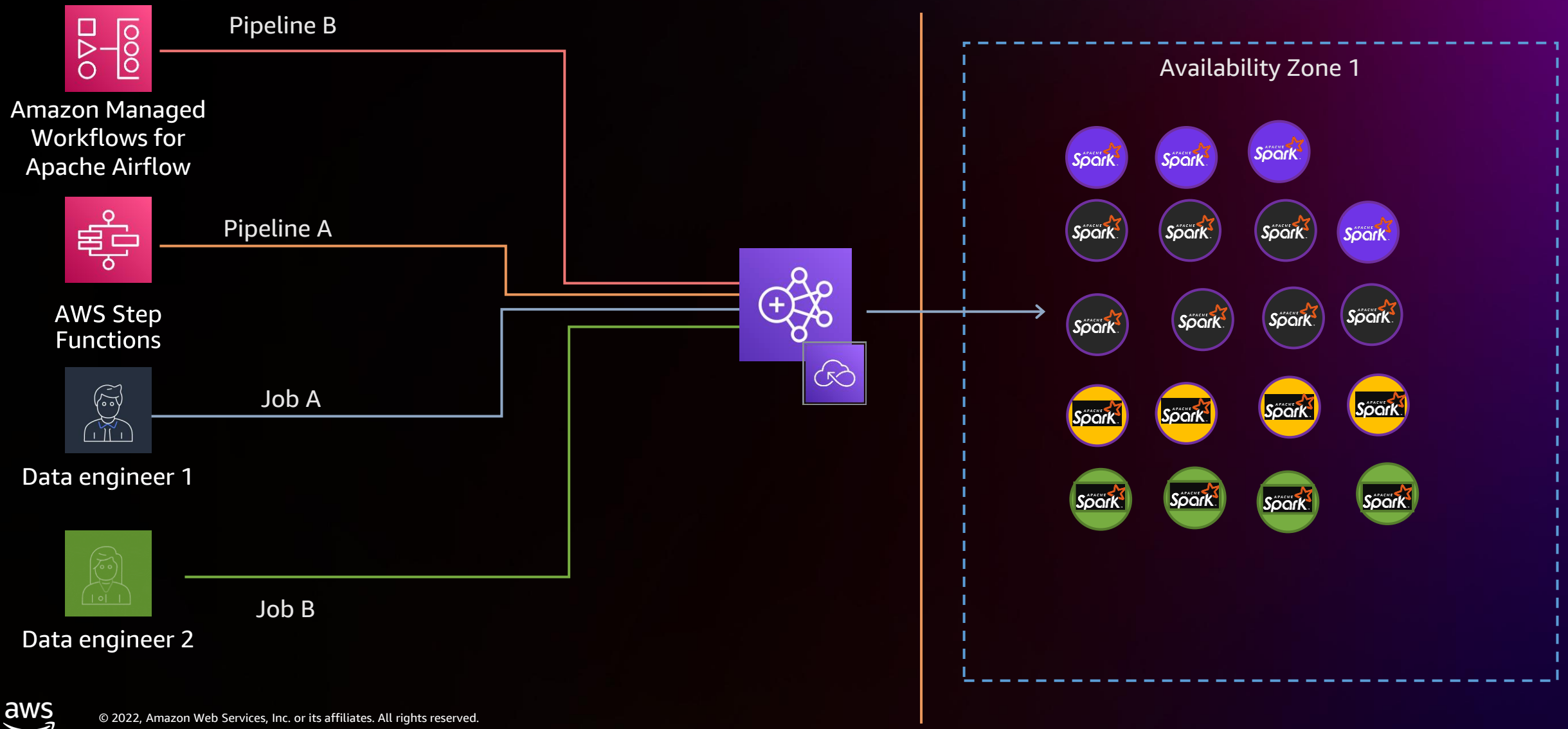
Workers

- Internally used to execute your workloads
- Workers run the OSS framework you choose
- You can change the size of workers to control performance

