AWS re:Invent

ANT308

A Deep dive Into running Apache Spark on Amazon EMR

Mert Hocanin
Principal Big Data Architect
Amazon Web Services

Abhishek Sinha
Principal Product Manager
Amazon Web Services

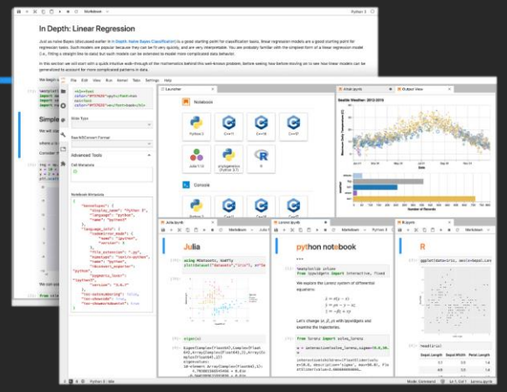
AWS re:Invent

aws

How will we spend our time today

1. Apache Spark Improvements through the year
2. Common architectural patterns
3. Demos

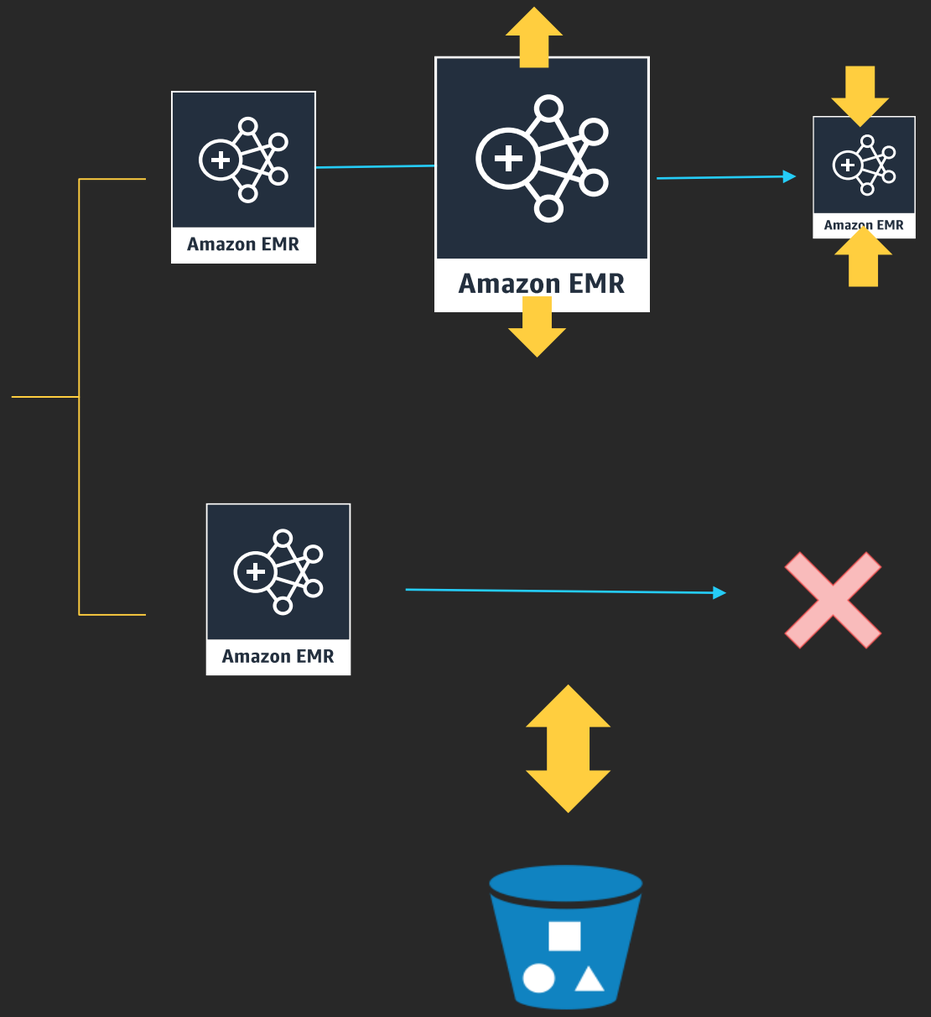
Data Engineering Platforms are evolving



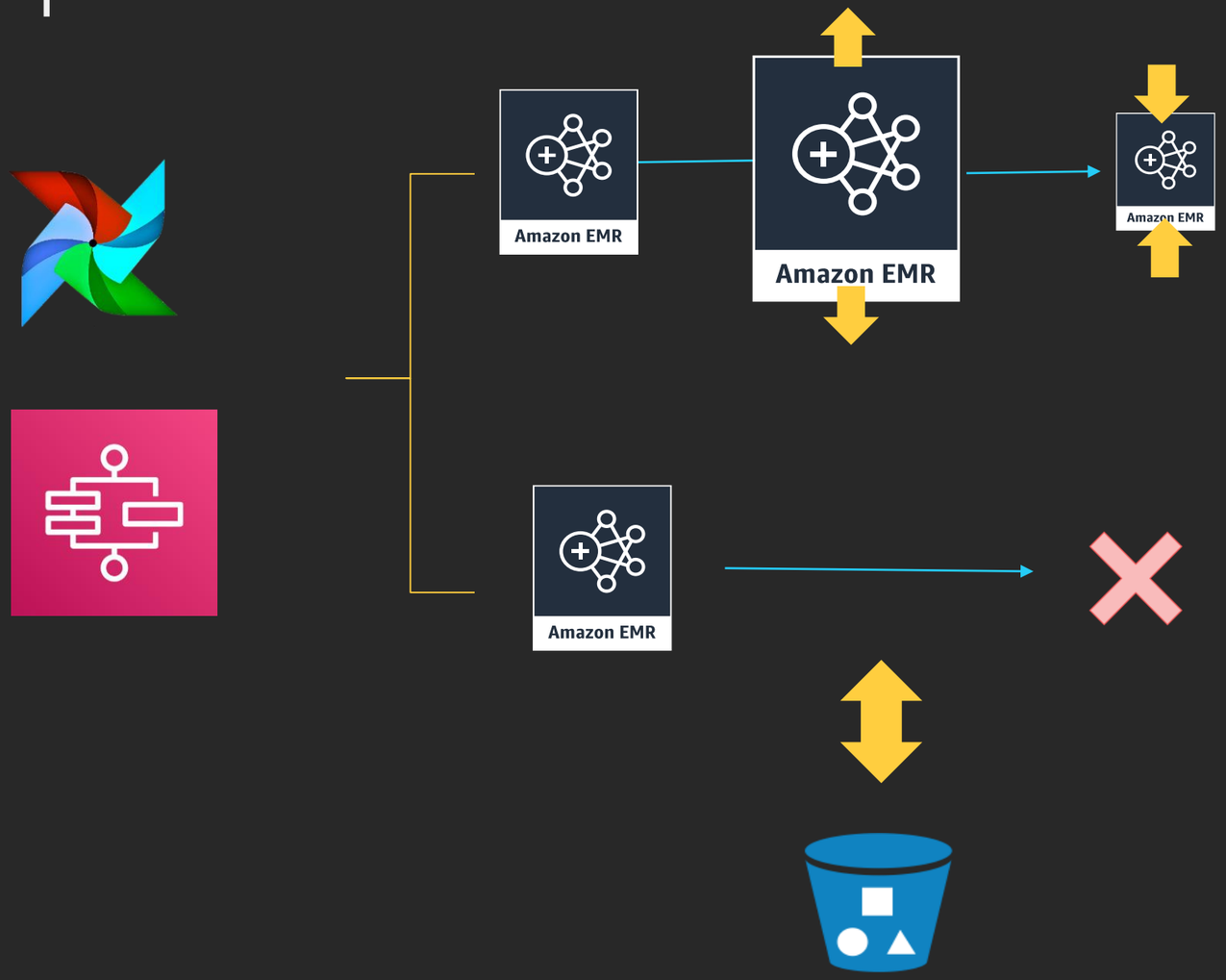
Notebooks

Apache Airflow

AWS Step Functions



Customers have started to adopt cloud-native patterns



Long running & auto scaling cluster

Purpose-built, job-scoped clusters

How do you decide

Long-running and auto scaling

1. Great for lines of business clusters
2. Great for short-running jobs
3. Ideal to save costs for multi-tenanted data science and data engineering jobs

Transient and job scoped

1. Work well for long-running, job-scoped pipelines
2. Separating production pipelines into job-scoped clusters reduces blast radius

A common architectural pattern for data platforms

Clients

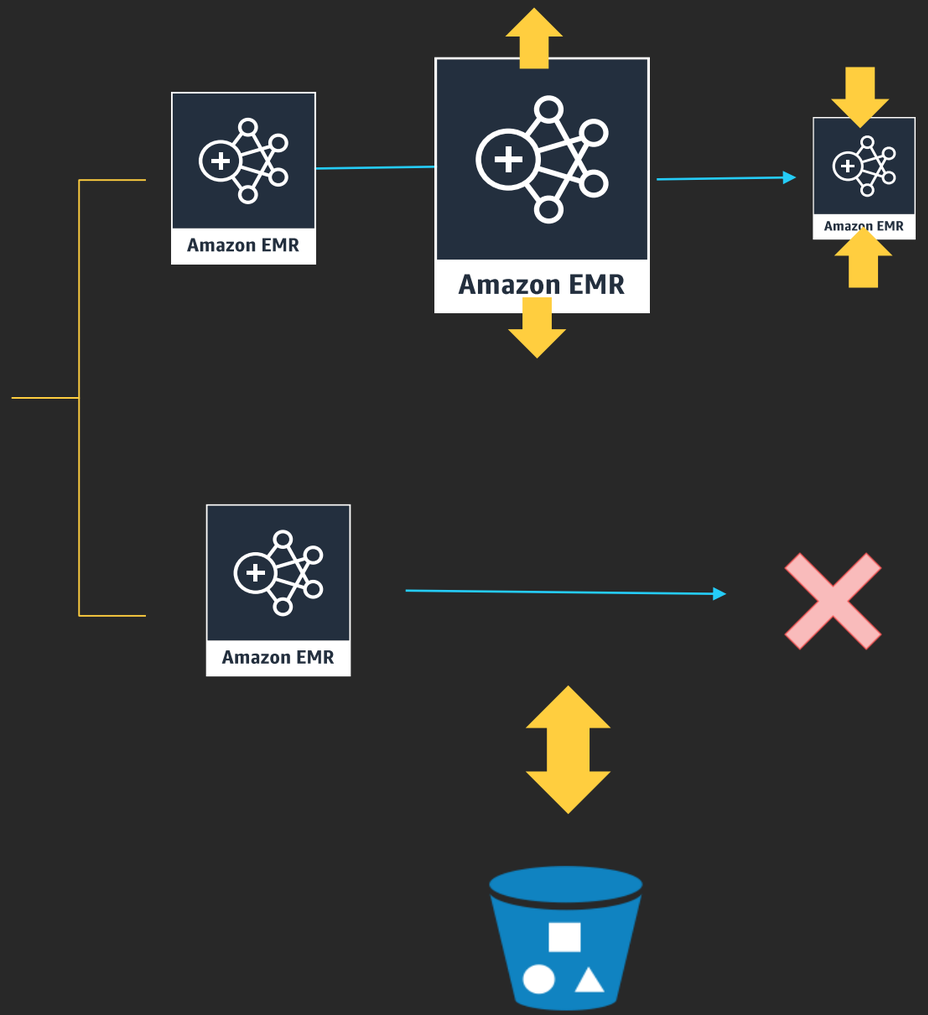
Notebooks

Orchestration

Apache Airflow

AWS Step Functions

Execution



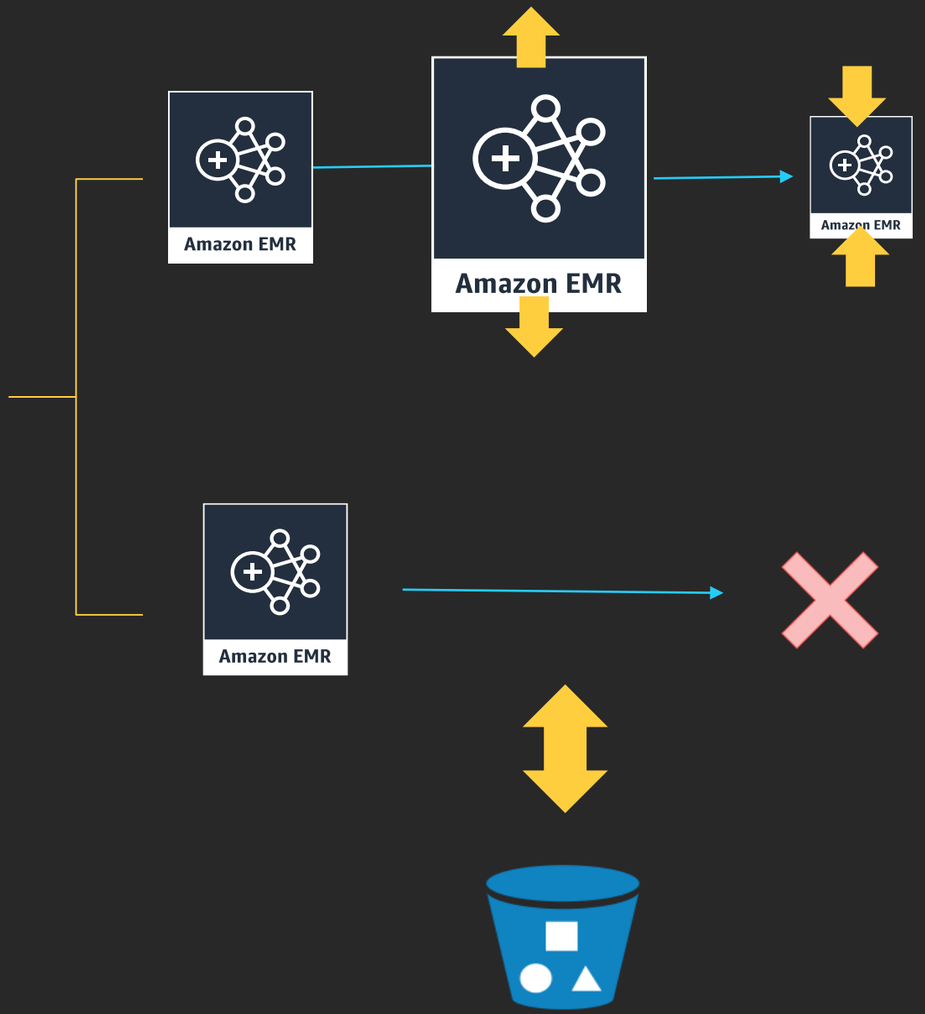
A common architectural pattern for data platforms

Improvements

Notebooks

Apache Airflow

AWS Step Functions



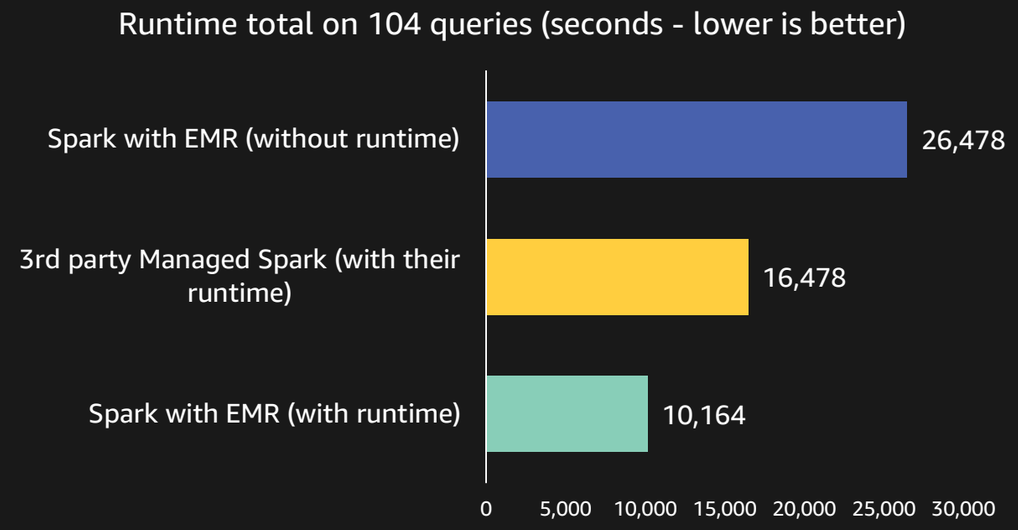
1

EMR Runtime For Apache Spark

Performance improvements in Spark for Amazon EMR

NEW!

Performance-optimized runtime for Apache Spark, 2.6x faster performance at 1/10th the cost



**Based on TPC-DS 3TB Benchmarking running 6 node C4x8 extra large clusters and EMR 5.28, Spark 2.4*

Runtime built on a optimized version of Apache Spark

Best performance

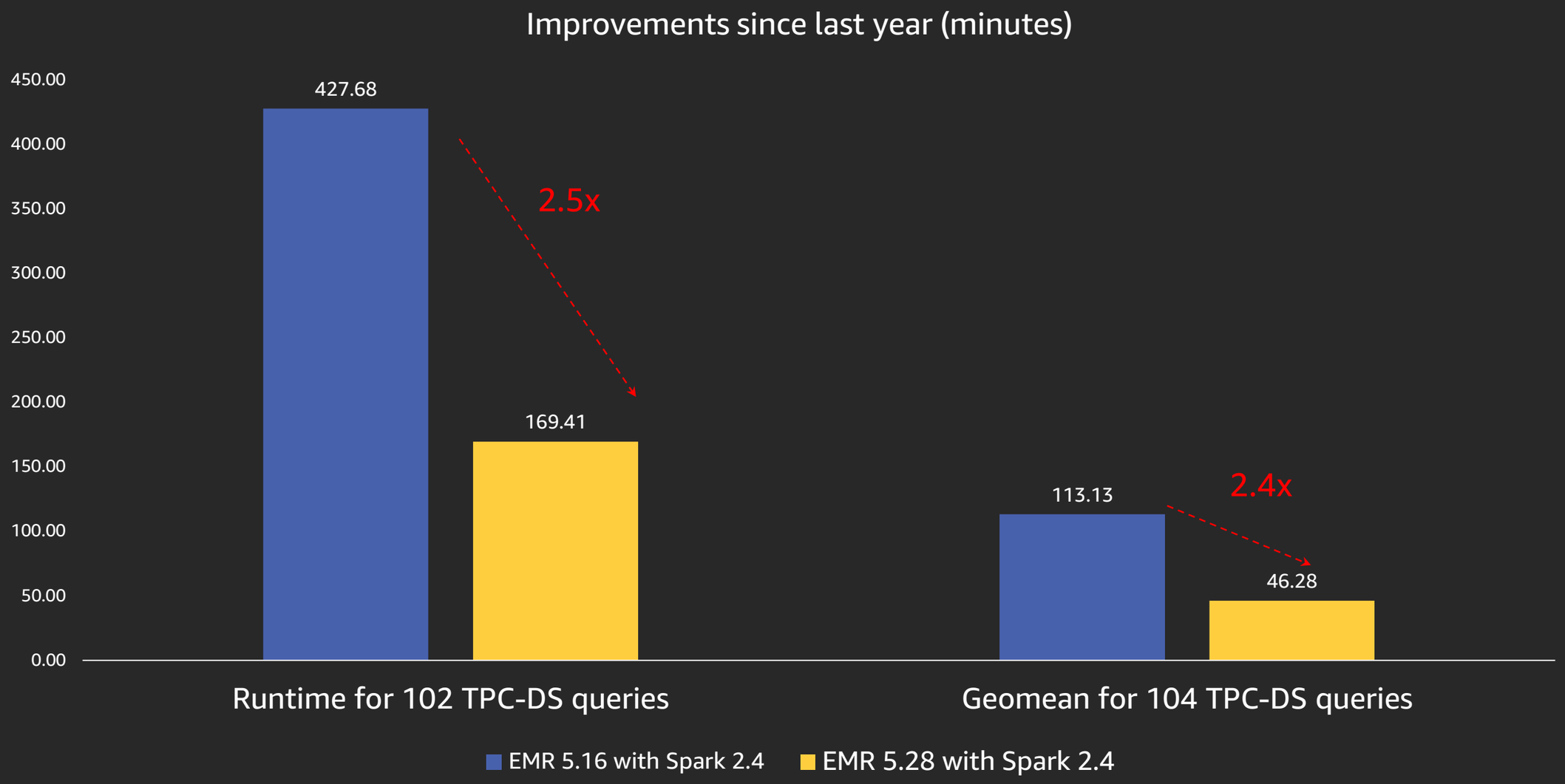
- **2.6x faster** than Spark with EMR without runtime
- **1.6x faster** than 3rd party Managed Spark (with their runtime)