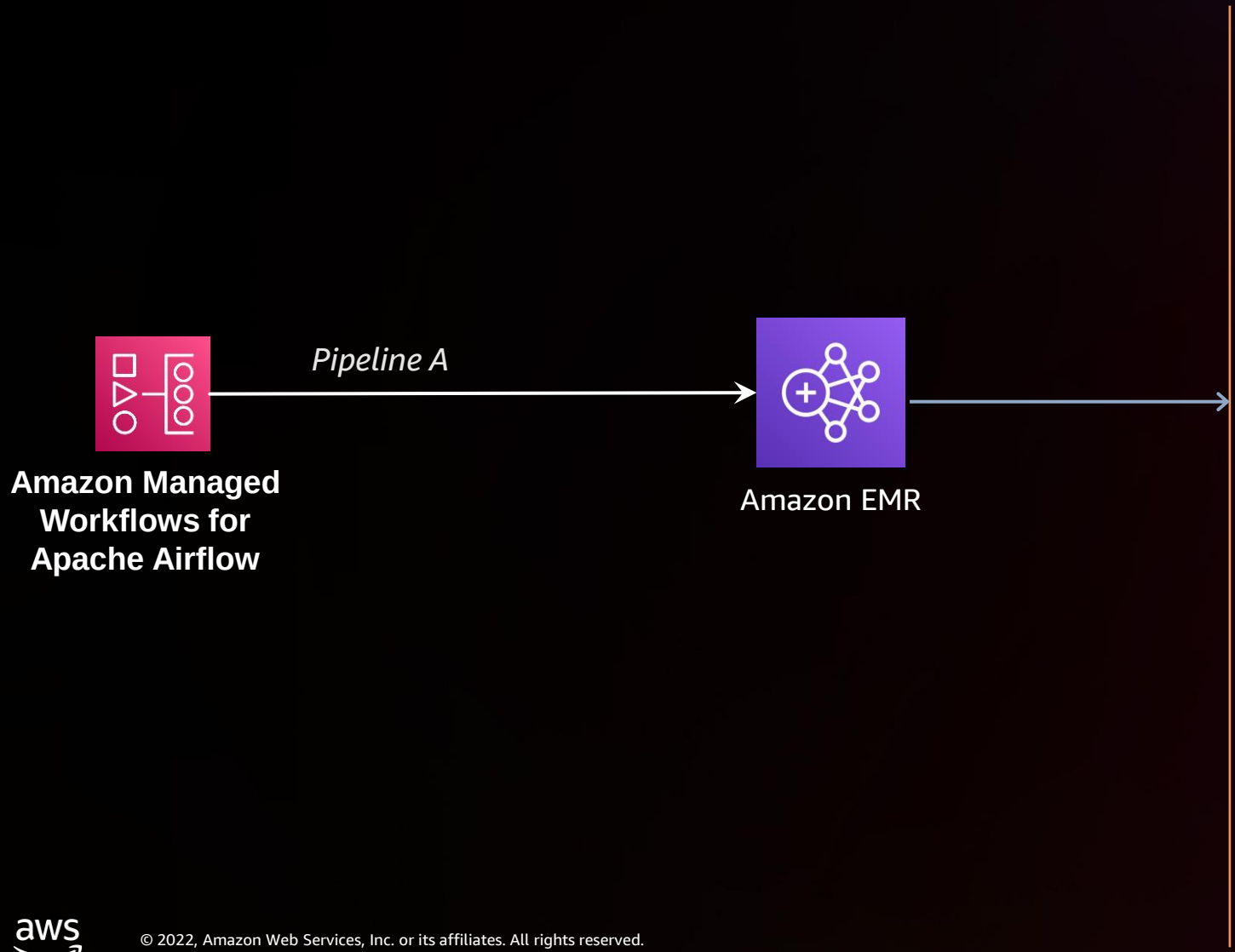
Pre-initialized workers

- Optional feature to pre-initialize workers
- Jobs start immediately
- Helps you maintain a warm pool

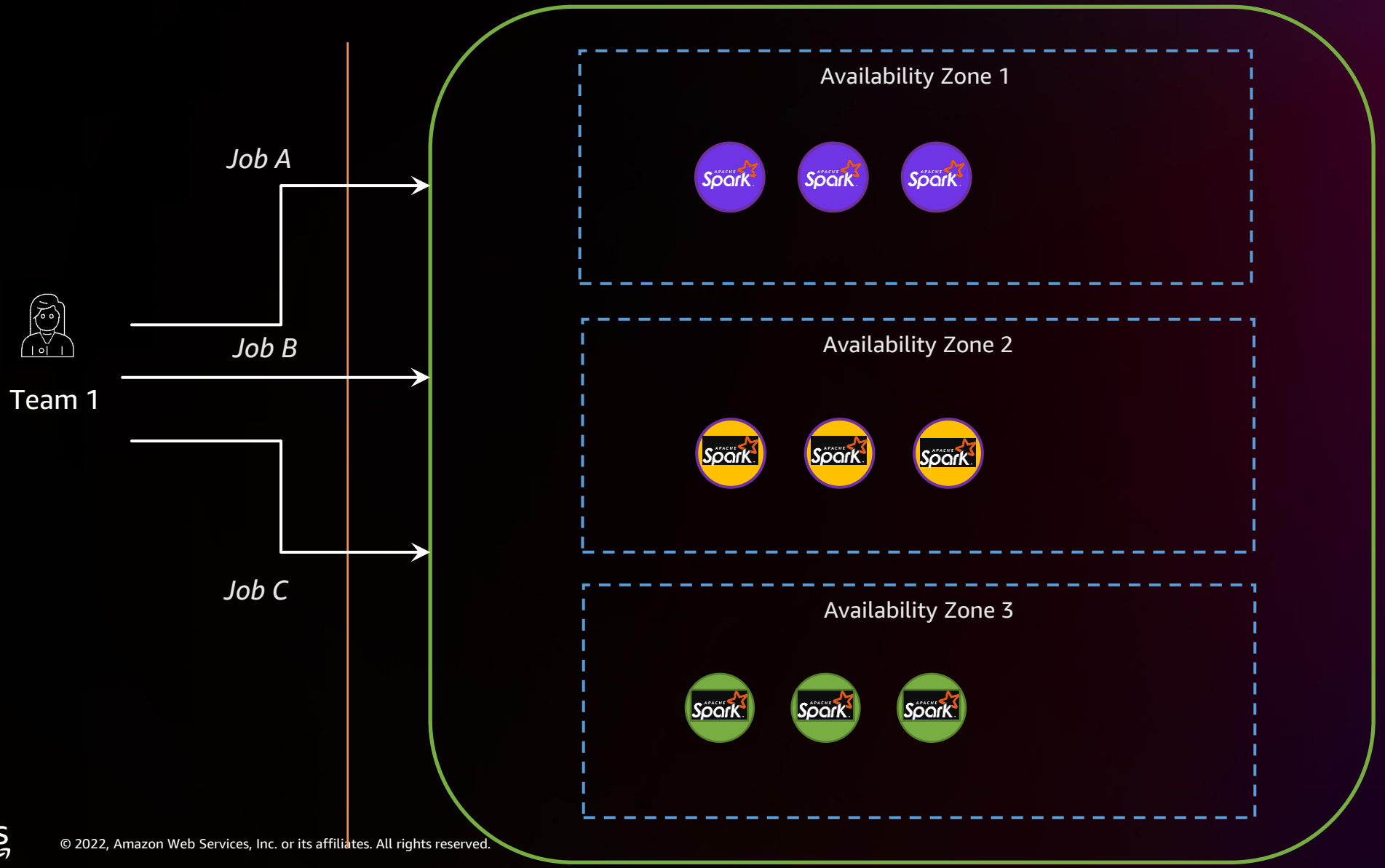
Shared applications with Amazon EMR Serverless