Lowest price

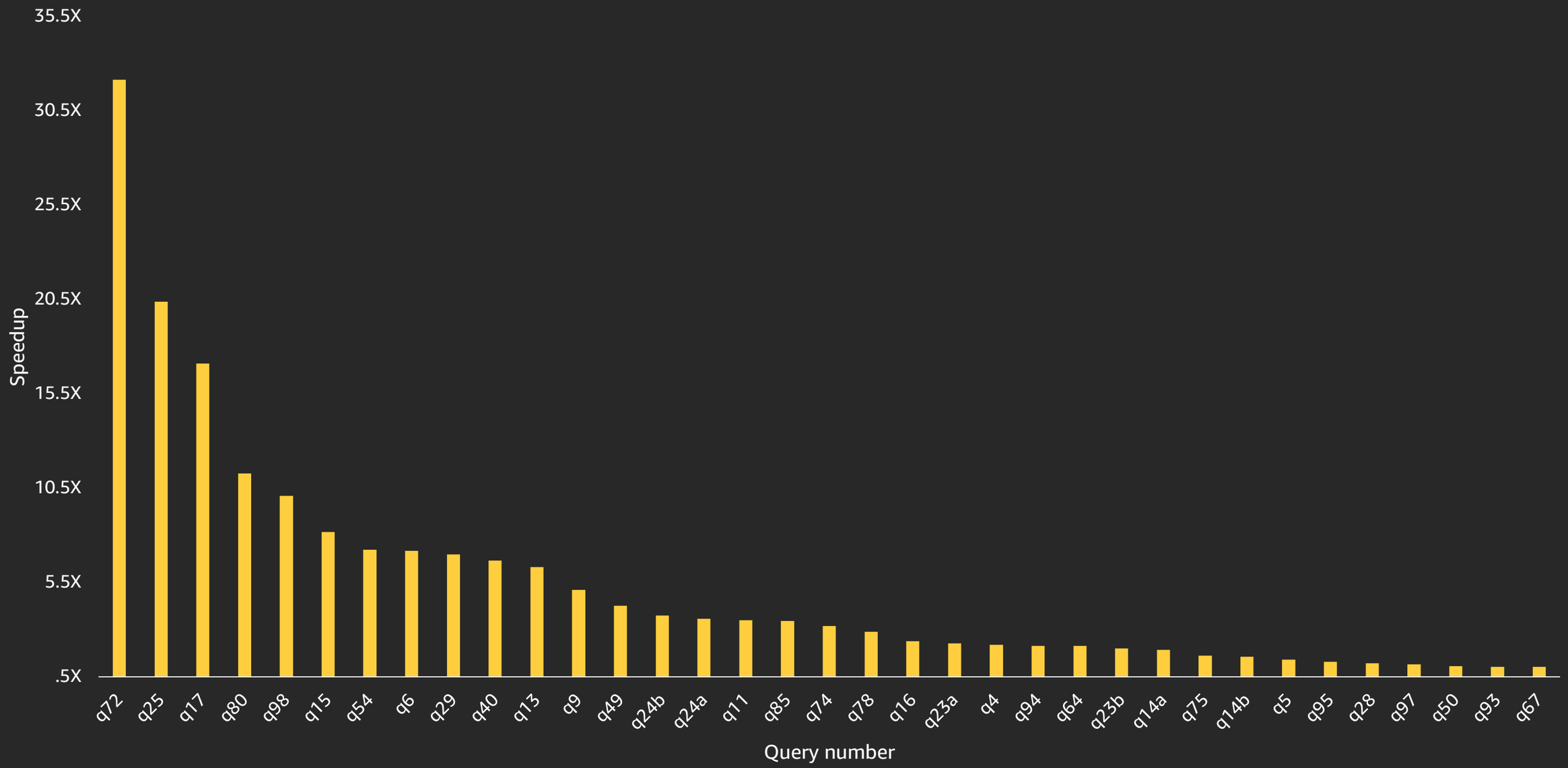
- 1/10th the cost of 3rd party Managed Spark (with their runtime)

100% compliant with Apache Spark API's

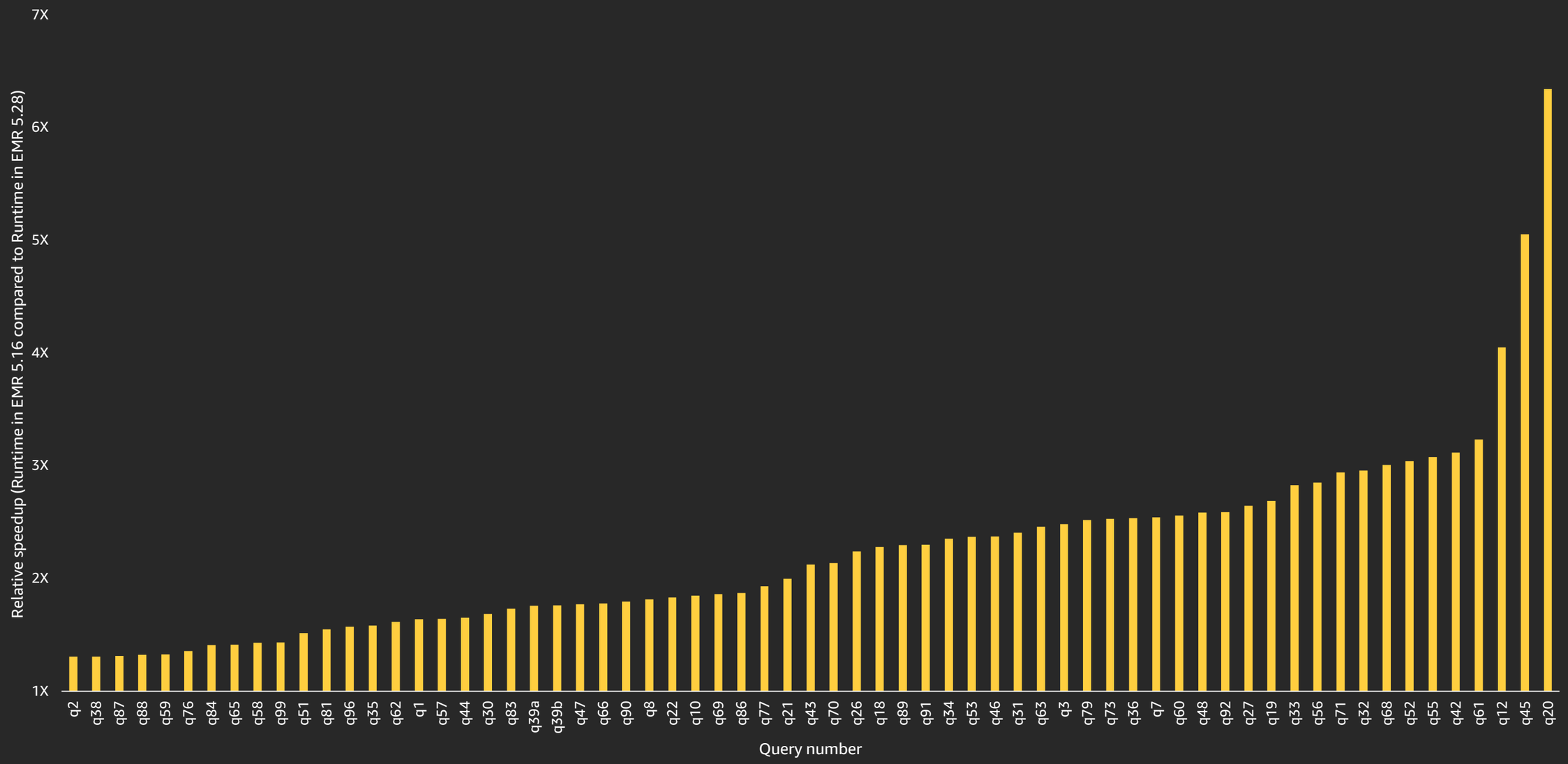
Upgrade to the latest version helps you save costs



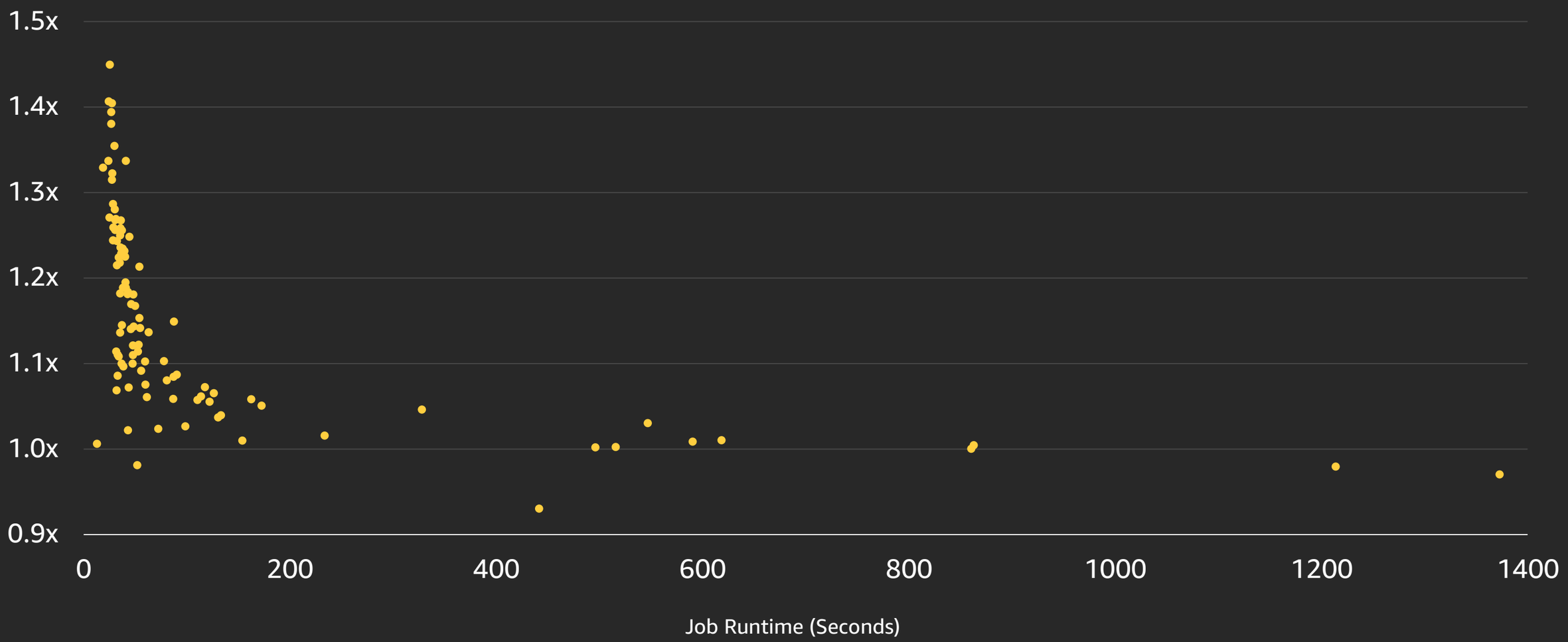
Average of 5x speed up on long-running queries



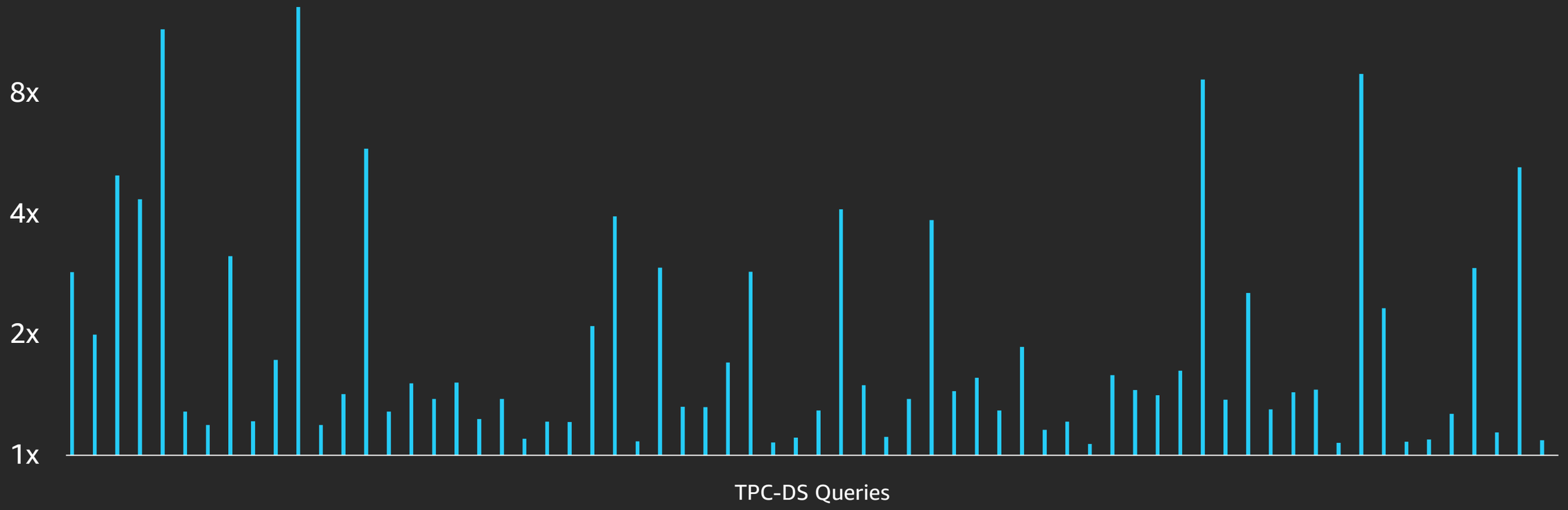
Average of 2x speed up on short-running queries



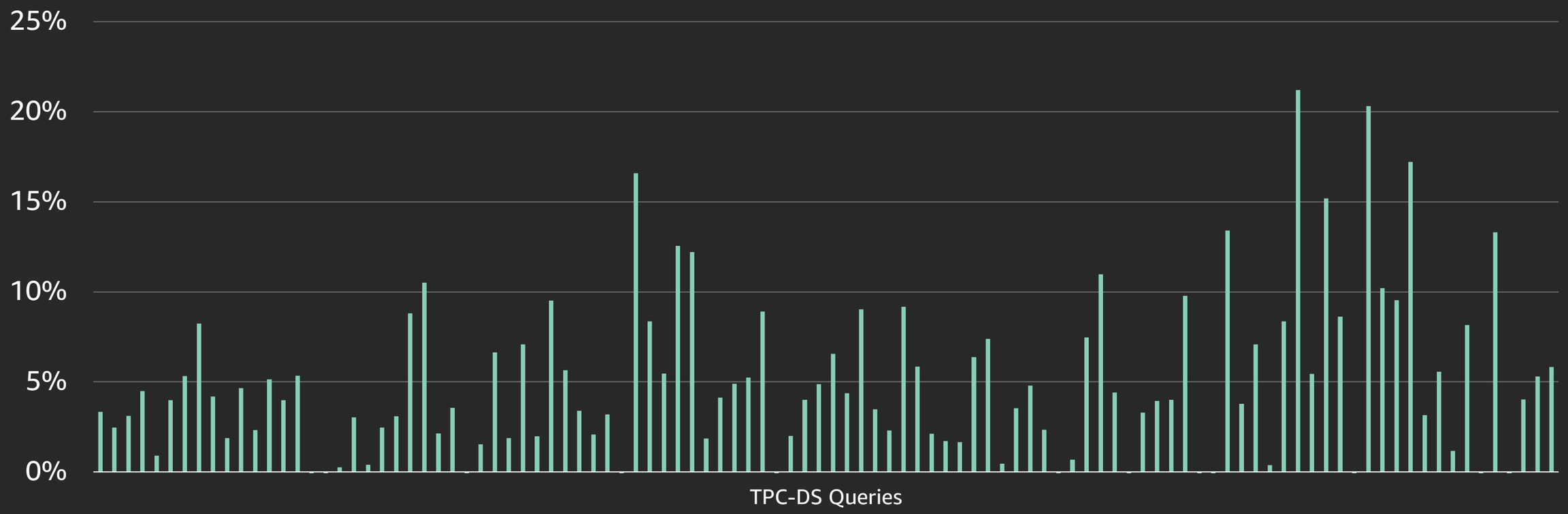
Job startup—eager worker allocation



Planning/optimization—dynamic partition pruning



Query execution—data prefetch



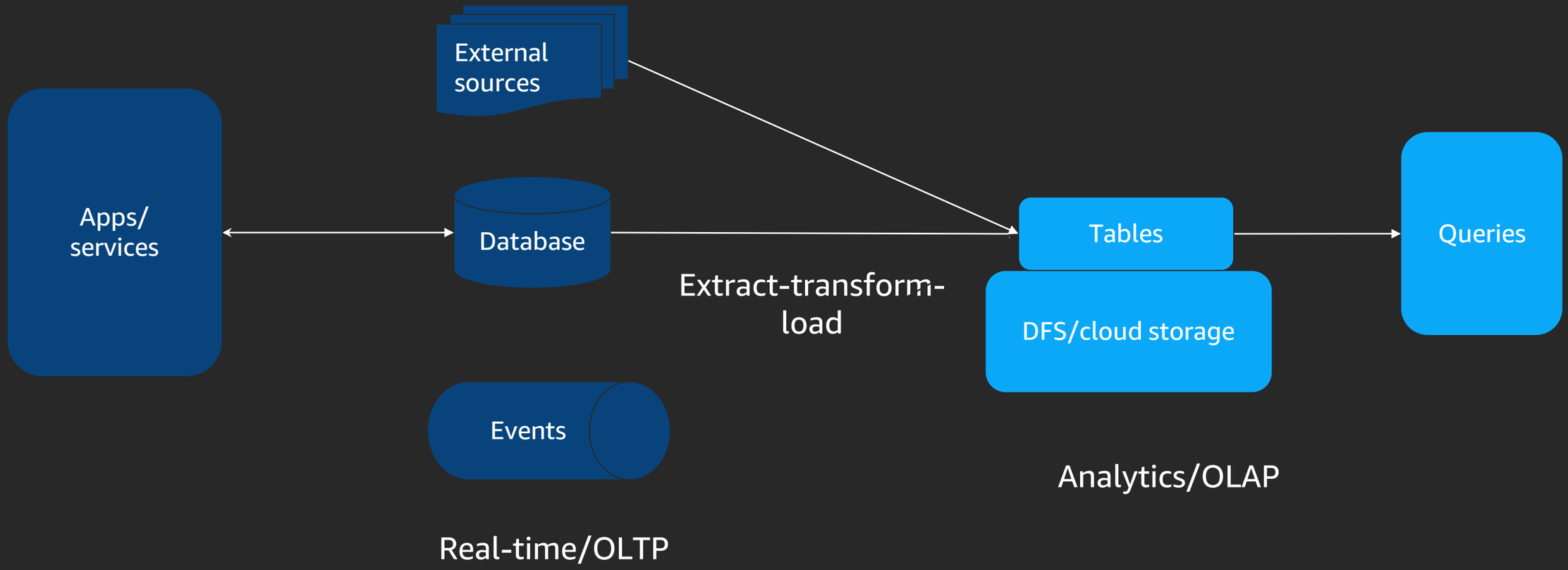
Optimization deep dive

- Configuration
  - CPU/disk ratios, driver/executor conf, heap/GC, native overheads, instance defaults
- Planning/optimization
  - Dynamic partition pruning, join reordering w/o stats and more
- Query execution
  - Data pre-fetch and more
- Job startup
  - Eager executor allocation, and more

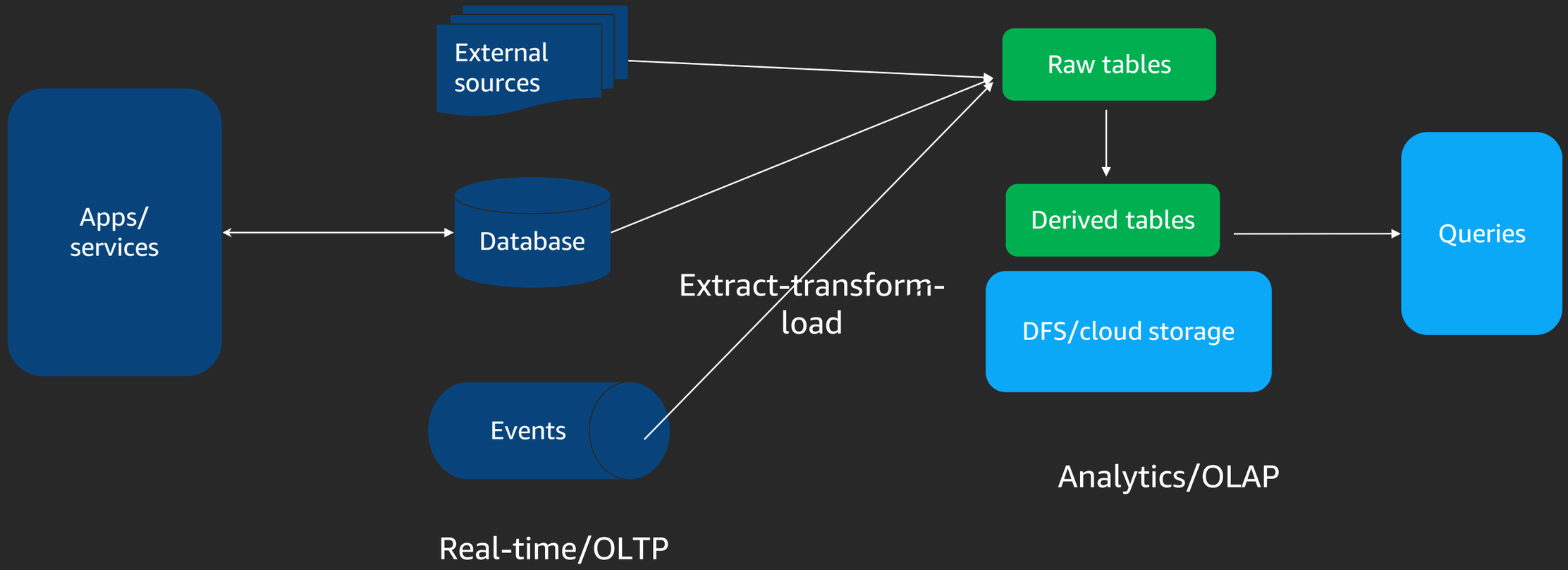
2

Record-level insert, update, and delete with Apache Hudi

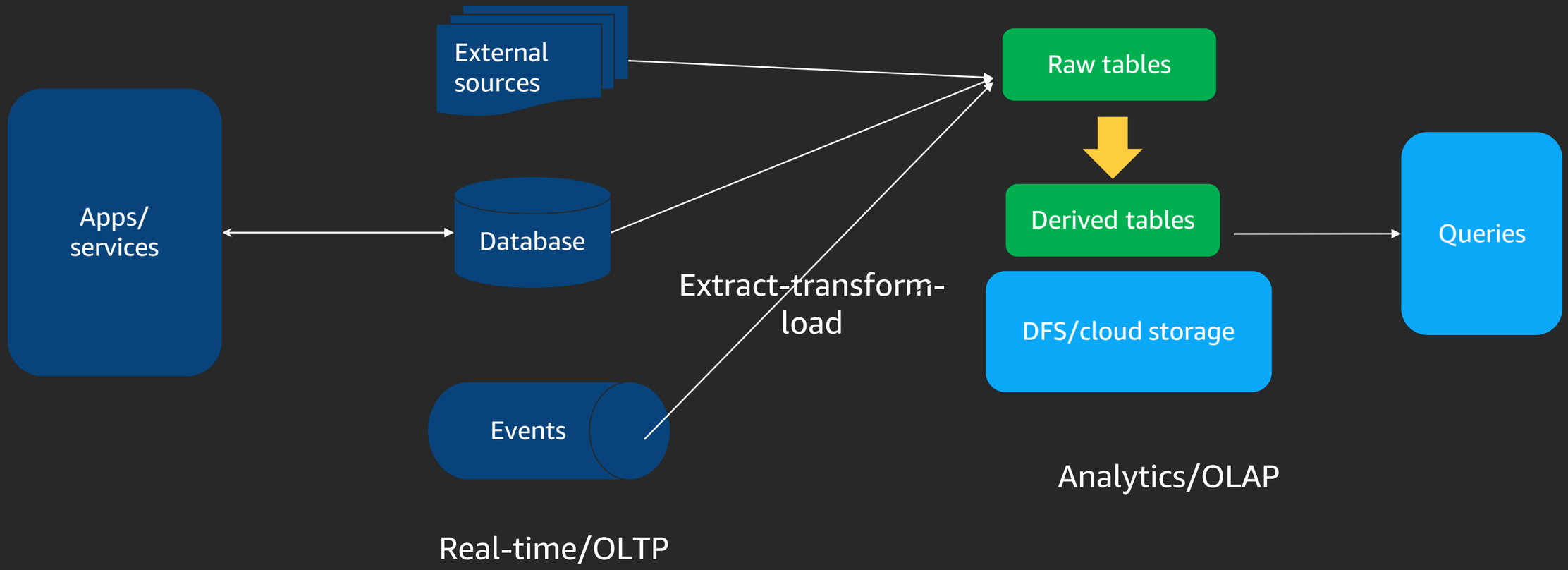
Building a data lake is simple



Typical lakes



Typical lakes



Common use cases are still difficult

Data deletions: Privacy regulation

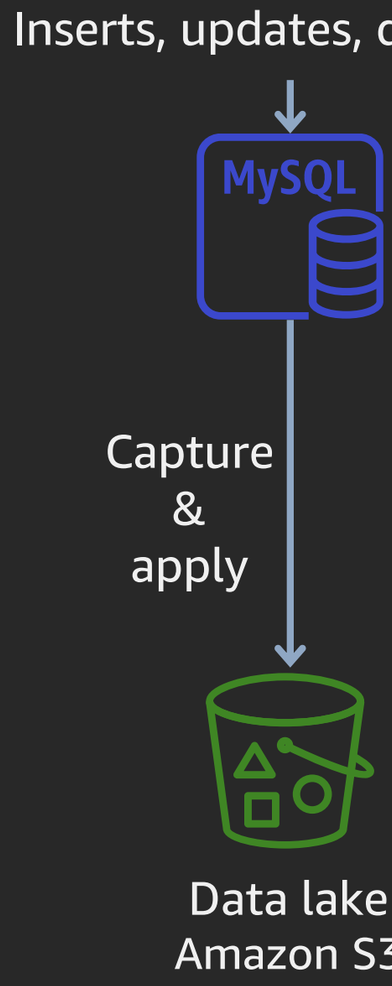
Enforcing data privacy laws

- Delete data within a specific time frame
- Delete data across all data sets
- Identifying files & partitions with specific data
- Lack of indexes on data lake storage