Multi-AZ out of the box



Isolated compute environments with defined guardrails

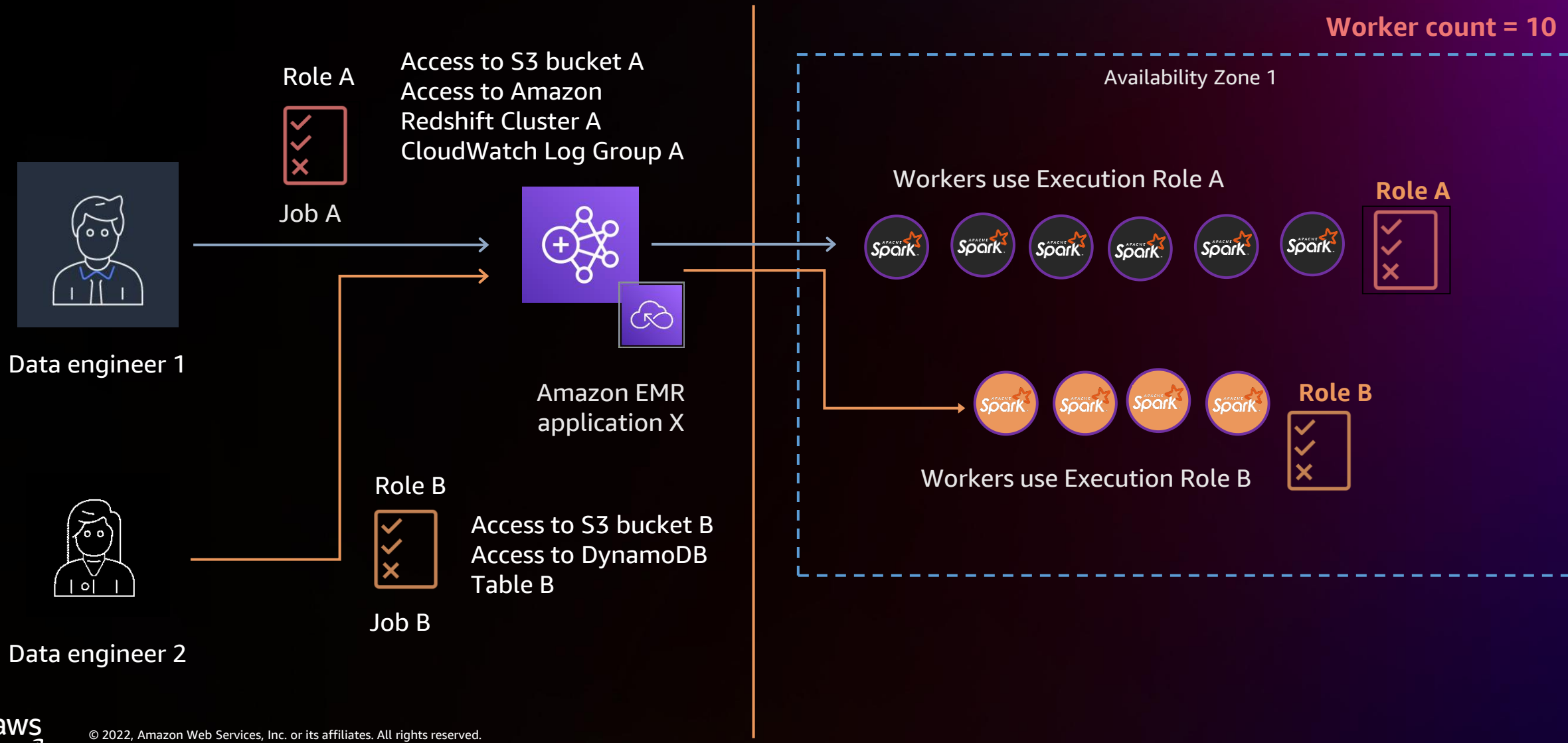


Max vCPUs:
4,000

Max Memory GB:
16,000

Max Storage:
120,000 GB

Per-job execution role



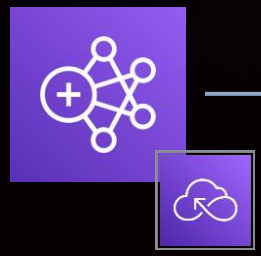
Debugging Jobs is super simple

Worker count = 6

Availability Zone 1

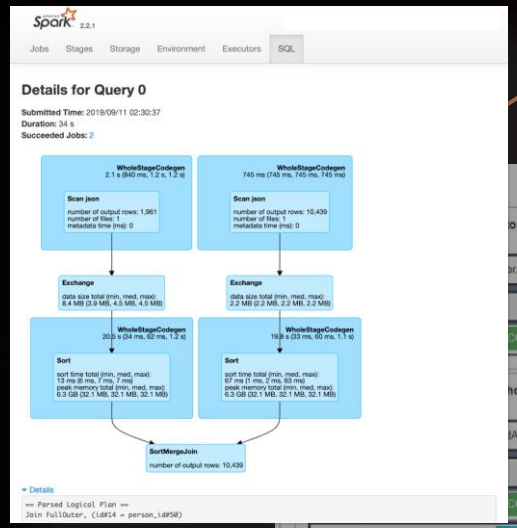
Data engineer 1

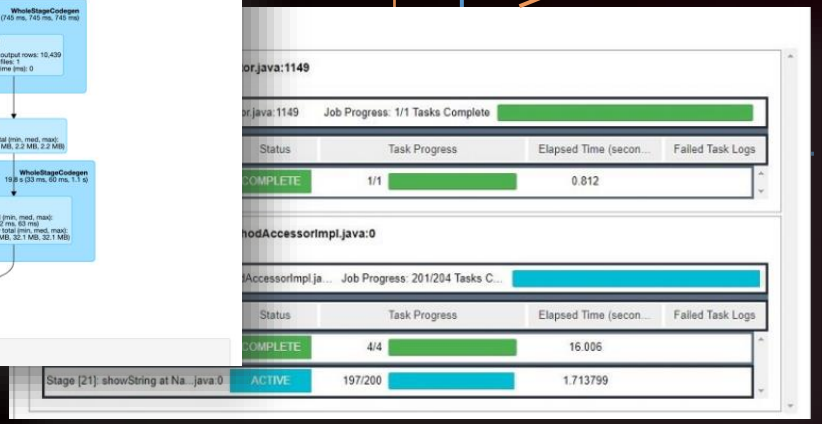
Job A



Workers use Execution Role A

Real-time Spark and Hive Tez UI for live Jobs