Change Data Capture & apply

Applying change streams to Amazon Simple Storage Service (Amazon S3) data

- Bulk loads don’t scale
- No support for upserts
- Data quality is a serious concern
- Need similar guarantees as a database



Users

Column	Type
userID	int
country	string
last_modified	long
…	…

Streaming data ingestion

High-volume, time-ordered data

- Duplicate events mess up analytics
- Need fast ingestion to Amazon S3
- Schema management, checkpointing

Produce events in Apache Kafka

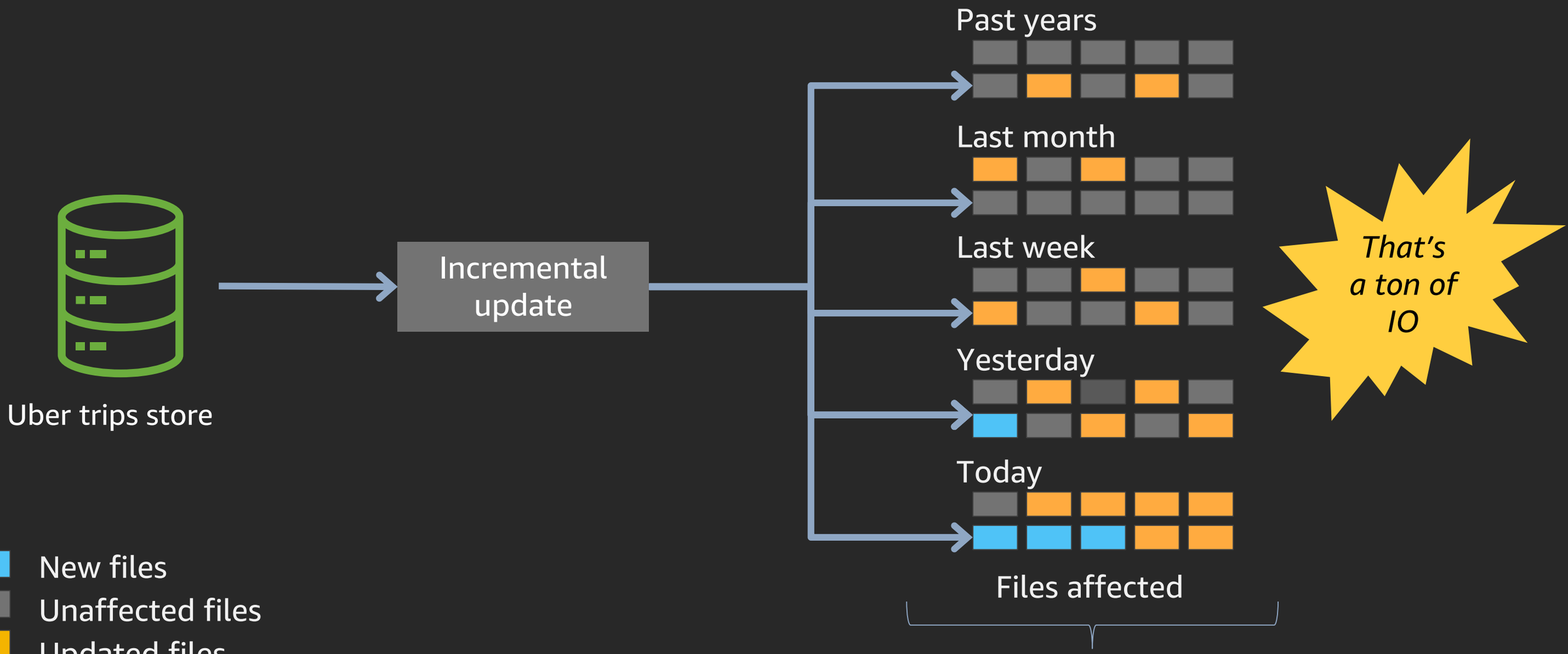
Amazon S3

Impressions

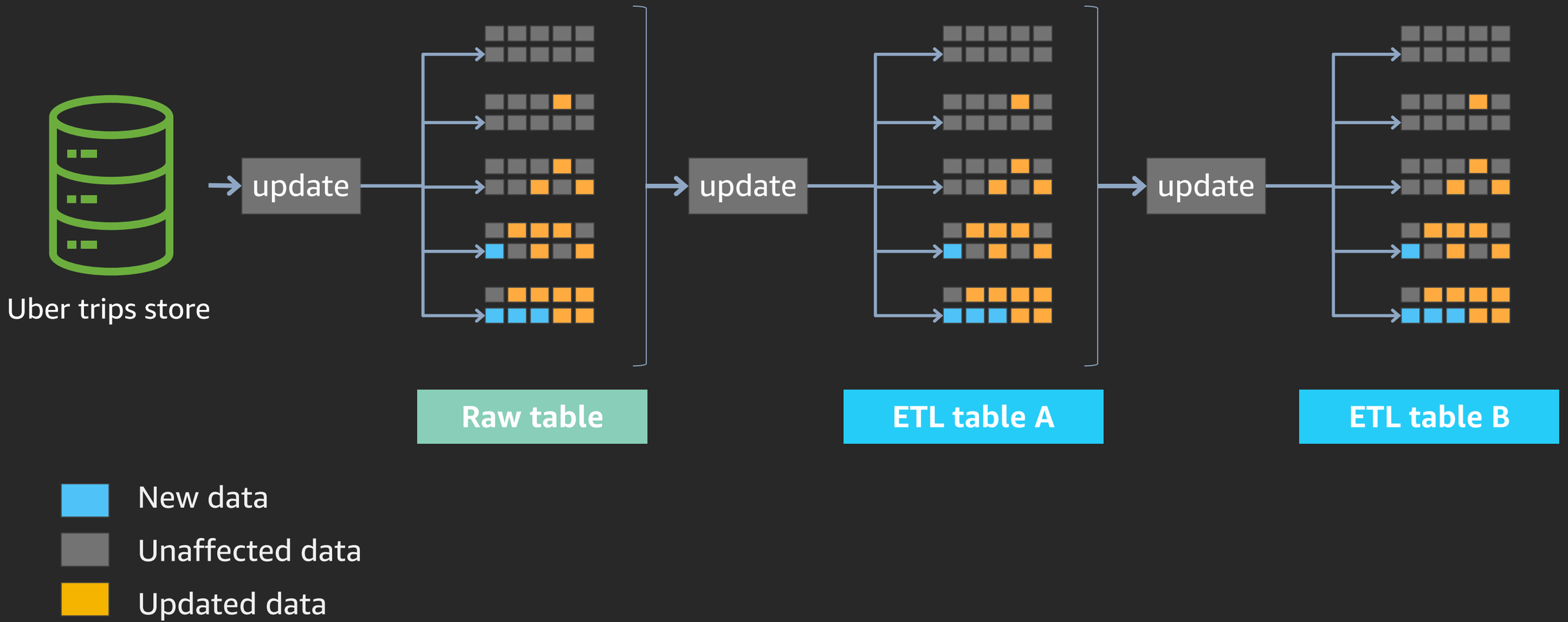
Field Name	Type
event_id	string
datestr	string
time	long

Uber’s Motivating Use Case

Motivating use case



Cascading effects

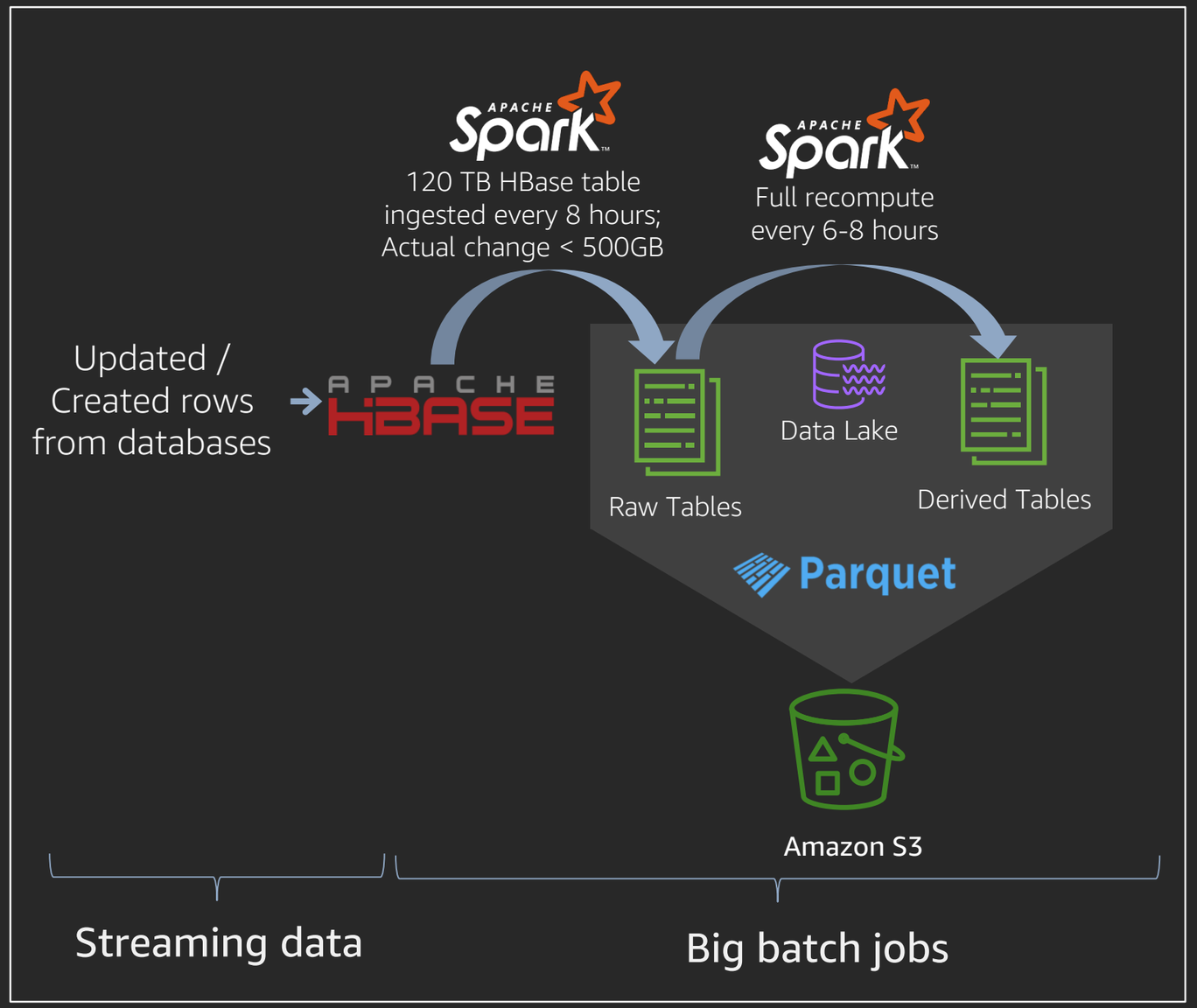


Slow data lake

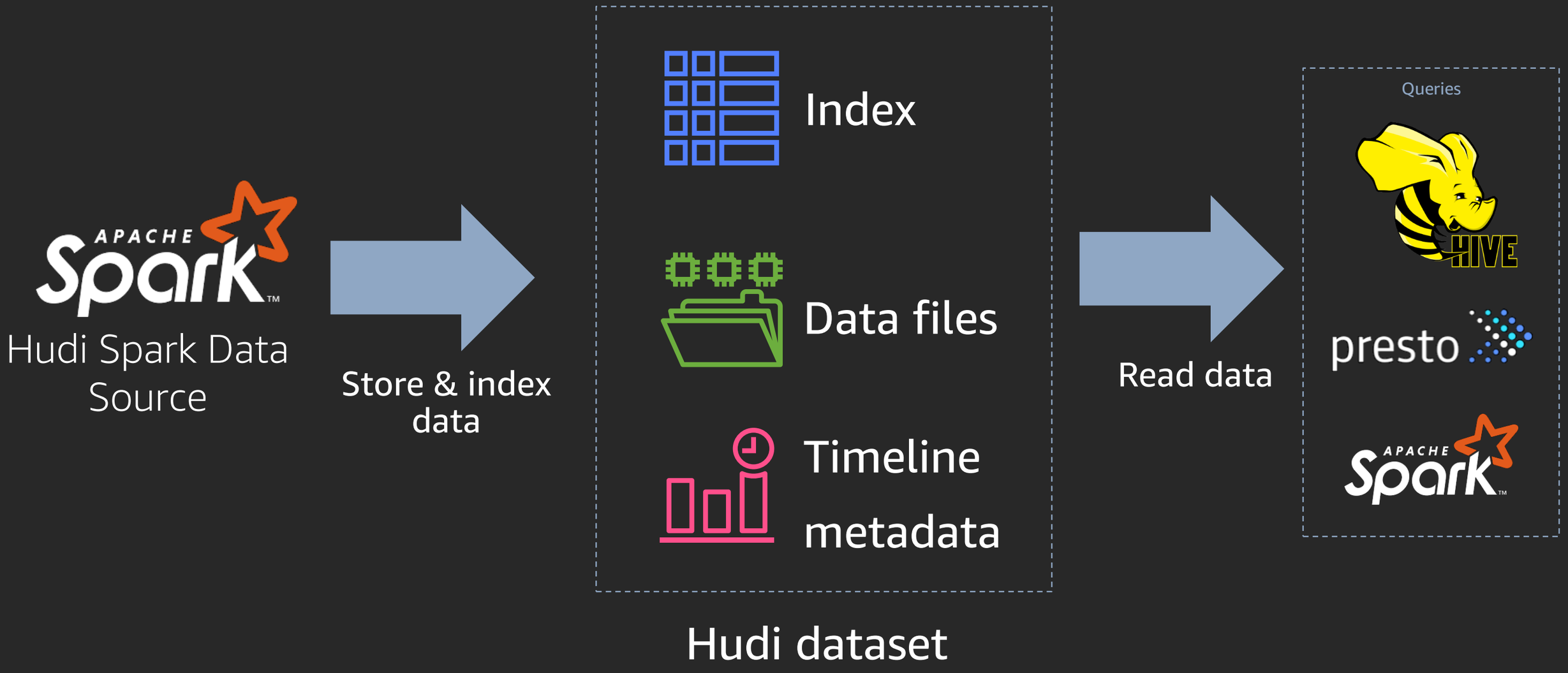
120 TB Apache Hbase table ingested every

8 hours to account for less than 500GB of

data



Introducing Apache Hudi (incubating)



Why You Should Care

- Near real-time data ingestion
- Batch jobs on steroids
- Unified, optimized analytical storage
- Row-level deletions to simplify data privacy
- Building block for great data lakes
- All open source, open formats

Primitives supported

- upsert() support with fast, pluggable indexing
- Atomic publish with rollback, save points
- Snapshot isolation between writer & queries
- Manages file sizes, layout using statistics
- Async compaction of row & columnar data
- Timeline metadata to track lineage

Storage types

Copy On Write

Read heavy

Hudi
Dataset

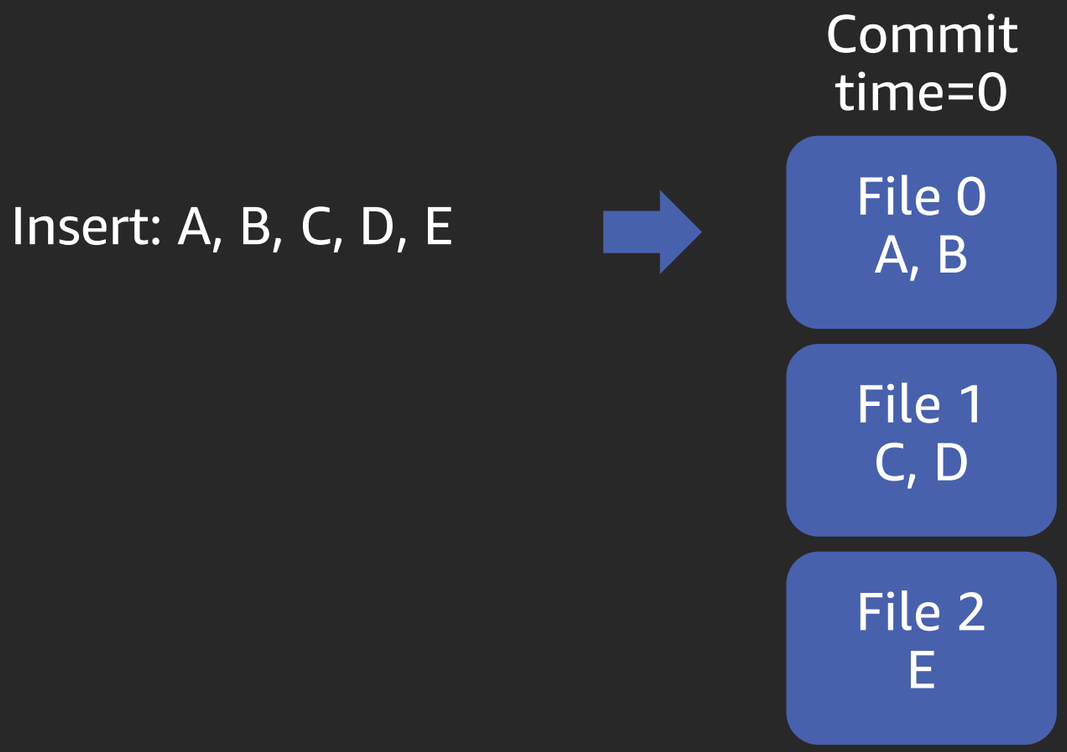
Merge On Read

Write heavy

Storage types & Views

Storage Type: Copy On Write

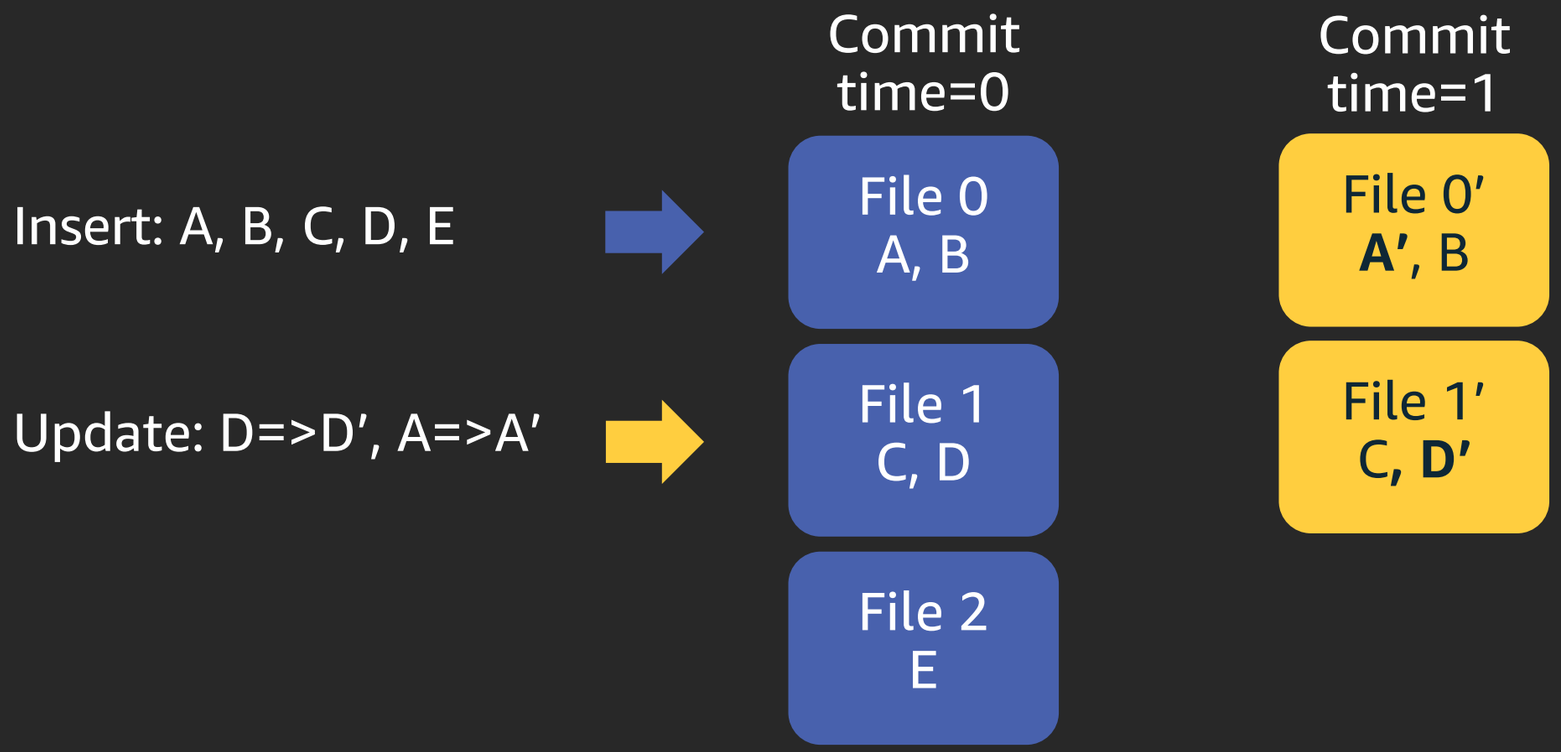
Views/Queries: Read-Optimized, Incremental



Storage types & Views

Storage Type: Copy On Write

Views/Queries: Read-Optimized, Incremental

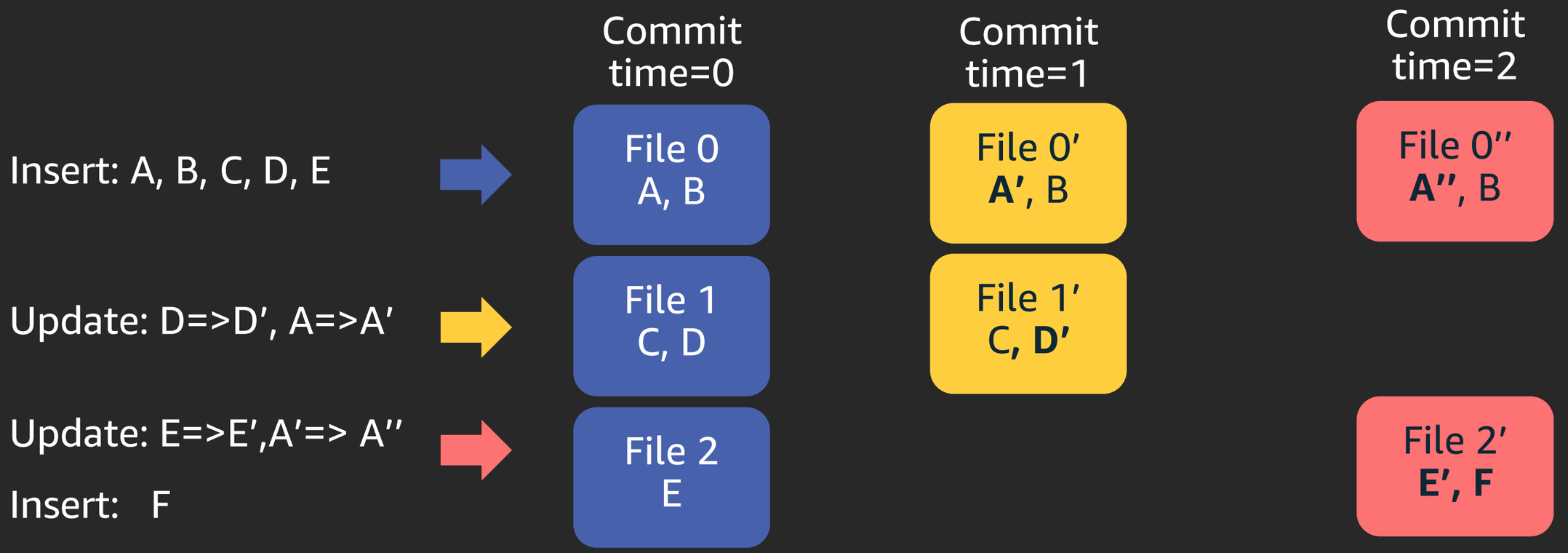


Read-Optimized	A,B,C,D,E	A',B,C,D',E
Incremental	A,B,C,D,E	A',D'

Storage types & Views

Storage Type: Copy On Write

Views/Queries: Read-Optimized, Incremental



Storage types & Views

Storage type: Copy On Write

Views: Read Optimized, Incremental

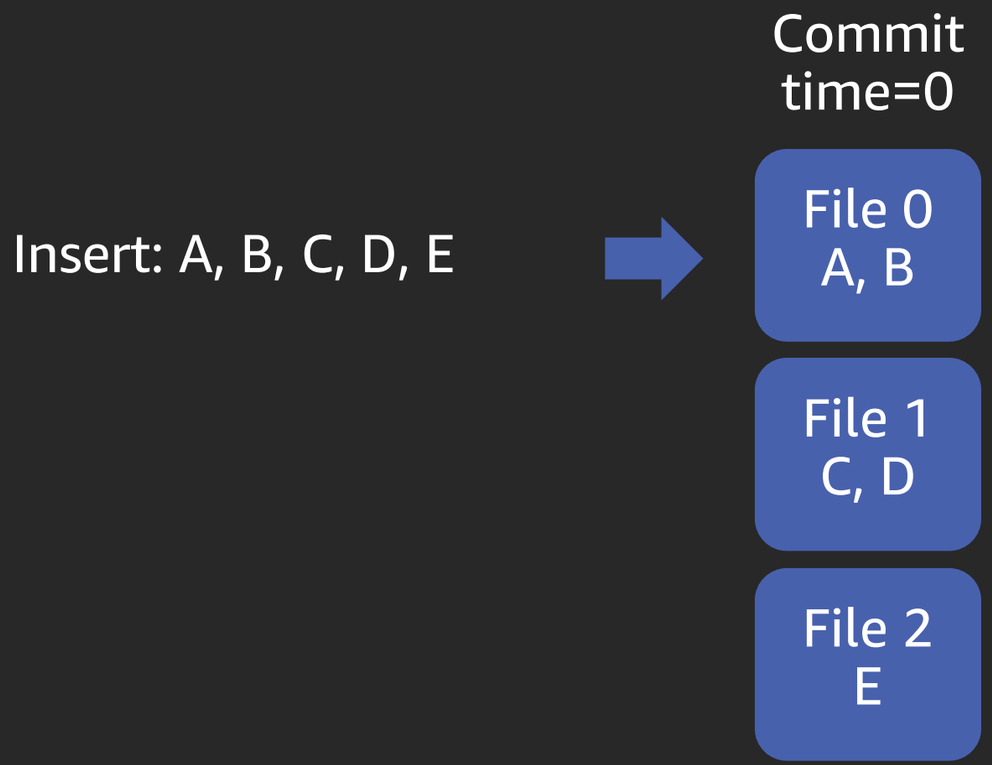
When to Use

- Your current job is rewriting entire table/partition to deal with updates
- Your workload is fairly well understood and does not have sudden bursts
- You’re already using Parquet files for your tables
- You want to keep things operationally simple

Storage types & Views

Storage type: Merge On Read

Views/Queries: Read Optimized, Incremental, Real Time

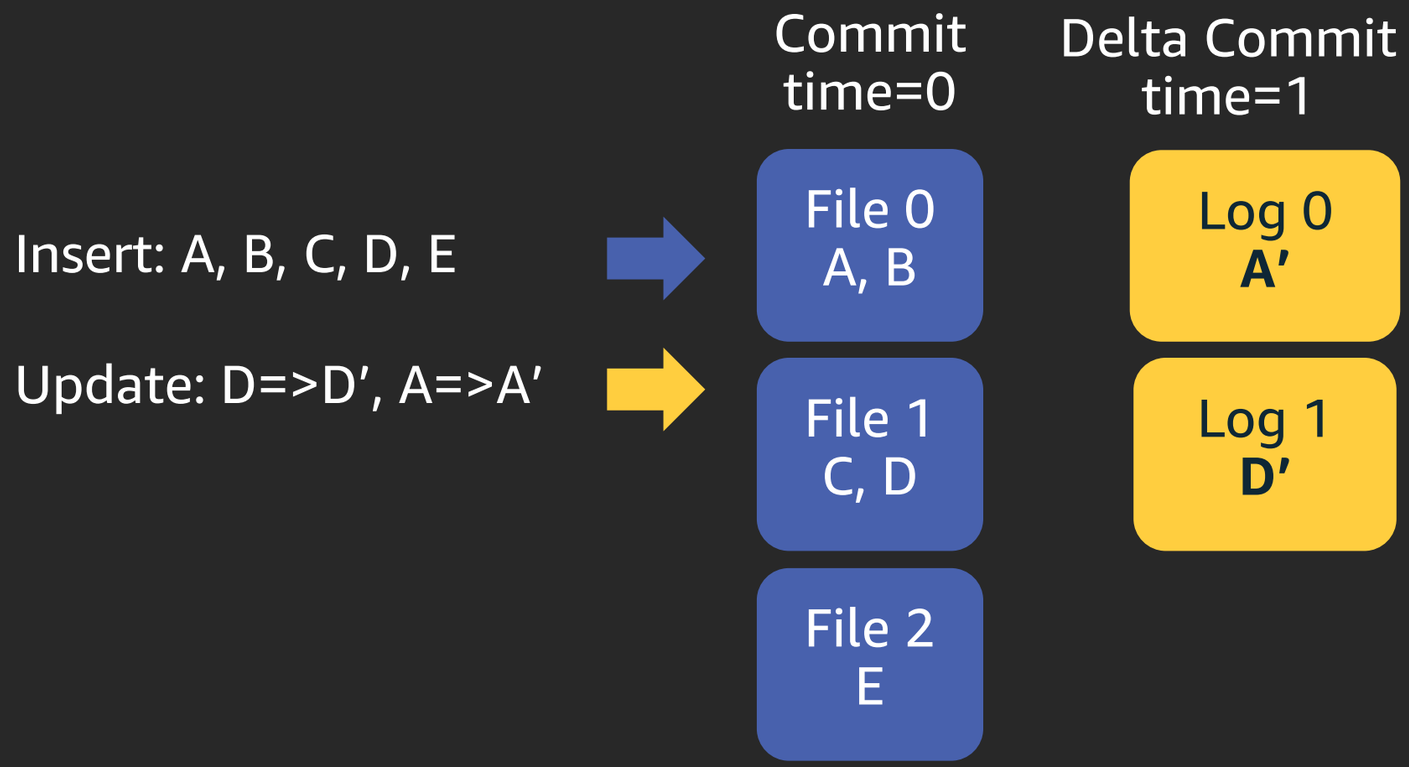


Real time	A,B,C,D,E
Incremental	A,B,C,D,E
Read-Optimized	A,B,C,D,E

Storage types & Views

Storage type: Merge On Read

Views/Queries: Read Optimized, Incremental, Real time

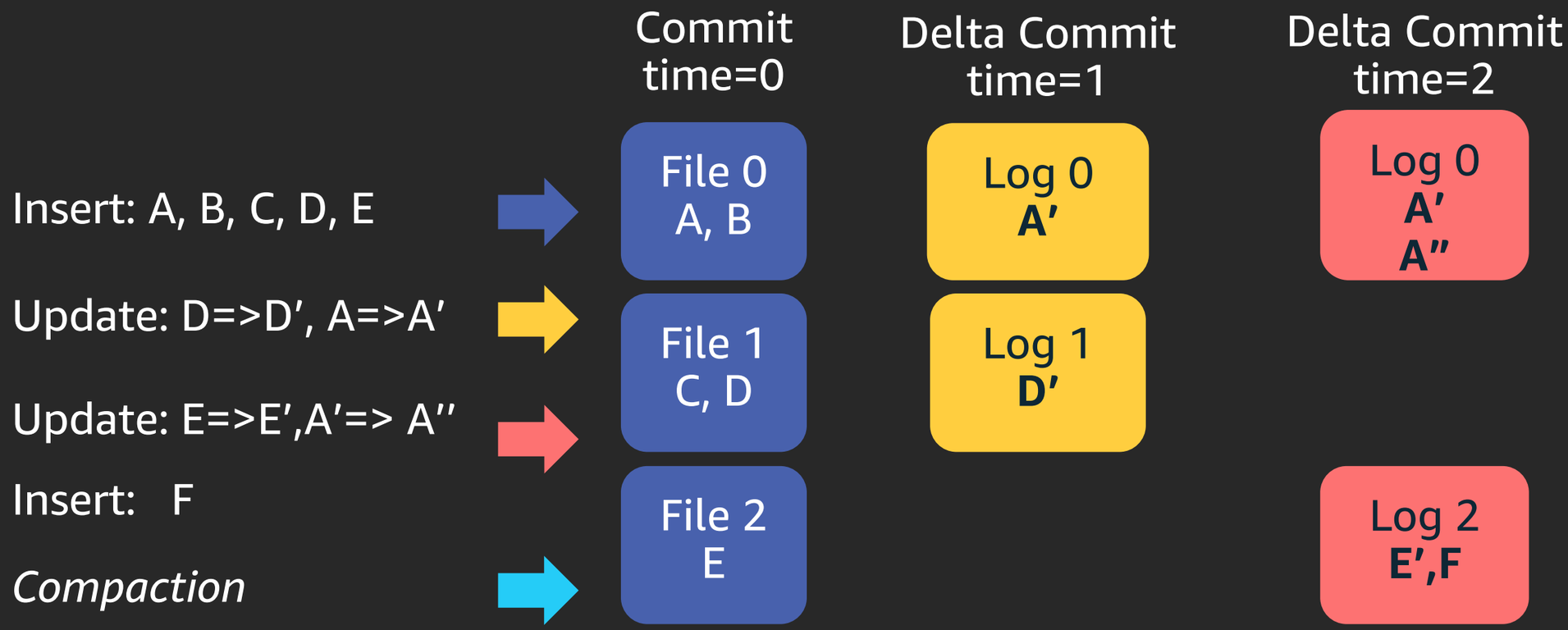


Real time	A,B,C,D,E	A',B,C,D',E
Incremental	A,B,C,D,E	A',D'
Read-Optimized	A,B,C,D,E	A,B,C,D,E

Storage types & Views

Storage type: Merge On Read

Views/Queries: Read Optimized, Incremental, Real time

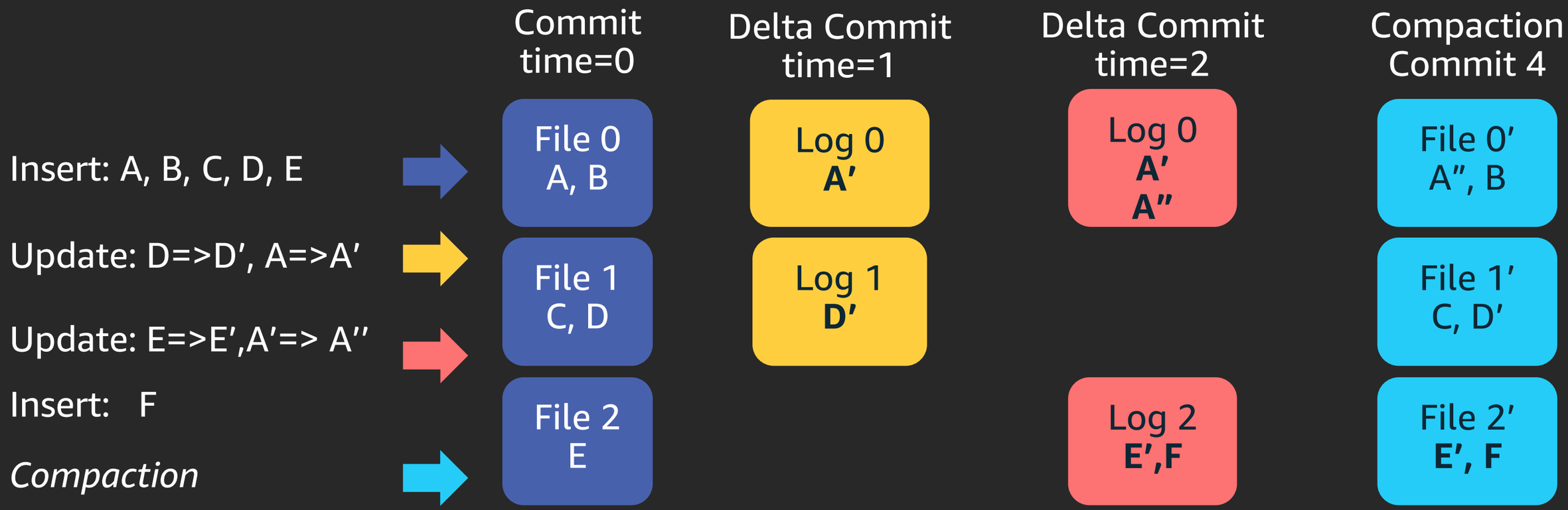


	Commit time=0	Delta Commit time=1	Delta Commit time=2
Real time	A,B,C,D,E	A',B,C,D',E	A'',B,C,D',E',F
Incremental	A,B,C,D,E	A',D'	A'',E', F
Read-Optimized	A,B,C,D,E	A,B,C,D,E	A,B,C,D,E

Storage types & Views

Storage type: Merge On Read

Views/Queries: Read Optimized, Incremental, Real time



Real time	A,B,C,D,E	A',B,C,D',E	A'',B,C,D',E',F	A'',B,C,D',E',F
Incremental	A,B,C,D,E	A',D'	A'',E', F	
Read-Optimized	A,B,C,D,E	A,B,C,D,E	A,B,C,D,E	A'',B,C,D',E',F

Storage types & Views

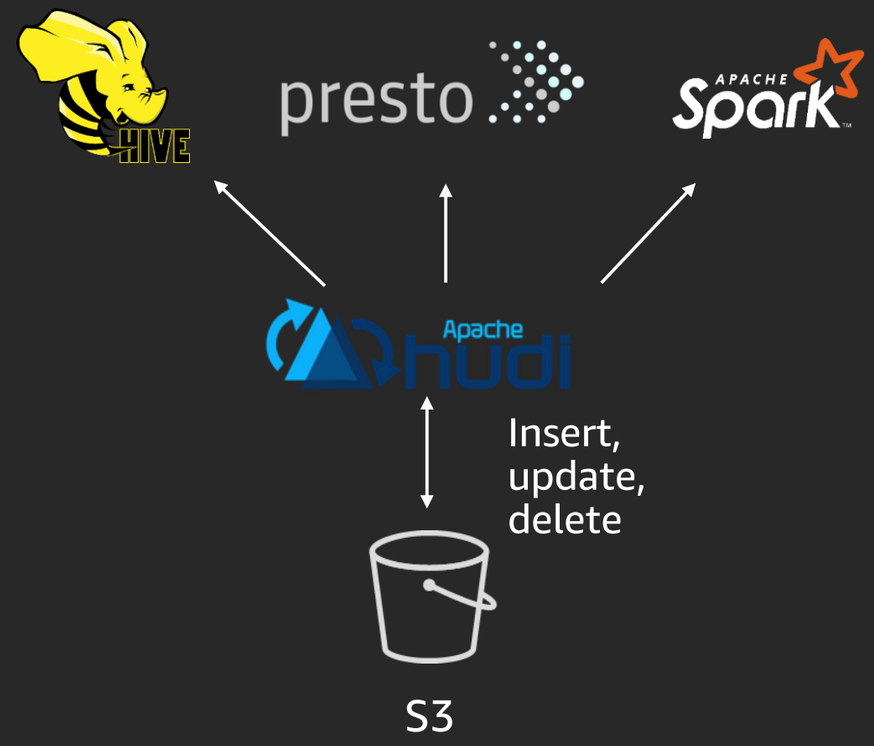
Storage type: Merge On Read

Views: Read Optimized, Incremental, Real time

When to Use

- You want ingested data available for query as fast as possible
- Your workload can have sudden spikes or changes in pattern
  - Example: Bulk updates to older transactions in upstream database cause updates to old partitions in Amazon S3

So why should you use Apache Hudi



Open source, open format, vendor neutral with Apache HUDI

Support for Spark, Hive, and Presto

Enables data lakes to

a) Comply with data privacy laws

b) Consume real-time streams and change data captures

c) Reinstate late arriving data

d) Track change history and rollback

Learn more

ANT239 - Insert, upsert, and delete data in Amazon S3 using Amazon EMR – Vinoth Chandar (Lead for Apache Hudi), Paul Codding (Amazon EMR) - Wednesday, Dec 4, 1:45 PM - 2:45 PM– Mirage, Events Center B3 Green

ANT301-R1 - [REPEAT 1] Build & optimize Spark data pipelines for incremental data processing (Workshop)