Spark history server (and Persistent Tez) UI for detailed analysis for completed Jobs

Monitor Amazon EMR Serverless with CloudWatch

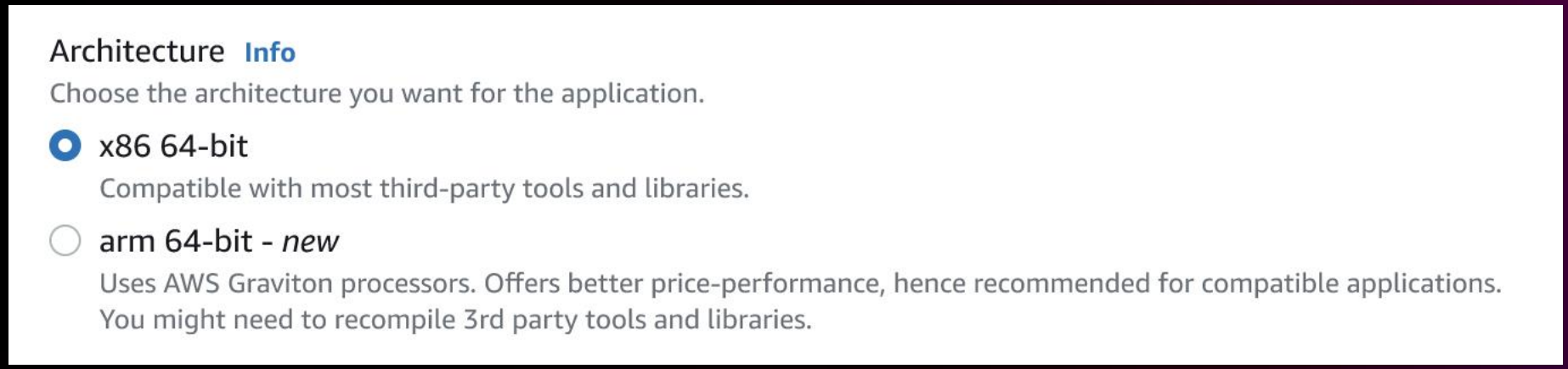
Real-time application capacity usage metrics

Job runs metrics

Git repo for dashboard : https://github.com/aws-samples/emr-serverless-samples/tree/main/cloudformation/emr-serverless-cloudwatch-dashboard

NEW!

Amazon EMR Serverless supports Graviton2



- **Up to 15%** improvement in performance
- **20%** lower cost
- **Up to 35%** better price-performance

NEW!