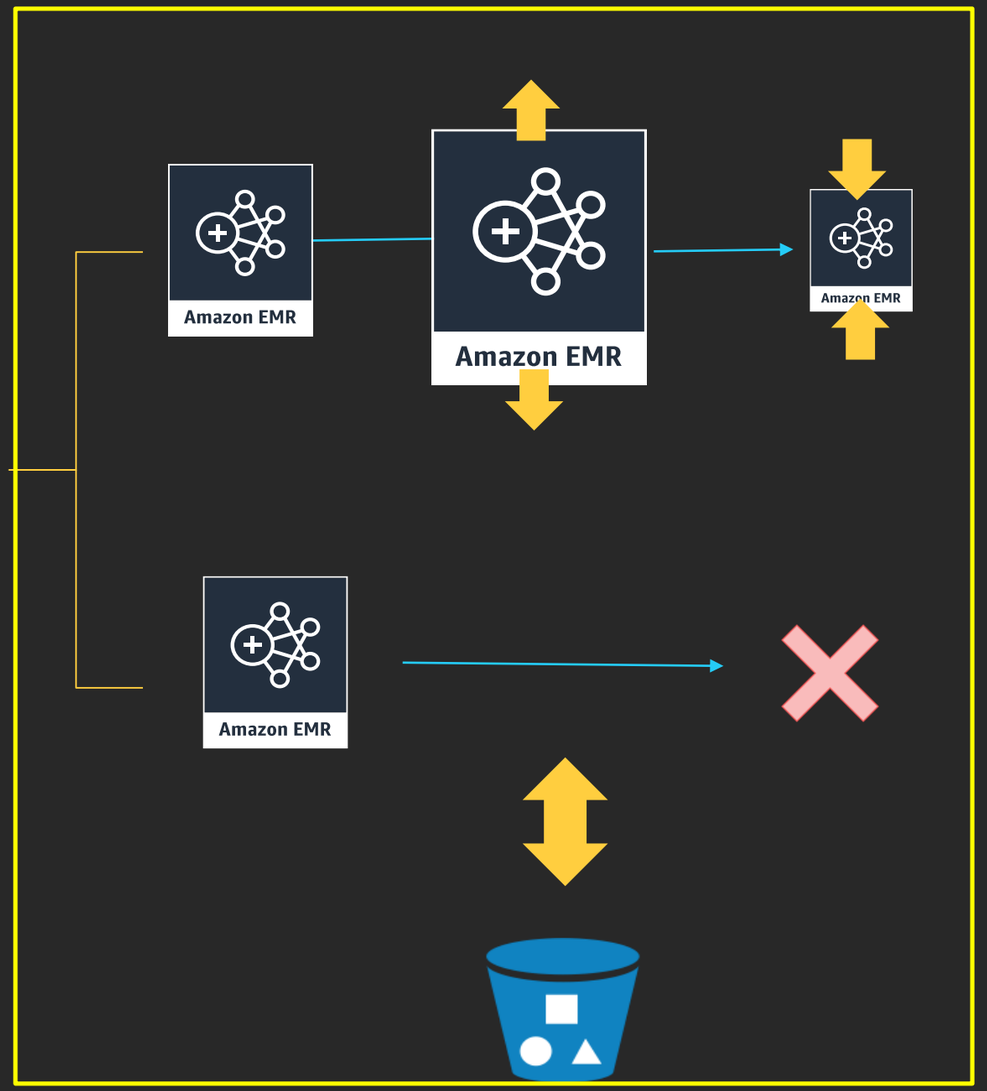
Demo – Apache Hudi with EMR

Simplified view of data engineering platforms

Notebooks

Apache Airflow

AWS Step Functions



Reduced cost with capacity-aware Spot provisioning

Scale up with Spot Instances

10 node cluster running for 14 hours
Cost = 1.0 * 10 * 14 = $140

Scale up cluster with Spot Instances

Add 10 more nodes on Spot

Scale up cluster with Spot Instances

20 node cluster running for 7 hours
Cost = 1.0 * 10 * 7 = $70
= 0.5 * 10 * 7 = $35

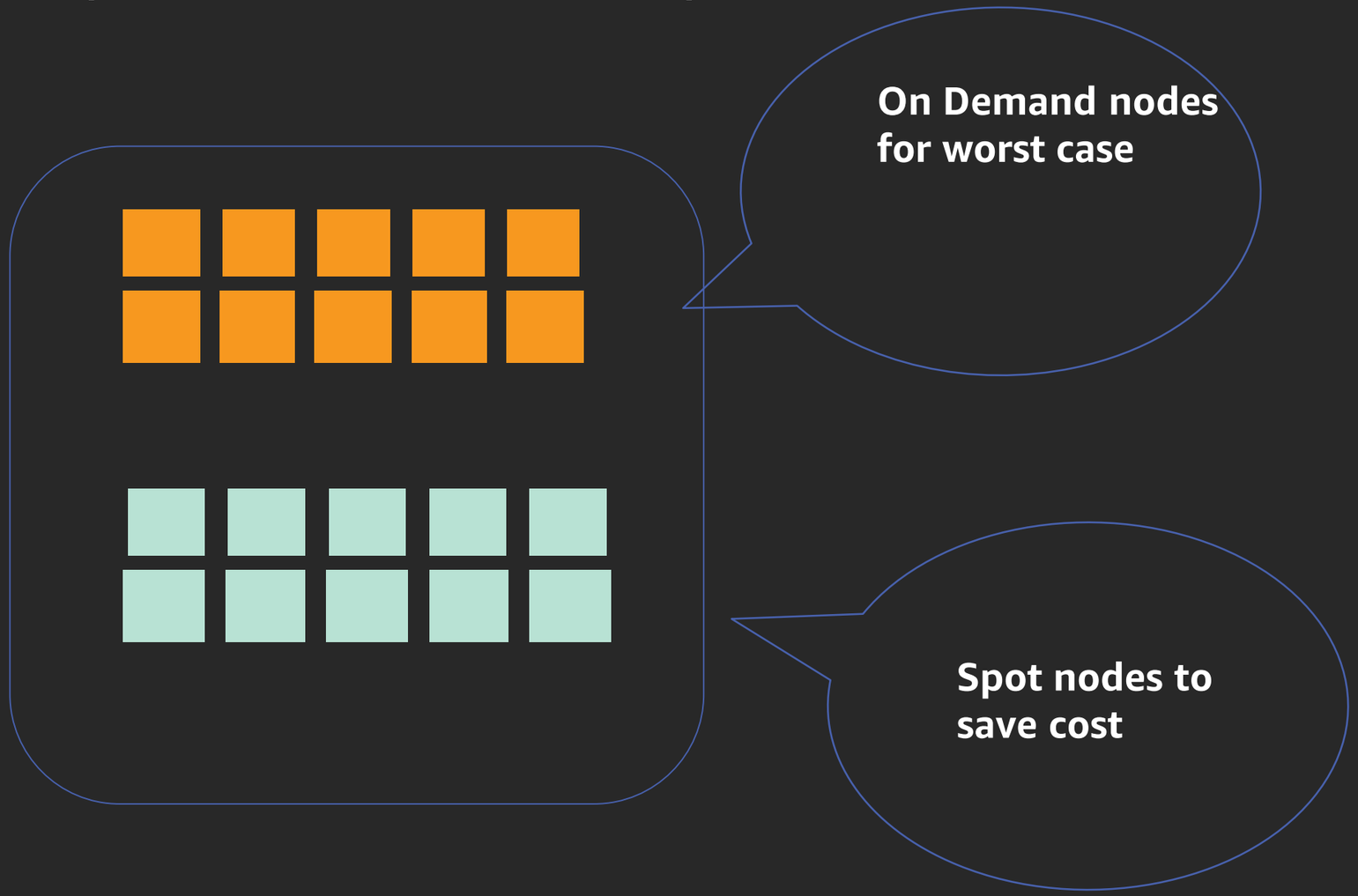
Total $105

Scale up cluster with Spot Instances

50 % less run-time (14 → 7)

25% less cost (140 → 105)

Scale up cluster with Spot Instances



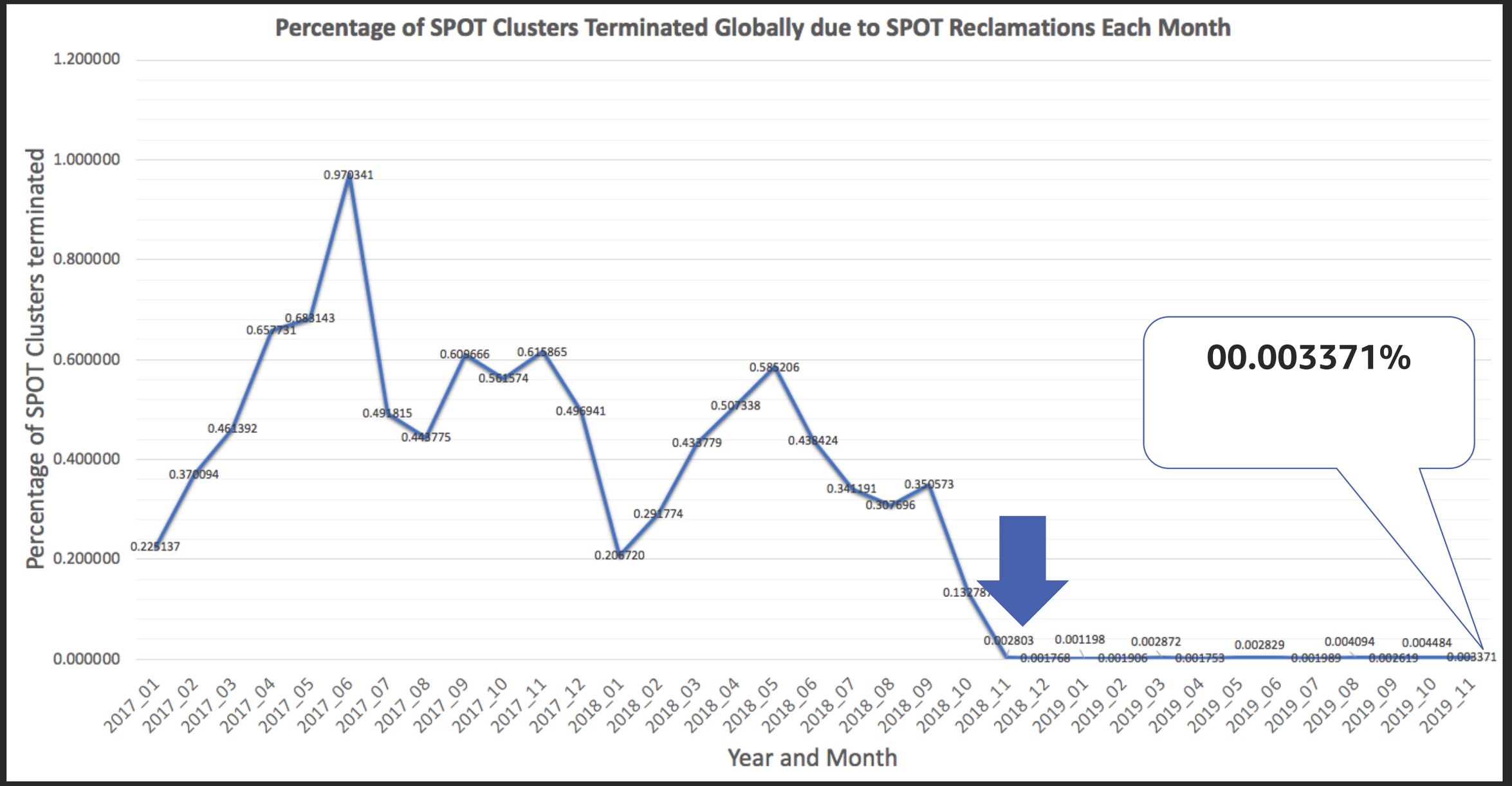
Scale up cluster with Spot Instances

On-demand nodes for worst case

Up to 60% lower cost with compute savings plan

Spot nodes to save cost

“Spot” the interruption



Managed resize (private beta)

Managed resize (beta)

Completely managed environment for automatically resizing clusters

- No configurations required except min/max cost constraints
- More data points and faster reaction time
- Can save 20-60% costs depending on the workload pattern

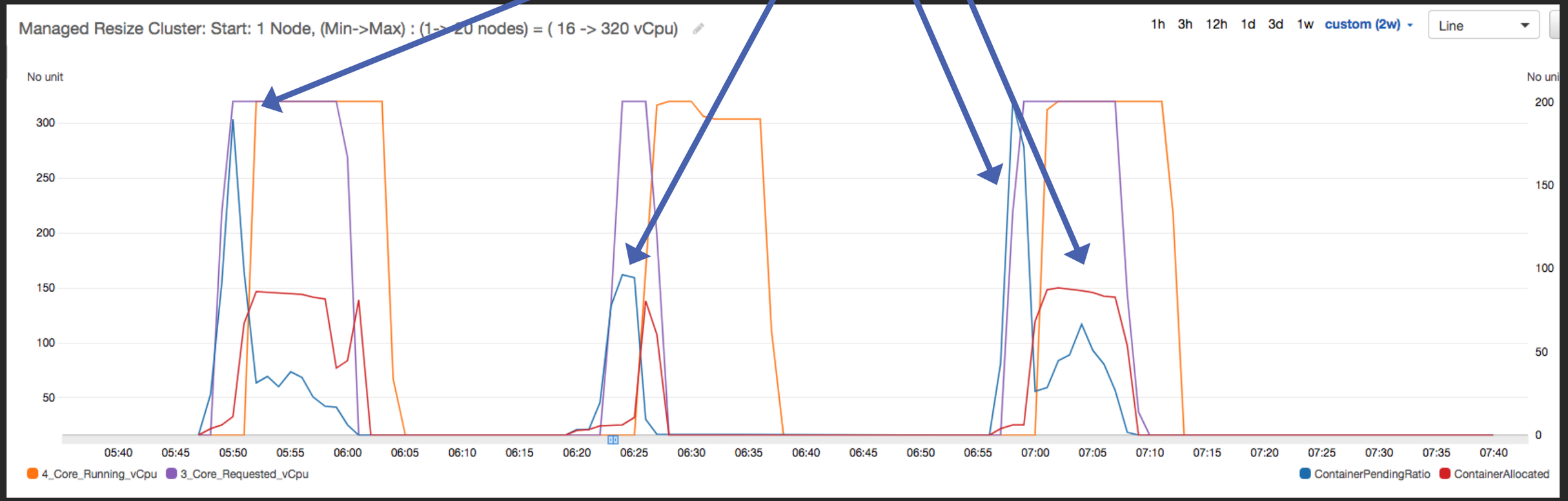
Similar to autoscaling Amazon EMR Clusters

- Use autoscaling for DIY scaling with custom metrics
- Managed resize for completely managed option

Email emr-pm@amazon.com to participate in the beta

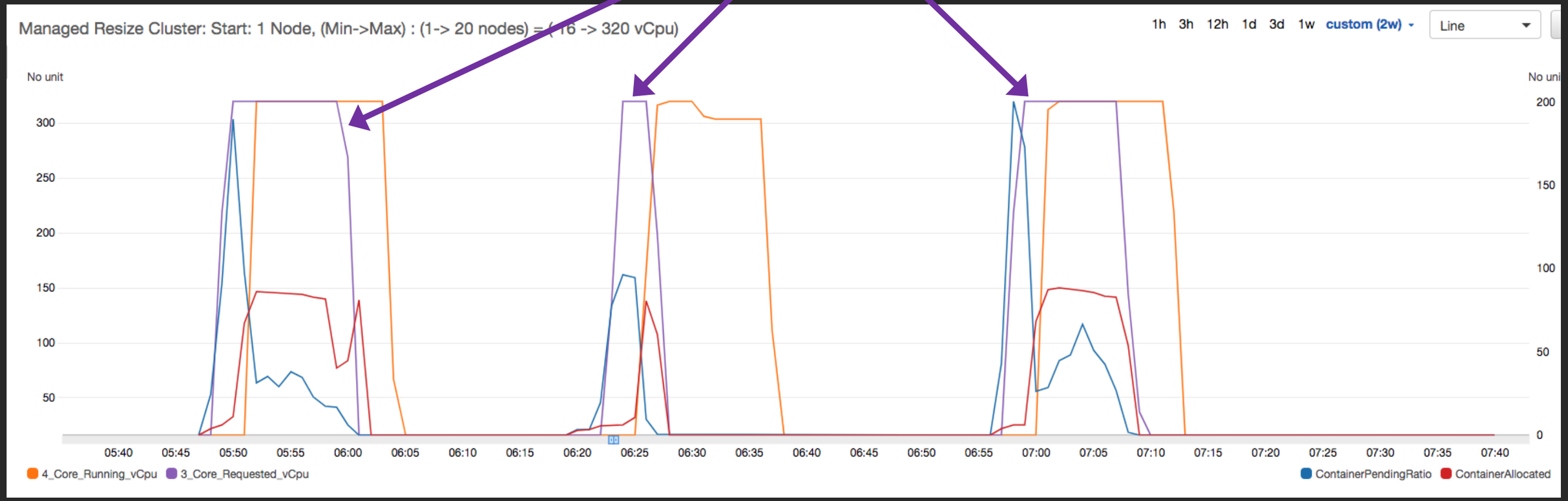
Managed resize

Load patterns



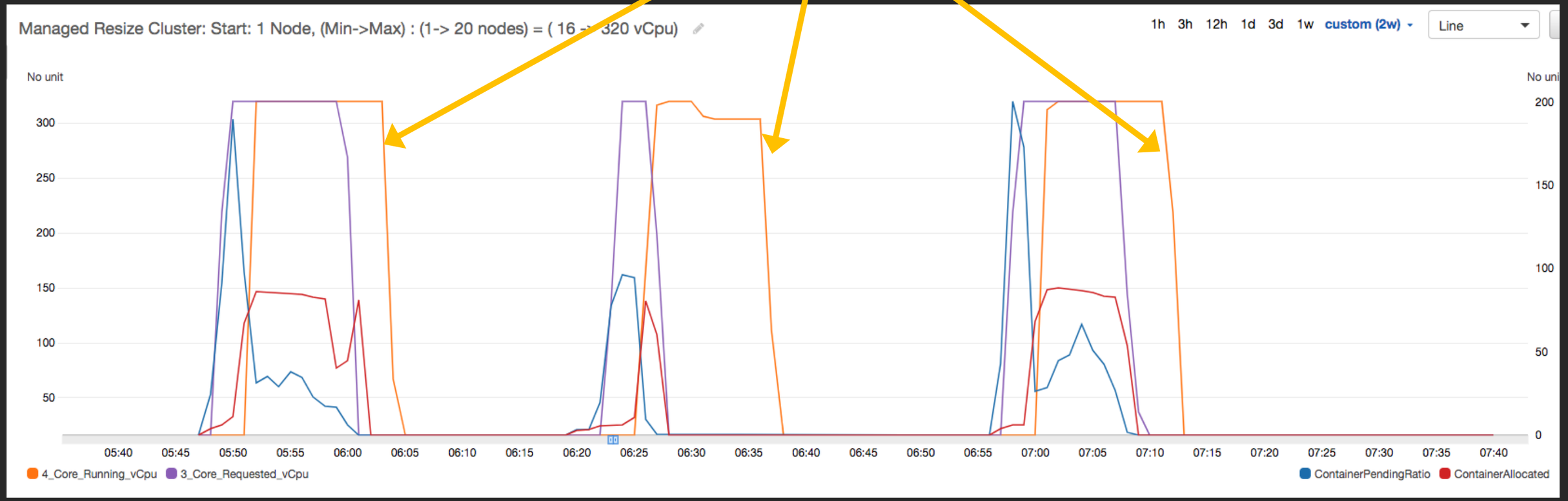
Managed resize

Requested scale



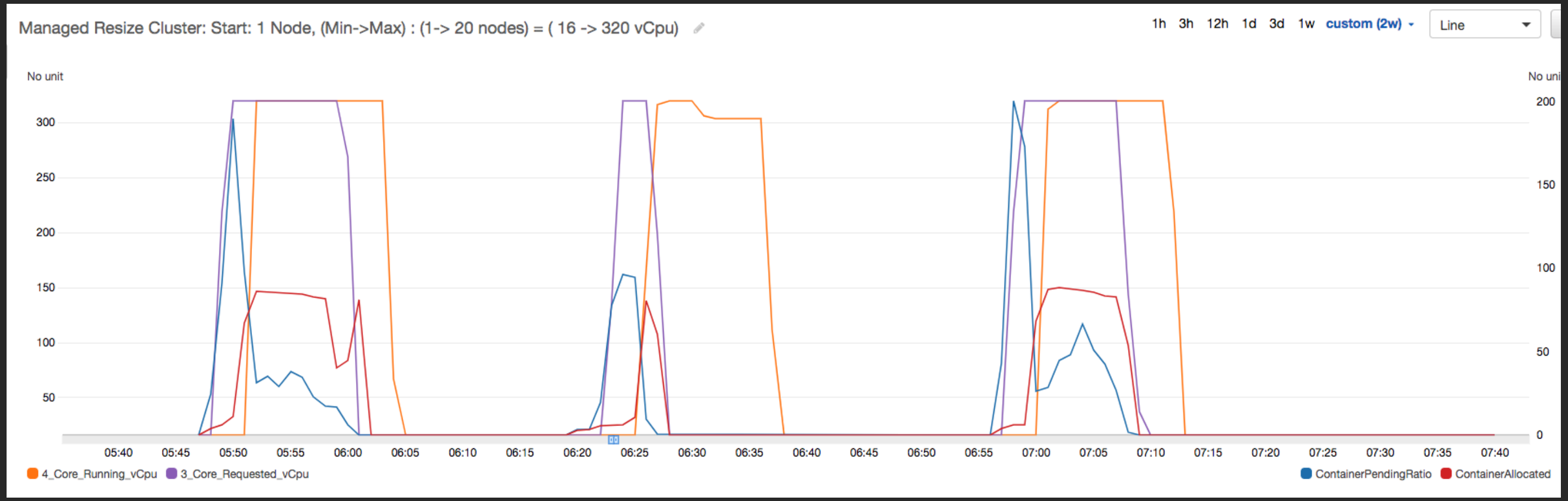
Managed resize

Scale down



Managed resize

63% savings compared to fixed 20-node cluster

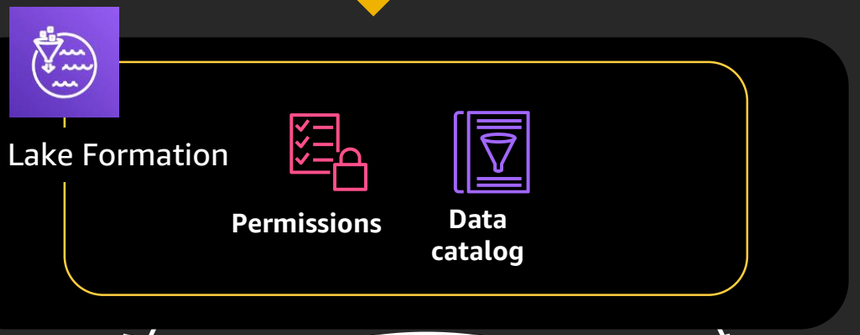


5

Fine-grained access control with AWS Lake Formation

Lake Formation: Secure once, access in many ways

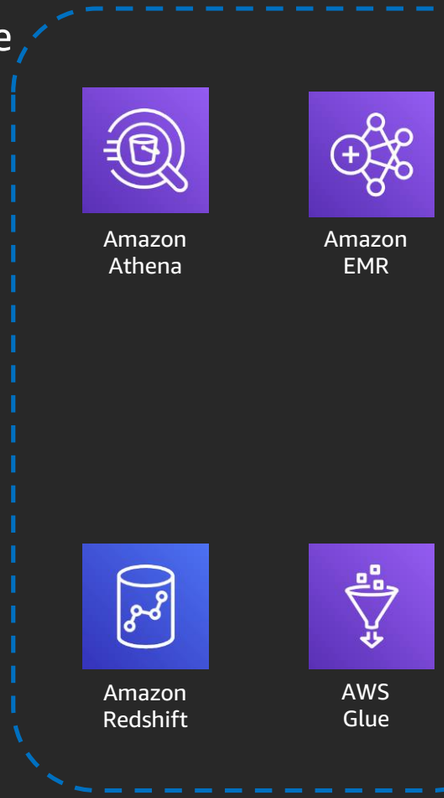
1. Set up user access in Lake Formation



Amazon S3

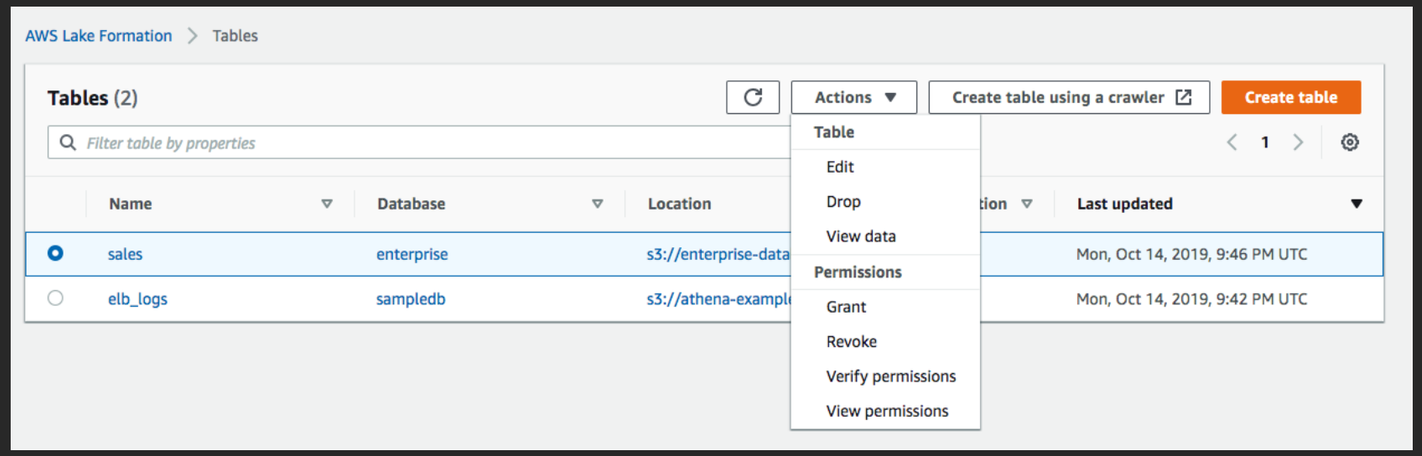
3. Service signs the user into Lake Formation