Amazon EMR Serverless

ALL THE BENEFITS OF EMR – WITHOUT MANAGING CLUSTERS

AWS Graviton2 Support

Application capacity and job metrics in Amazon Cloudwatch

Live debugging of jobs

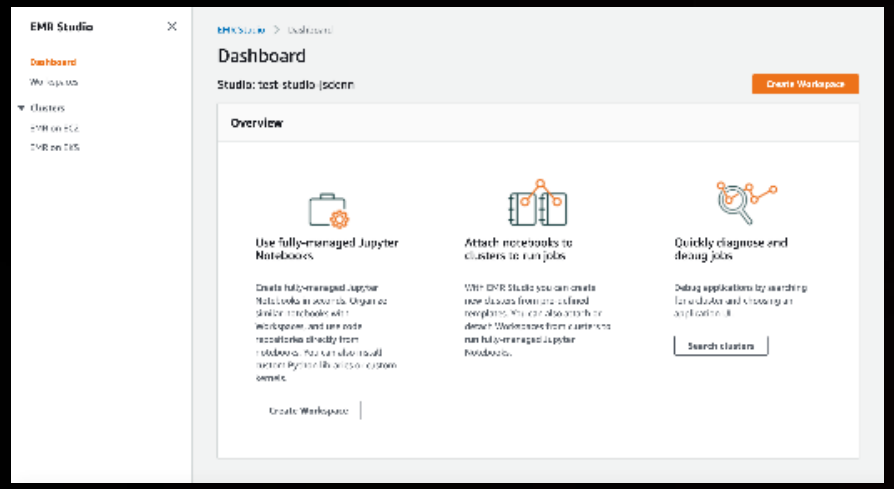
Fast Fine-Grained Auto Scaling

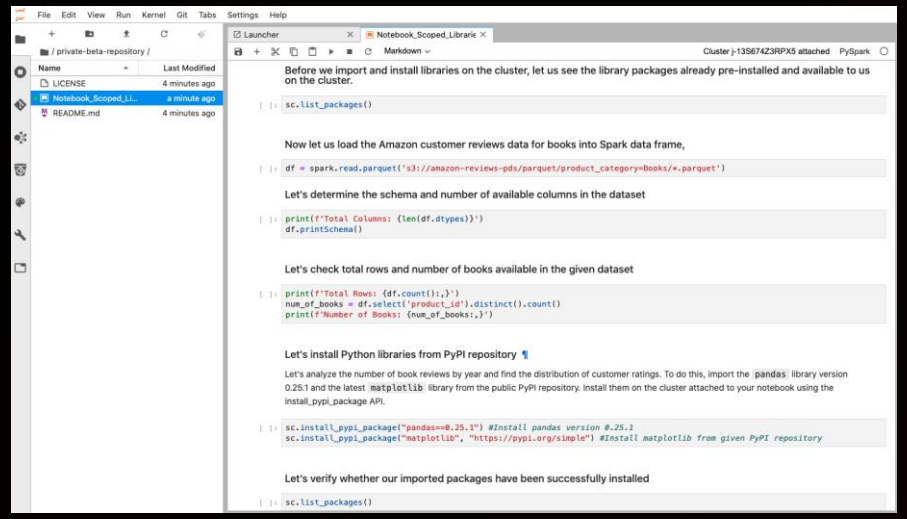
Auto stop
Auto start

Interactive notebooks

EMR Studio

FULLY MANAGED IDE FOR INTERACTIVE DATA ANALYTICS: DEVELOP, VISUALIZE, AND DEBUG APPLICATIONS





Single sign-on integration with IdP

Fully-managed Jupyter Notebooks

Integrated with Git Repositories

Simplified debugging with Spark UI and YARN UI

Browse, create, or delete EMR clusters

Run Notebooks in workflows using APIs

Run interactive data analysis using EMR on EKS clusters

NEW!

EMR Studio features

FULLY MANAGED IDE FOR INTERACTIVE DATA ANALYTICS: DEVELOP, VISUALIZE, AND DEBUG APPLICATIONS

IAM authentication and federation support

Multi-language support (R, PySpark, Scala, SQL)

Auto-terminate idle clusters

Real-time co-authoring of notebooks

SQL explorer with Presto support

Mount workspace directories to EMR clusters

Latest JupyterLab, JEG, Livy, SparkMagic

Security

Security pillars

Isolation

Authentication

Authorization

Encryption

Audit

- VPC
- Private subnets
- Security groups

- LDAP
- Kerberos
- AWS IAM Identity Center (EMR Studio)
- AWS IAM (EMR Studio)

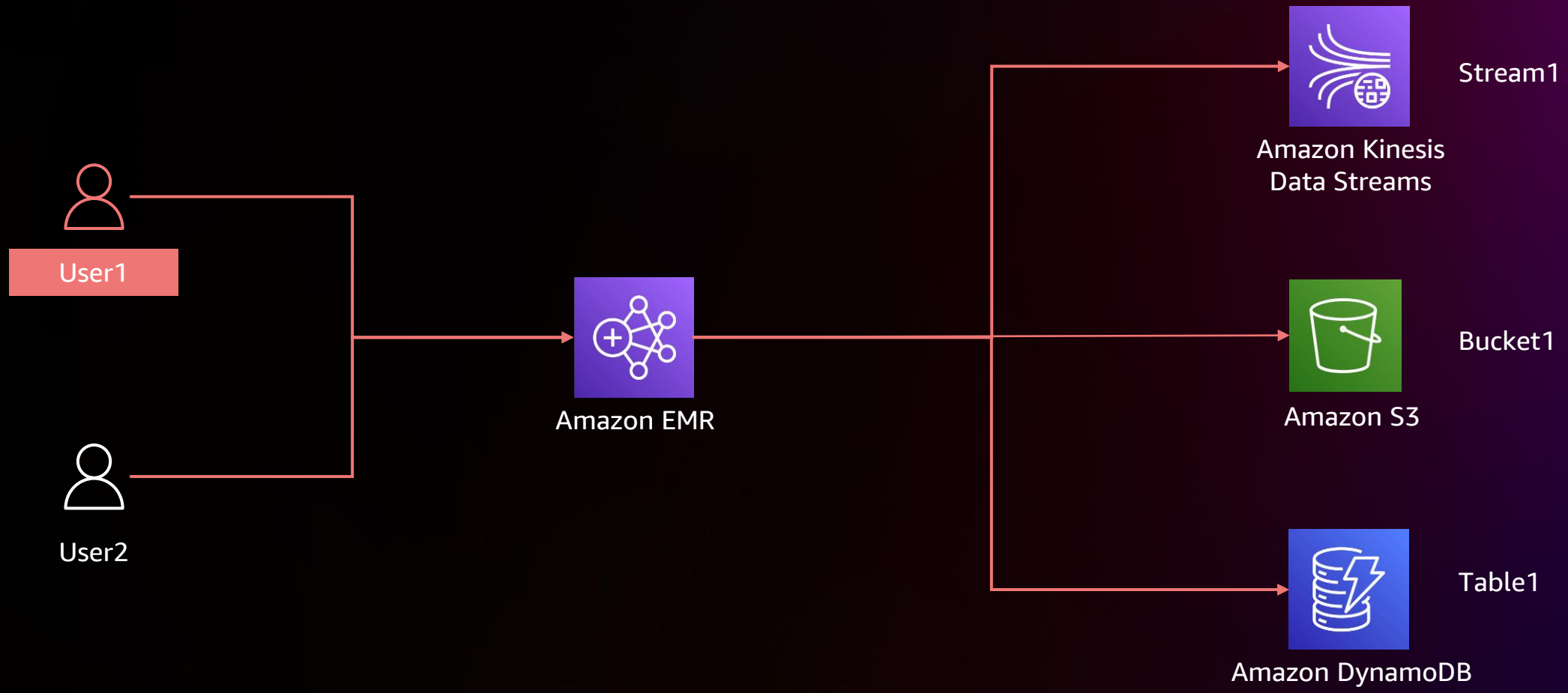
- Cluster IAM Role
- FGAC using Apache Ranger
- FGAC using AWS Lake Formation NEW!
- Job runtime role NEW!

- Encrpytion at rest
- Encrpytion in transit

- Audit using Ranger
- Audit using AWS Lake Formation NEW!

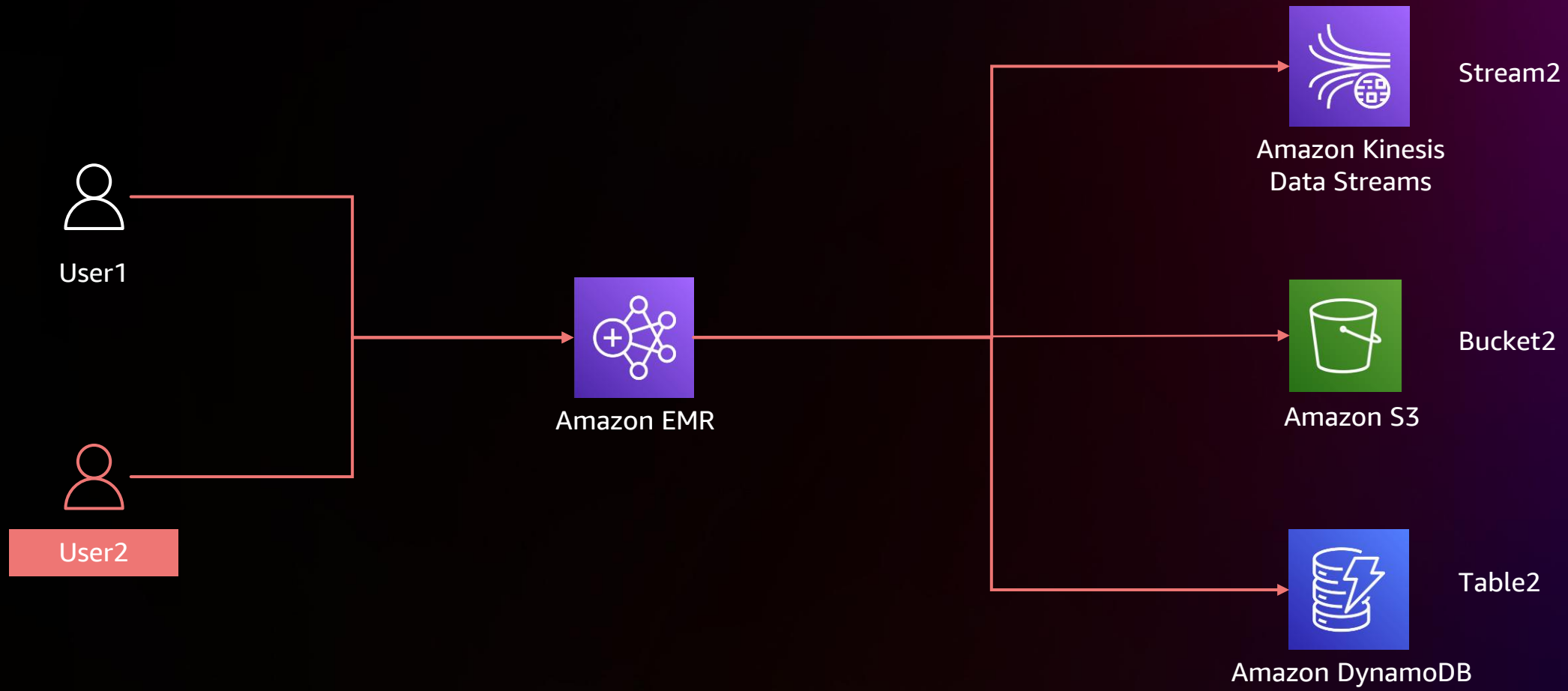
Multi-tenant shared clusters

User execution role: User1 has access to Stream1, Bucket1 and Table1



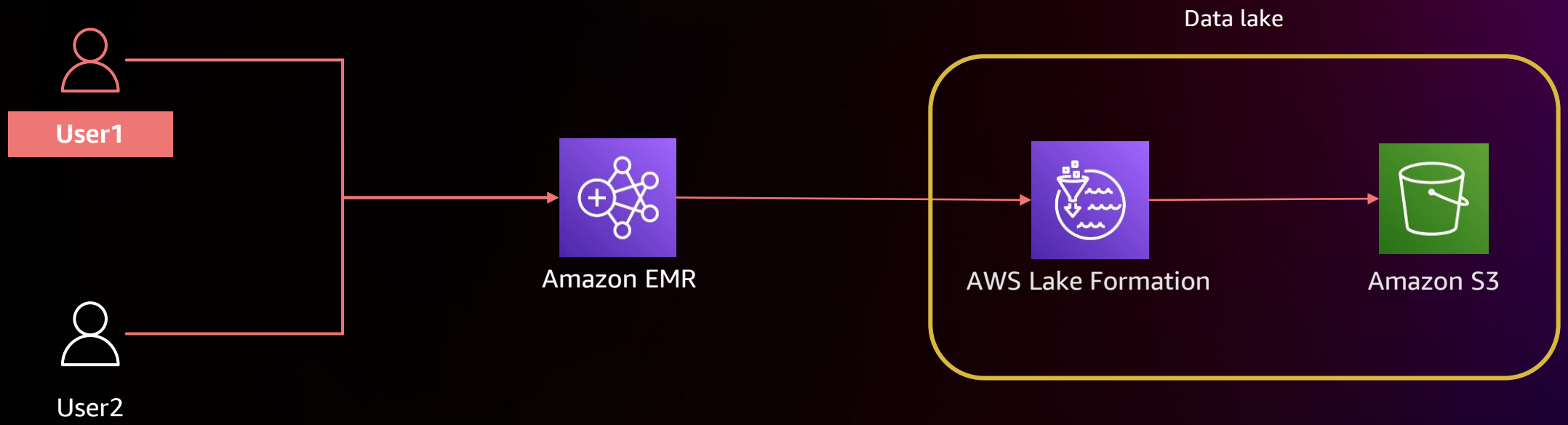
Multi-tenant shared clusters

Job runtime role: User2 has access to Stream2, Bucket2 and Table2



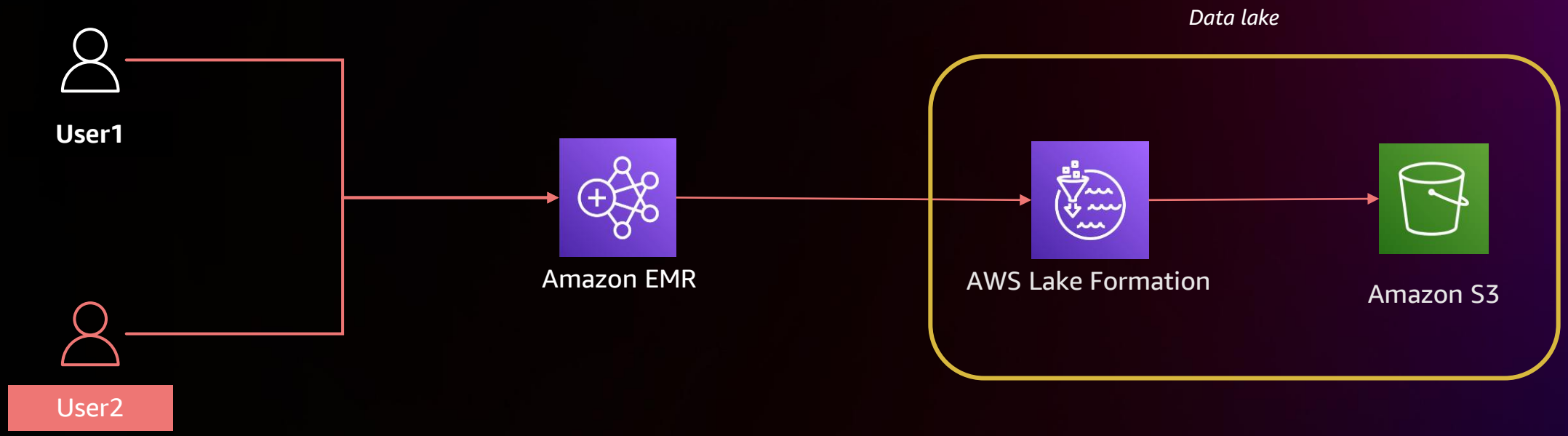
Multi-tenant shared clusters and FGAC

FGAC using AWS Lake Formation: User1 has access to Table1, Columns 1-10