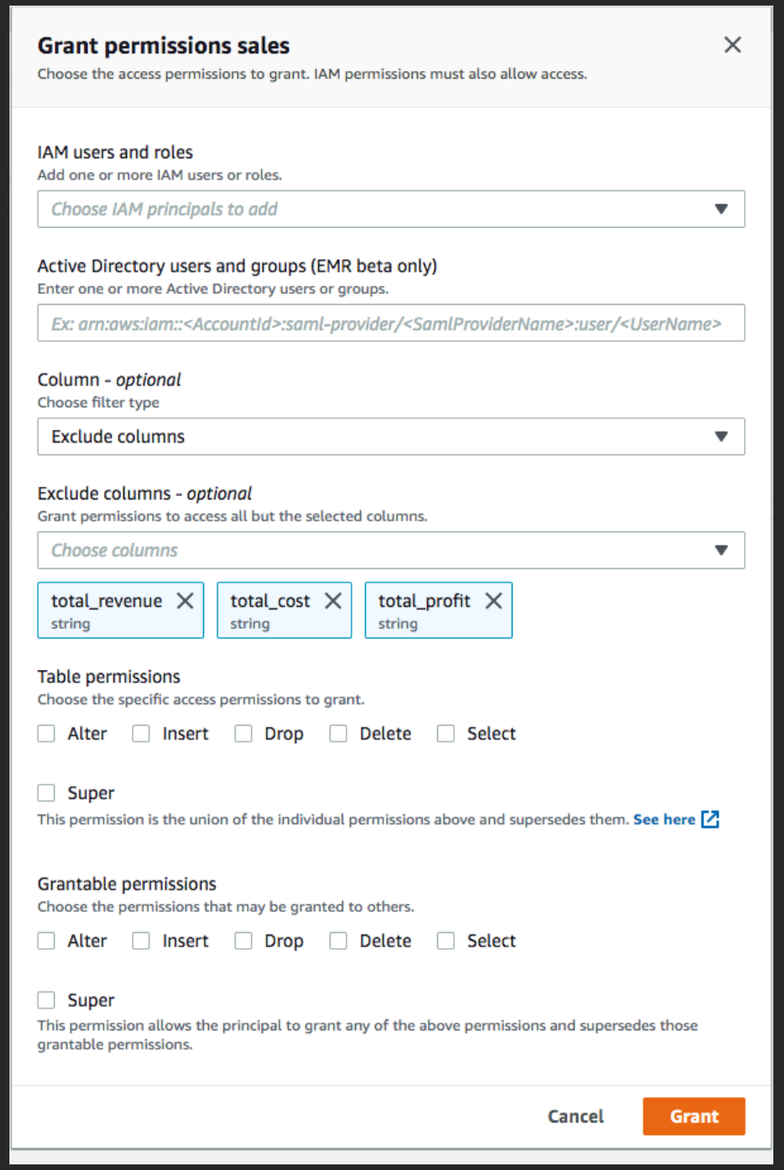
4. Lake Formation returns temporary credentials allowing data access



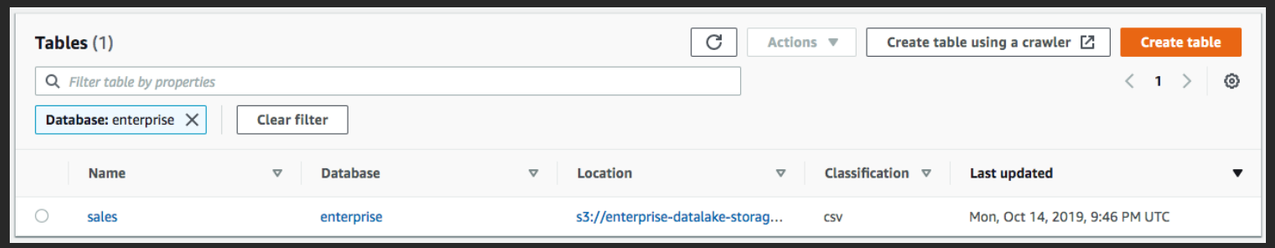
2. User tries to access data via one of the services

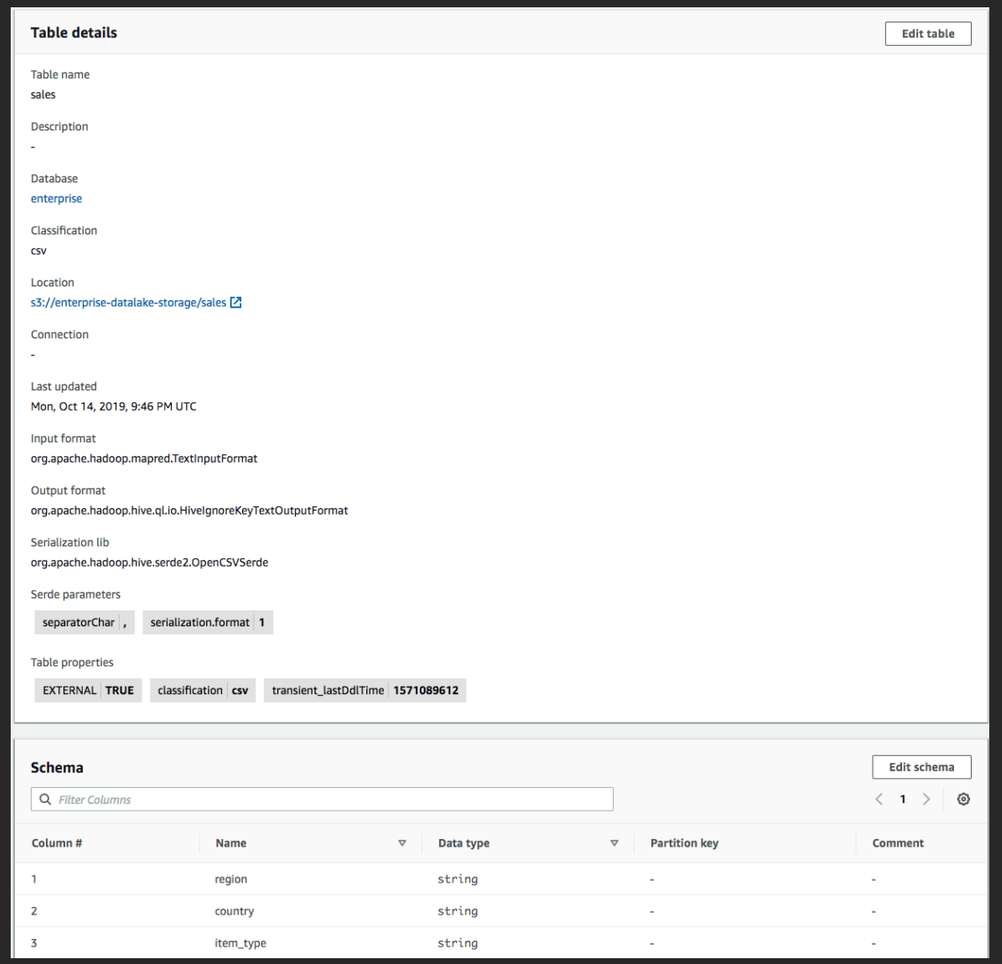
Control data access with GRANT and REVOKE





Permissions on tables and columns, not Amazon Simple Storage Service (Amazon S3)

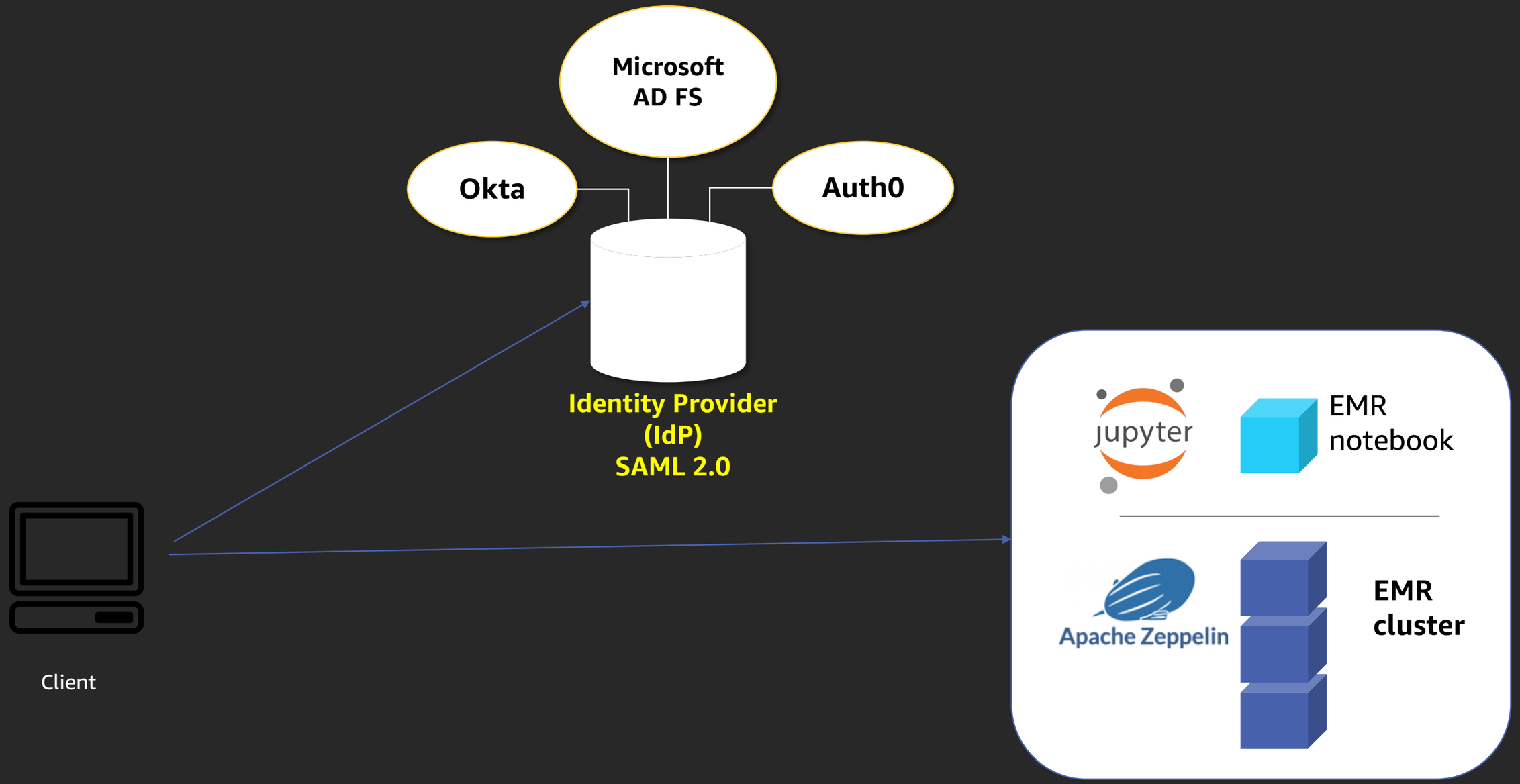




Why integrate Amazon EMR with Lake Formation?

- Fine-grained, column-level access to databases and tables
  - Allows shared multi-tenant clusters to securely access data
  - Uses the AWS Glue Data Catalog as the metadata store
- Federated single sign-on from your enterprise identity system
  - Active Directory Federation Services (AD FS), Auth0, Okta, and many others
  - Uses Security Assertion Markup Language (SAML) 2.0

Enterprise single sign-on access to notebooks



(PNG Image, 300 × 236 pixels)

about:logo

Zeppelin

https://ec2-35-170-74-144.compute-1.amazonaws.com:8442/gateway/default/zeppelin/#/

Zeppelin Notebook Job Search bob

Welcome to Zeppelin!

Zeppelin is web-based notebook that enables interactive data analytics.
You can make beautiful data-driven, interactive, collaborative document with SQL, code and even more!

Notebook

Import note

Create new note

Filter

Zeppelin Tutorial

Trash

Help

Get started with Zeppelin documentation

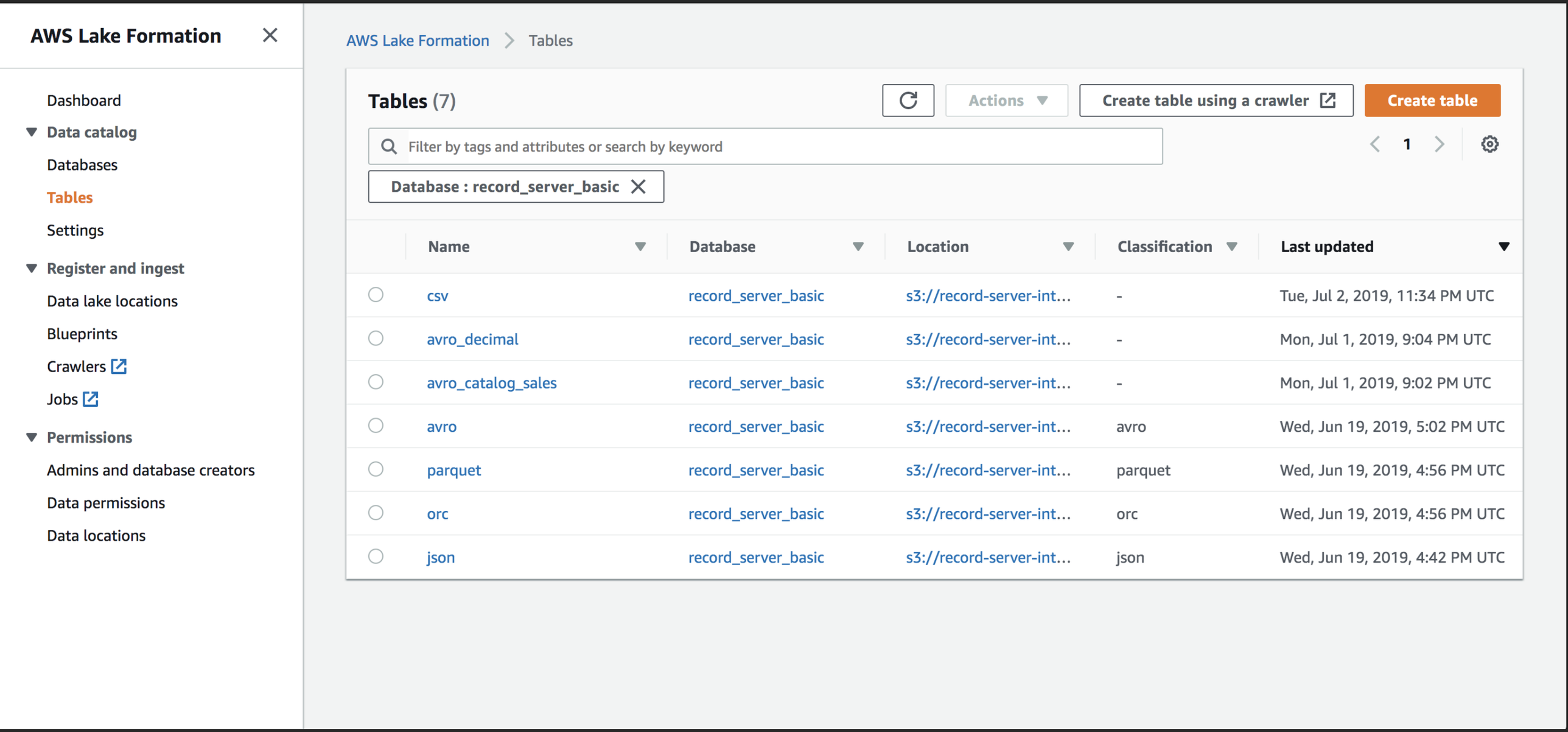
Community

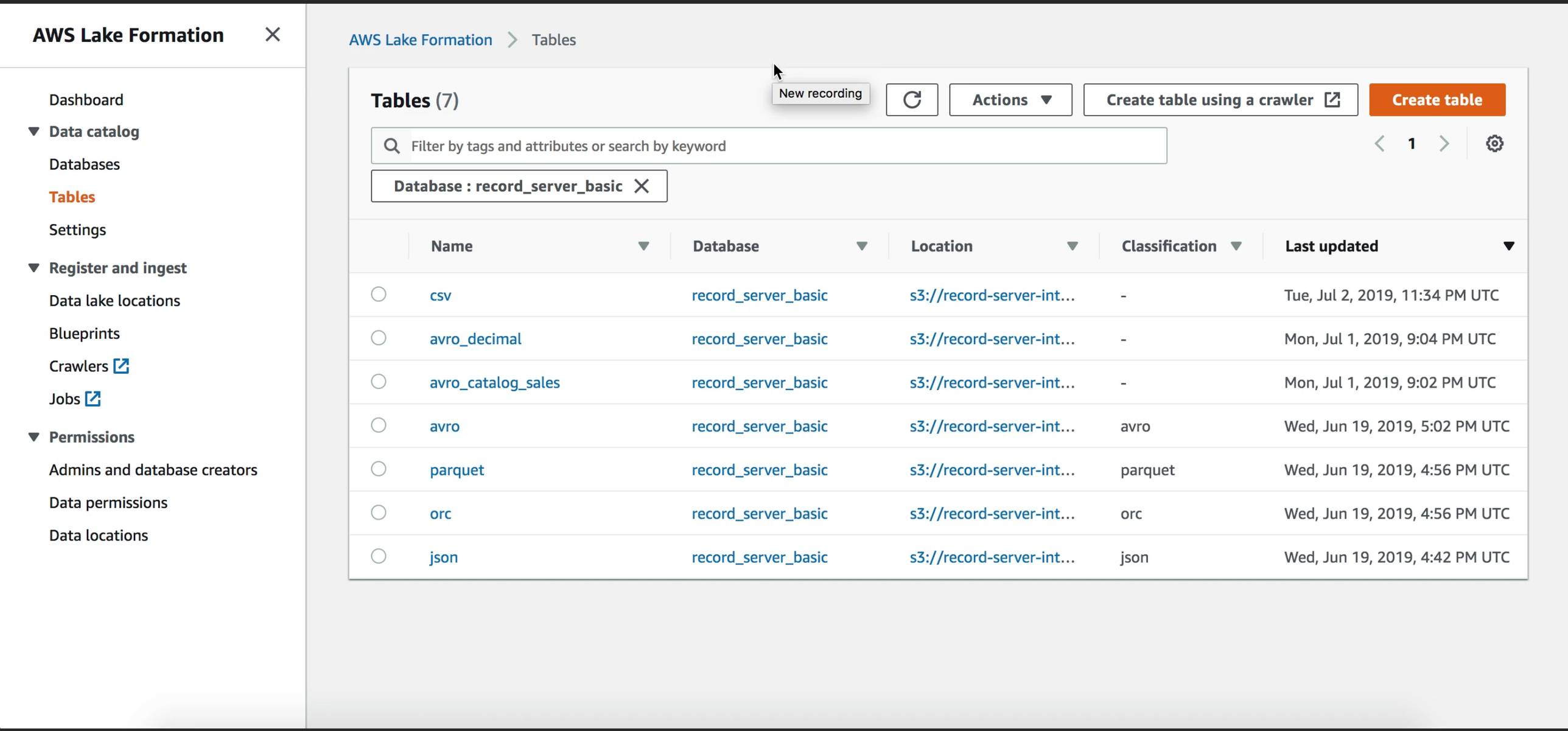
Please feel free to help us to improve Zeppelin,
Any contribution are welcome!