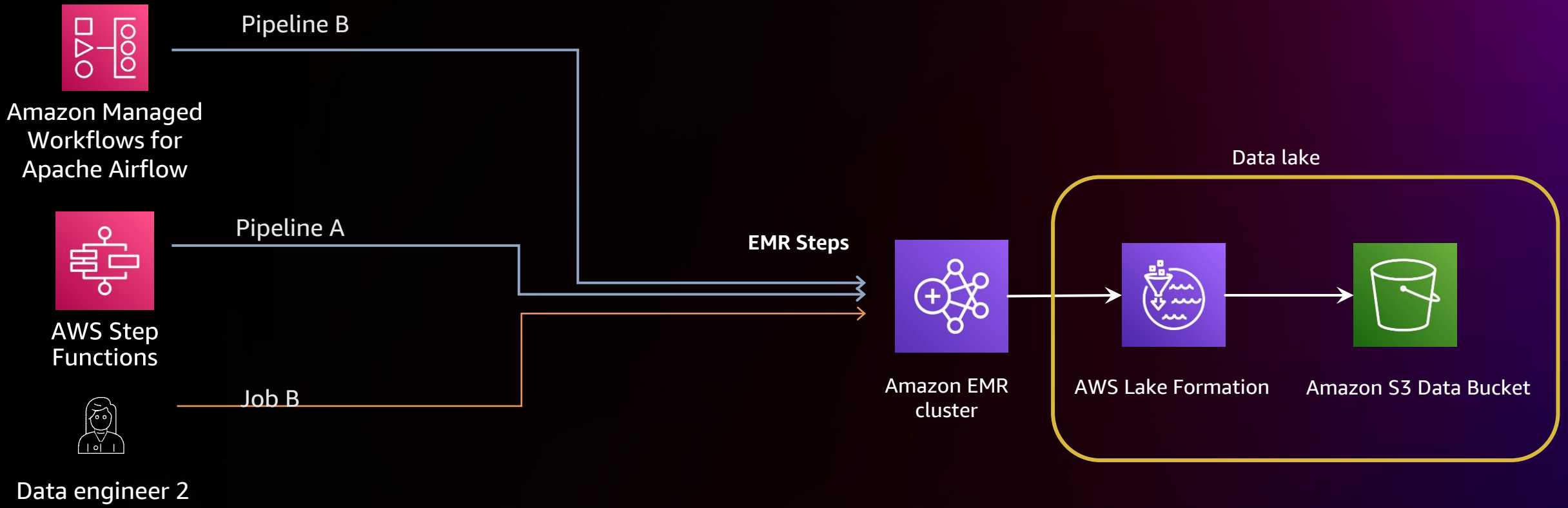


Multi-tenant shared clusters and FGAC

FGAC using AWS Lake Formation: User2 has access to Table1, Columns 5-15



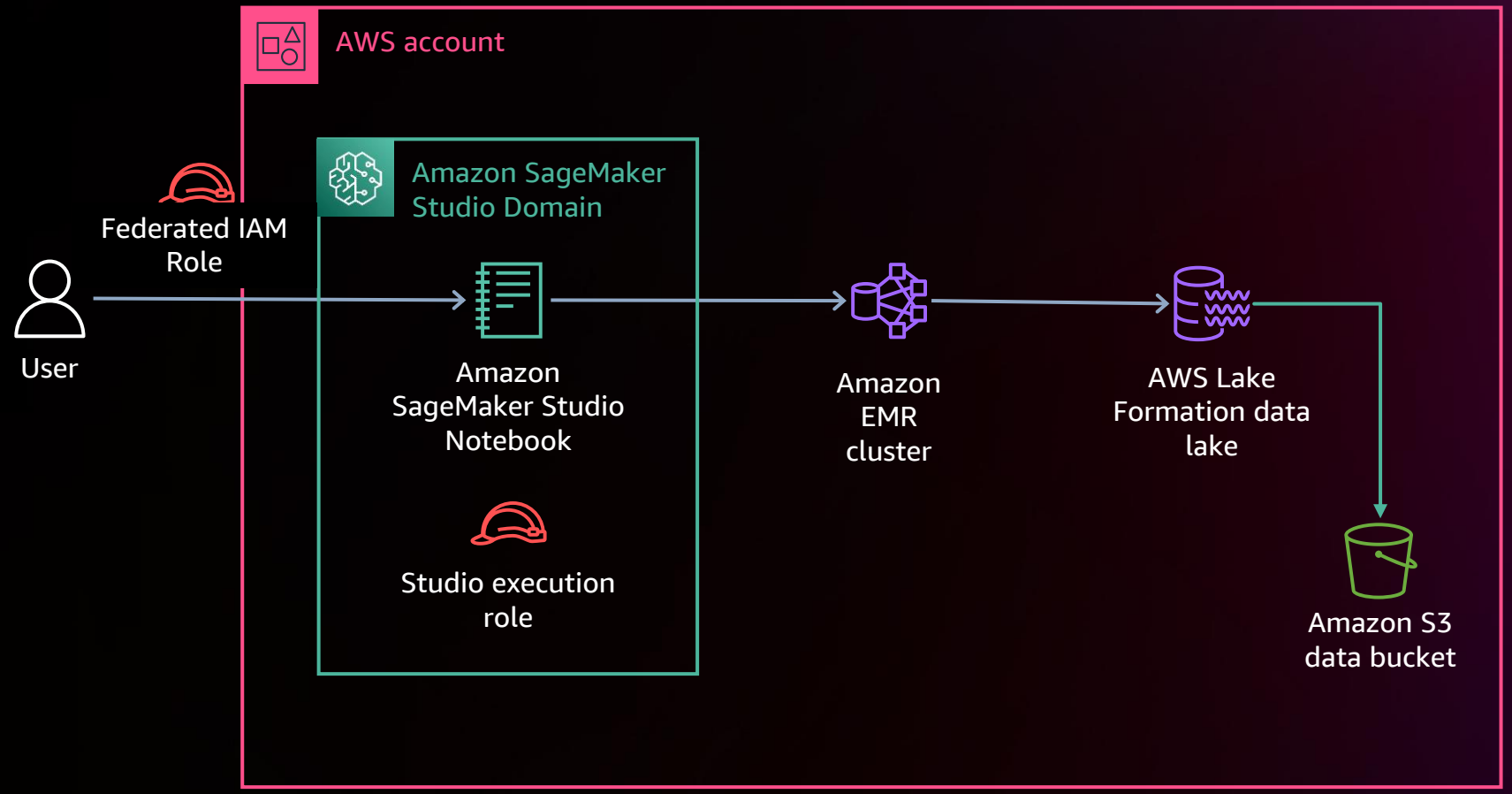
Data access control using runtime roles for Amazon EMR steps



Fine-grained access control for jobs using runtime roles with Amazon EMR steps

NEW!

Data access control from Amazon SageMaker



Fine-grained access control from interactive SageMaker Studio Notebooks

More EMR sessions

- ✓ ANT325 – How Disney+ processes clickstream data on Amazon EMR Serverless
- ✓ ANT303 – Security and data access controls in Amazon EMR
- ✓ ANT309-R1 – Workshop: Build analytics applications using Apache Spark with Amazon EMR Serverless
- ✓ ANT330-R – Run Apache Spark on Kubernetes with Amazon EMR on EKS
- ✓ ANT338 – FINRA – Scaling data processing with Amazon EMR at the speed of market volatility
- ✓ OPN402 – Apache Hudi on Amazon EMR: Tuning for cost and performance

Thank you!

Neil Mukerje
mukerjen@amazon.com

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