Mailing list

Issues tracking

Github


AWS Lake Formation
Dashboard
Data catalog
Databases
Tables
Settings
Register and ingest
Data lake locations
Blueprints
Crawlers
Jobs
Permissions
Admins and database creators
Data permissions
Data locations
AWS Lake Formation > Tables
Tables (7)
Actions
Create table using a crawler
Create table
Filter by tags and attributes or search by keyword
1
Database : record_server_basic
Name
Database
Location
Classification
Last updated
csv record_server_basic s3://record-server-int... - Tue, Jul 2, 2019, 11:34 PM UTC
avro_decimal record_server_basic s3://record-server-int... - Mon, Jul 1, 2019, 9:04 PM UTC
avro_catalog_sales record_server_basic s3://record-server-int... - Mon, Jul 1, 2019, 9:02 PM UTC
avro record_server_basic s3://record-server-int... avro Wed, Jun 19, 2019, 5:02 PM UTC
parquet record_server_basic s3://record-server-int... parquet Wed, Jun 19, 2019, 4:56 PM UTC
orc record_server_basic s3://record-server-int... orc Wed, Jun 19, 2019, 4:56 PM UTC
json record_server_basic s3://record-server-int... json Wed, Jun 19, 2019, 4:42 PM UTC


AWS Lake Formation
Dashboard
Data catalog
Databases
Tables
Settings
Register and ingest
Data lake locations
Blueprints
Crawlers
Jobs
Permissions
Admins and database creators
Data permissions
Data locations
AWS Lake Formation > Tables
Tables (7)
New recording
Actions
Create table using a crawler
Create table
Filter by tags and attributes or search by keyword
1
Database : record_server_basic
Name
Database
Location
Classification
Last updated
csv record_server_basic s3://record-server-int... - Tue, Jul 2, 2019, 11:34 PM UTC
avro_decimal record_server_basic s3://record-server-int... - Mon, Jul 1, 2019, 9:04 PM UTC
avro_catalog_sales record_server_basic s3://record-server-int... - Mon, Jul 1, 2019, 9:02 PM UTC
avro record_server_basic s3://record-server-int... avro Wed, Jun 19, 2019, 5:02 PM UTC
parquet record_server_basic s3://record-server-int... parquet Wed, Jun 19, 2019, 4:56 PM UTC
orc record_server_basic s3://record-server-int... orc Wed, Jun 19, 2019, 4:56 PM UTC
json record_server_basic s3://record-server-int... json Wed, Jun 19, 2019, 4:42 PM UTC

(PNG Image, 300 × 236 pixels)

https://ec2-35-170-74-144.compute-1.amazonaws.com:8442/gateway/default/zeppelin/#/

New recording

Spark SQL - Zeppelin | All Applications

ec2-35-170-74-144.compute-1.amazonaws.com:8088/cluster

All Applications

- Cluster
  - About
  - Nodes
  - Node Labels
  - Applications
    - NEW
    - NEW_SAVING
    - SUBMITTED
    - ACCEPTED
    - RUNNING
    - FINISHED
    - FAILED
    - KILLED
  - Scheduler
- Tools

Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	VCo
3	0	1	2	1	1.38 GB	12 GB	0 B	1

Cluster Nodes Metrics

Active Nodes	Decommissioning Nodes	Decommissioned Nodes	Lost Nodes	Unhealthy Nodes	Re
1	0	0	0	0	0

Scheduler Metrics

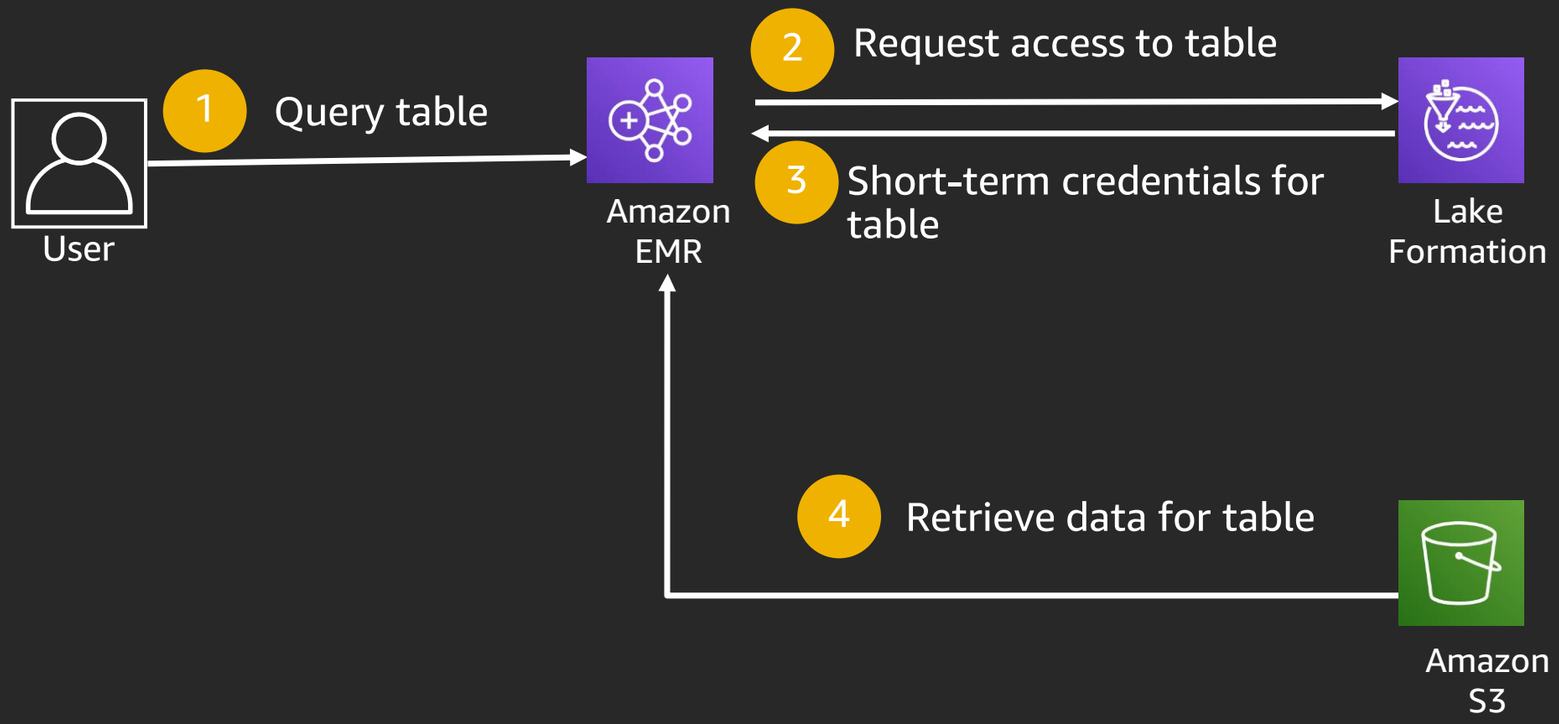
Scheduler Type	Scheduling Resource Type	Minimum Allocation	Maximum Allocation	
Capacity Scheduler	[MEMORY]	<memory:32, vCores:1>	<memory:12288, vCores:8>	0

Show 20 entries

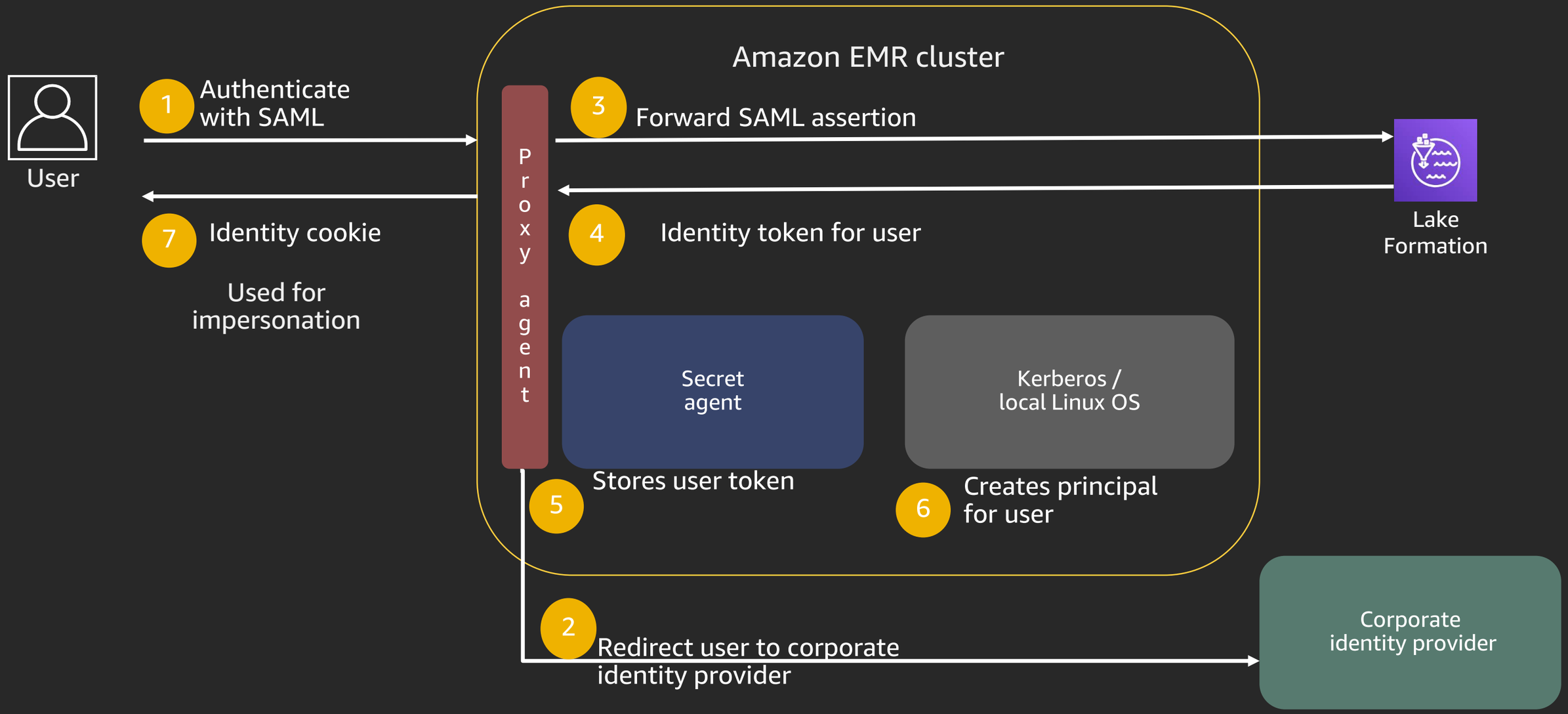
ID	User	Name	Application Type	Queue	Application Priority	StartTime	FinishTime	State	FinalStatus	Running Containers	Allocated CPU VCores	Allocated Memory MB	% of Queue
application_1564091049305_0003	paul	livy-ssion-2	SPARK	default	0	Fri Jul 26 08:33:56 -0700 2019	N/A	RUNNING	UNDEFINED	1	1	1408	11.5
application_1564091049305_0002	bob	livy-ssion-1	SPARK	default	0	Fri Jul 26 07:09:21 -0700 2019	Fri Jul 26 08:16:33 -0700 2019	FINISHED	SUCCEEDED	N/A	N/A	N/A	0.0
application_1564091049305_0001	bob	livy-session-0	SPARK	default	0	Thu Jul 25 15:01:06 -0700 2019	Thu Jul 25 16:07:32 -0700 2019	FINISHED	SUCCEEDED	N/A	N/A	N/A	0.0

Showing 1 to 3 of 3 entries

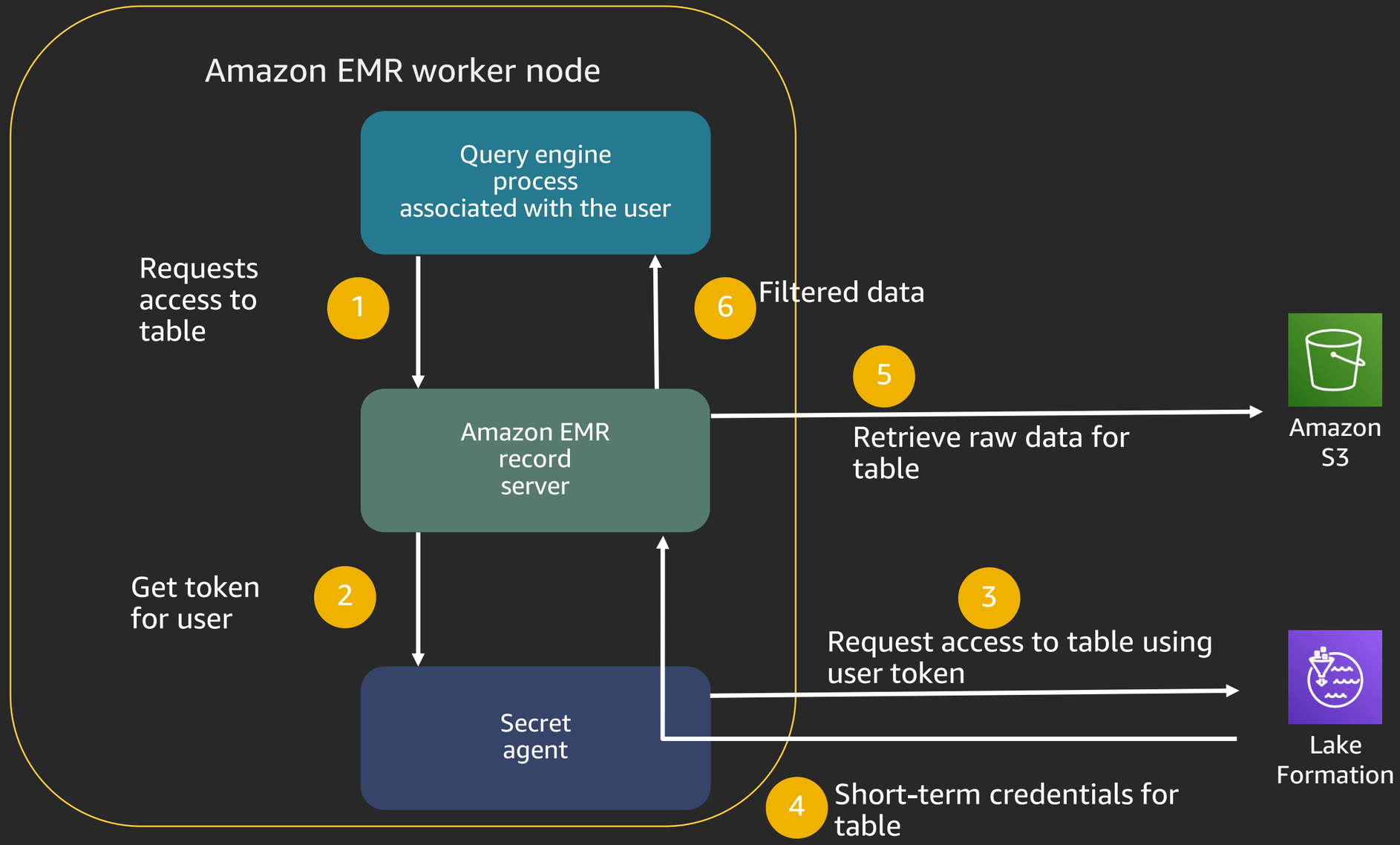
Query execution overview



Amazon EMR authentication



Query execution under the hood



Amazon EMR: Supported applications

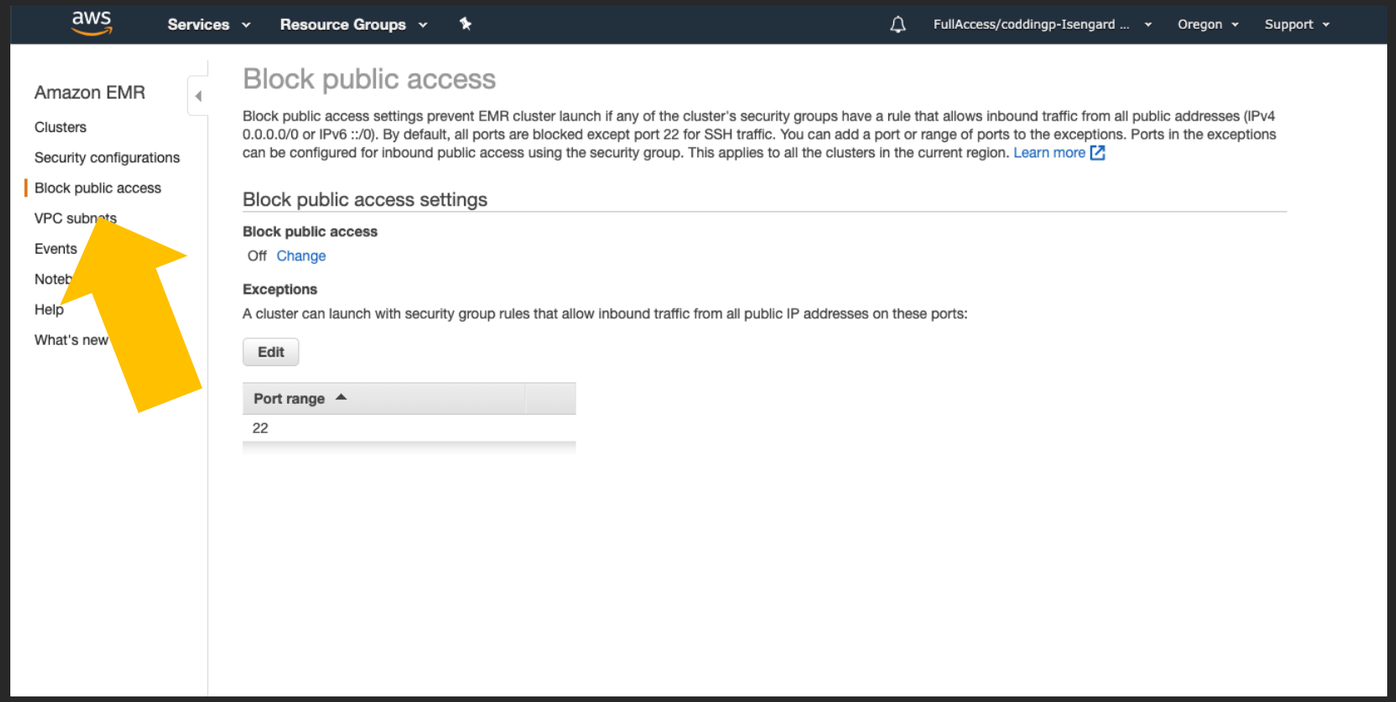
- AWS Glue Data Catalog
- Identity providers with support for SAML
- Applications
  - Spark SQL
  - Amazon EMR notebooks and Zeppelin with Livy

6

Block public access to open ports

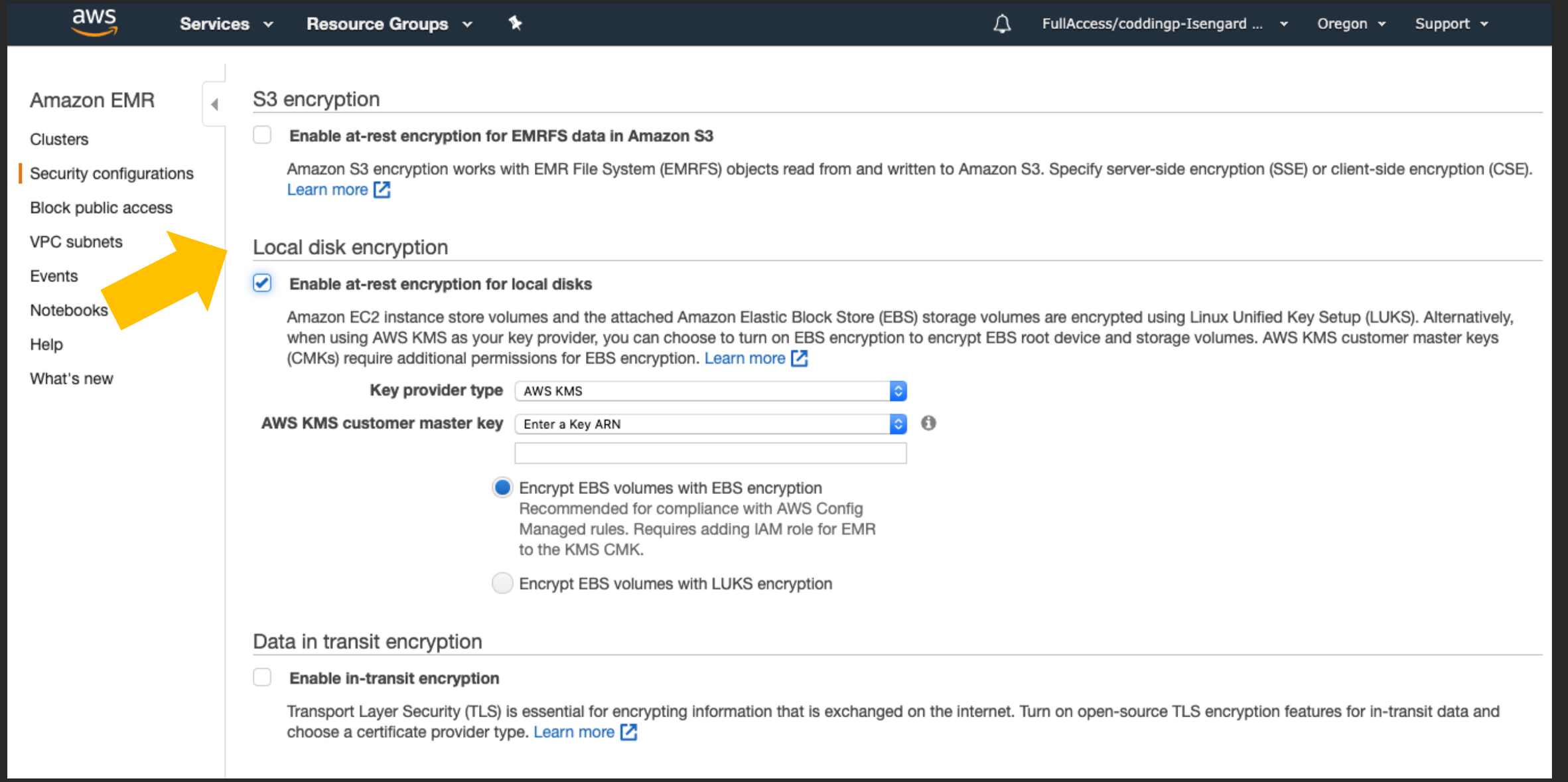
Block unintended network exposure

- Prevent users from launching clusters with security groups that allow access from any IP address (IPv4 0.0.0.0/0 or IPv6 ::/)
- Policy set at the account level, with region-specific configuration
- Allows exceptions to for public access to a single port or port range

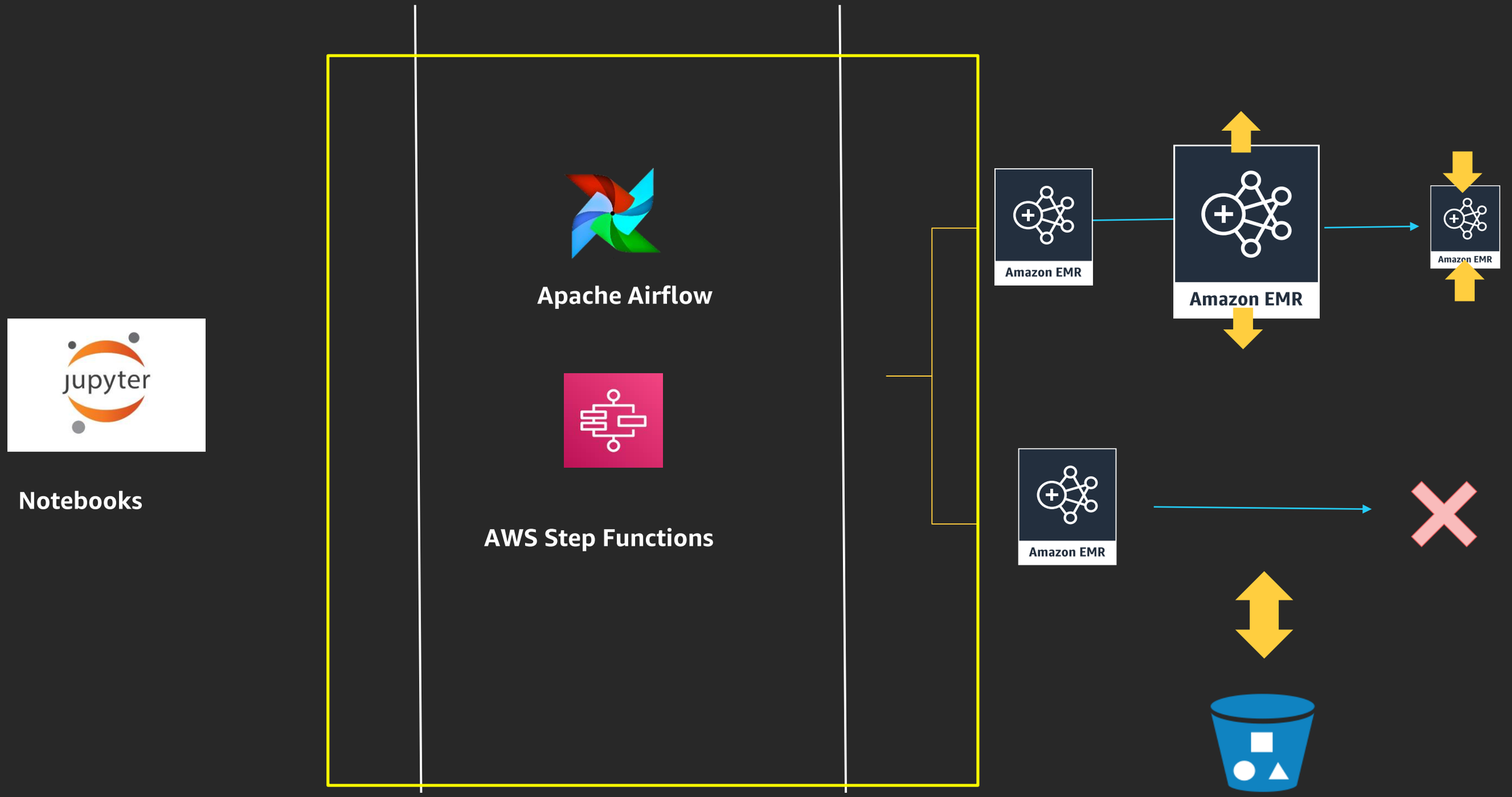


7

Native EBS encryption


aws
Services
Resource Groups
FullAccess/coddingp-Isengard ...
Oregon
Support
Amazon EMR
Clusters
Security configurations
Block public access
VPC subnets
Events
Notebooks
Help
What's new
S3 encryption
Enable at-rest encryption for EMRFS data in Amazon S3
Amazon S3 encryption works with EMR File System (EMRFS) objects read from and written to Amazon S3. Specify server-side encryption (SSE) or client-side encryption (CSE).
Learn more
Local disk encryption
Enable at-rest encryption for local disks
Amazon EC2 instance store volumes and the attached Amazon Elastic Block Store (EBS) storage volumes are encrypted using Linux Unified Key Setup (LUKS). Alternatively, when using AWS KMS as your key provider, you can choose to turn on EBS encryption to encrypt EBS root device and storage volumes. AWS KMS customer master keys (CMKs) require additional permissions for EBS encryption. Learn more
Key provider type
AWS KMS
AWS KMS customer master key
Enter a Key ARN
Encrypt EBS volumes with EBS encryption
Recommended for compliance with AWS Config Managed rules. Requires adding IAM role for EMR to the KMS CMK.
Encrypt EBS volumes with LUKS encryption
Data in transit encryption
Enable in-transit encryption
Transport Layer Security (TLS) is essential for encrypting information that is exchanged on the internet. Turn on open-source TLS encryption features for in-transit data and choose a certificate provider type. Learn more

Simplified view of data engineering platforms



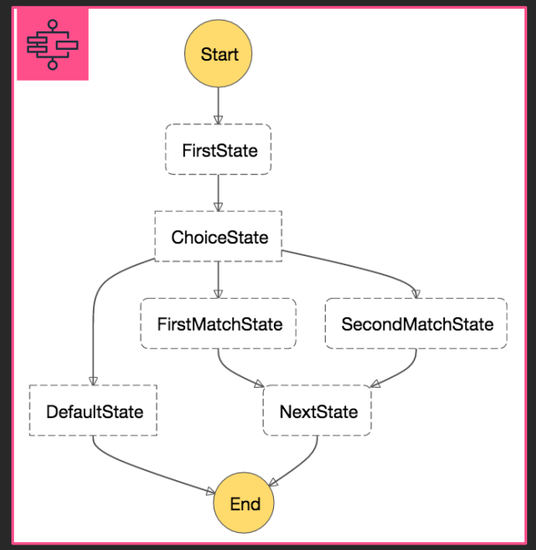
Integration with AWS Step Functions

AWS Step Functions: Visual workflows

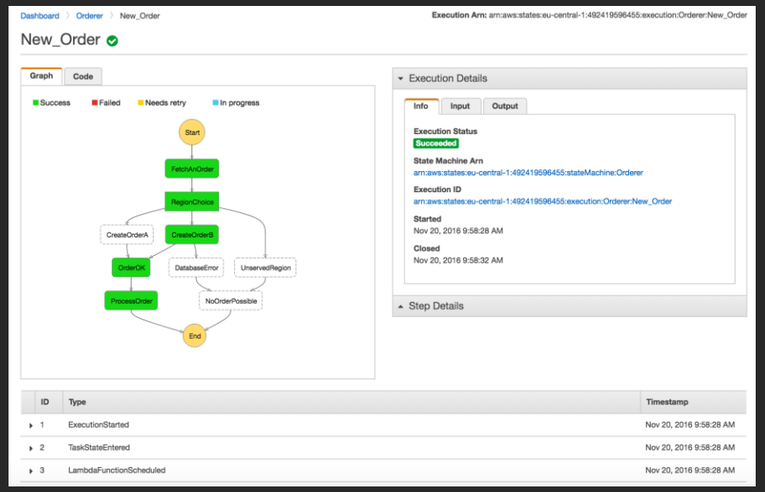
Define in JSON or Python

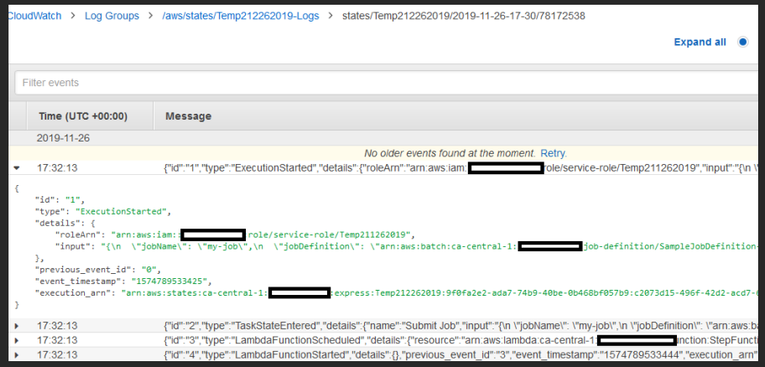
```
Code
{
  "Comment": "An AWL example using a choice state.",
  "StartAt": "FirstState",
  "States": {
    "FirstState": {
      "Type": "Task",
      "Resource": "arn:aws:lambda:REGION:ACCOUNT_ID:function:FUNCTION_NAME",
      "Next": "ChoiceState"
    },
    "ChoiceState": {
      "Type" : "Choice",
      "Choices": [
        {
```

Visualize in the console



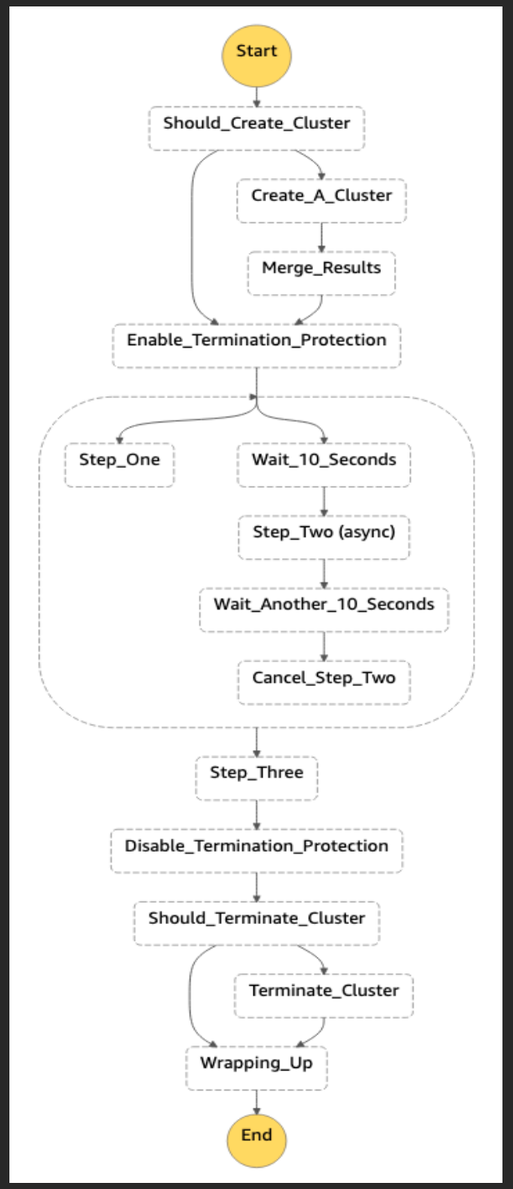
Monitor executions

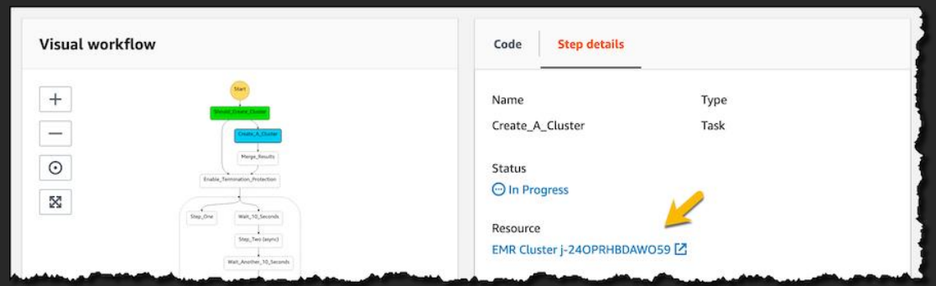


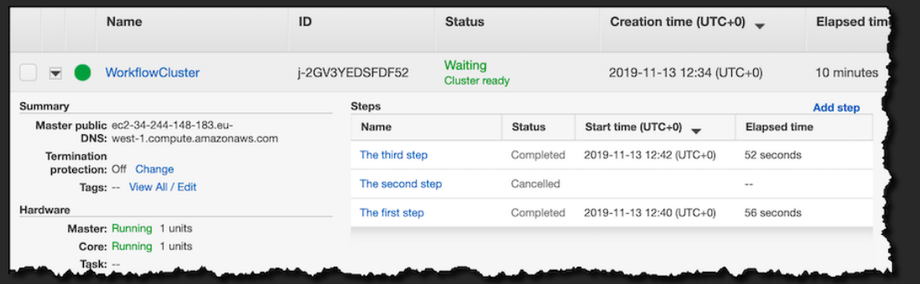


Automate workflows using AWS Step Functions

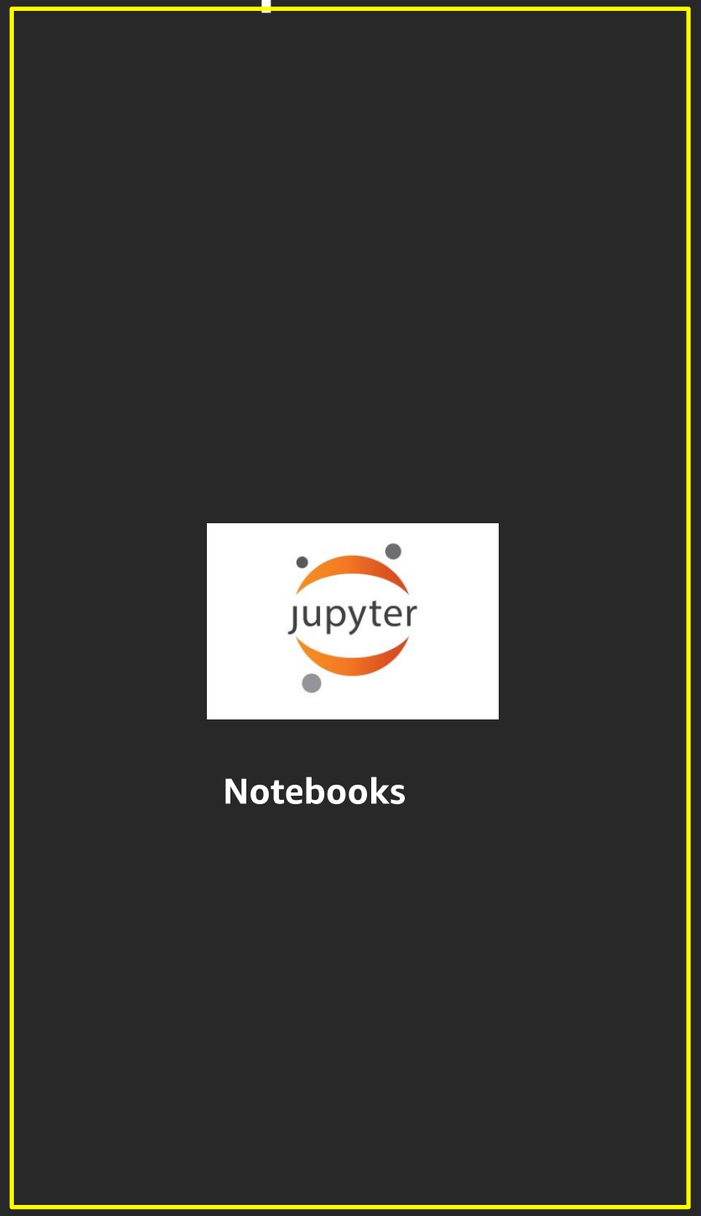
1. Create, scale, and modify clusters
2. Add, cancel, or run steps in parallel
3. Synchronous and asynchronous steps
4. Handle exceptions/failures
5. Scale clusters up or down
6. Reuse clusters
7. Terminate them





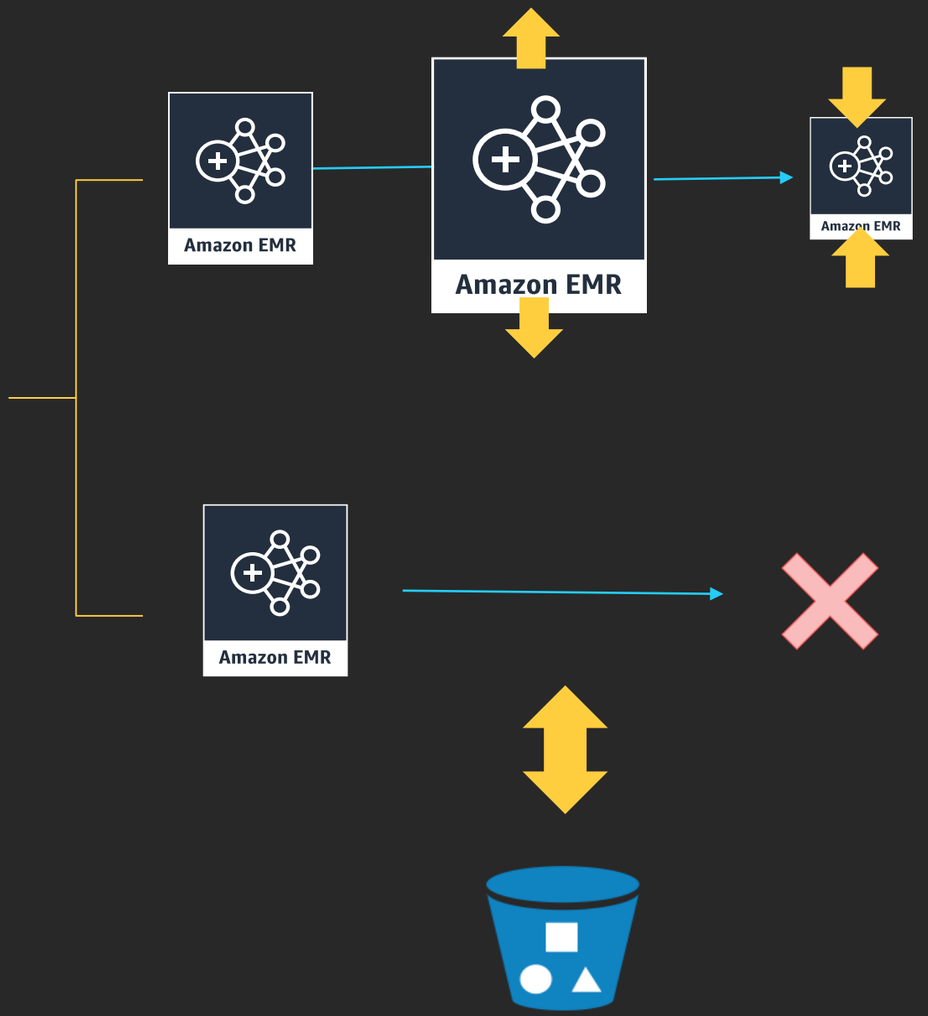


Simplified view of data engineering platforms



Apache Airflow

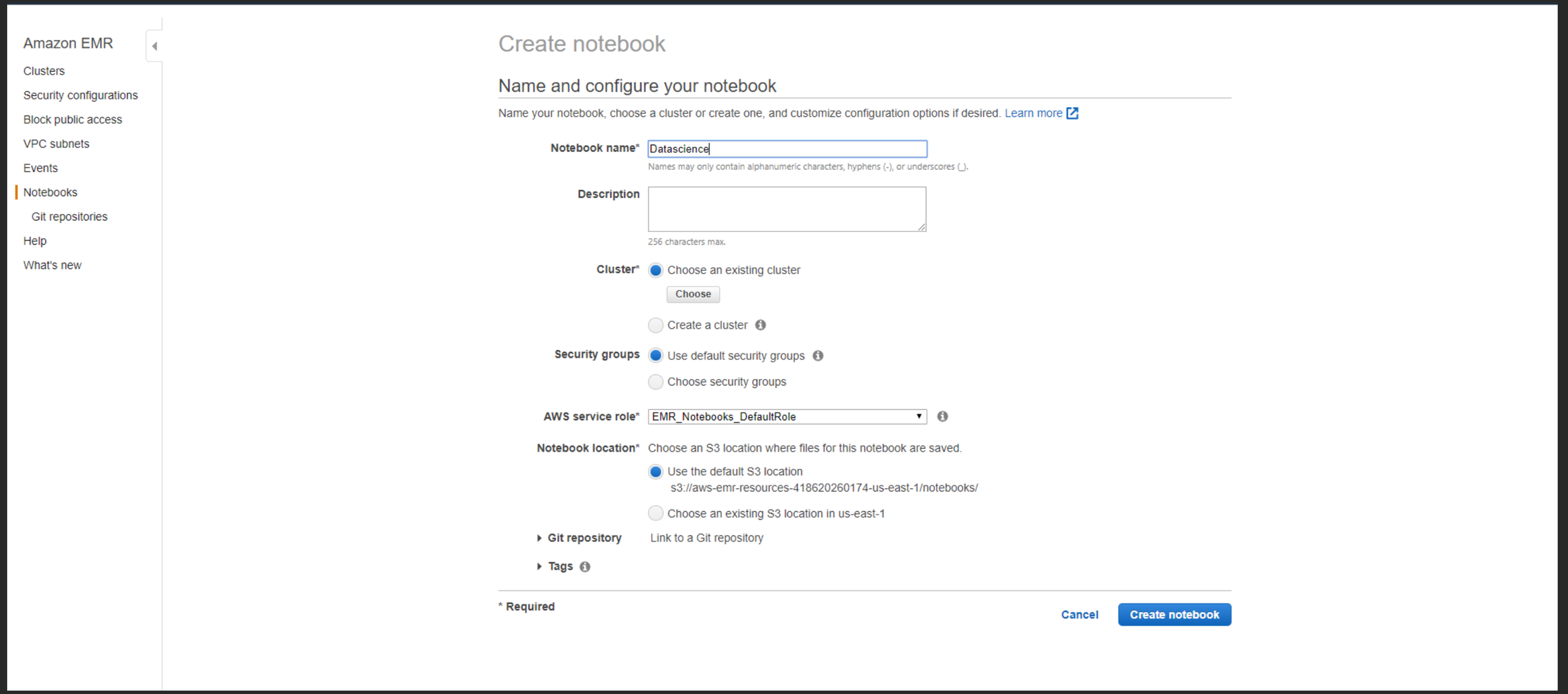
AWS Step Functions



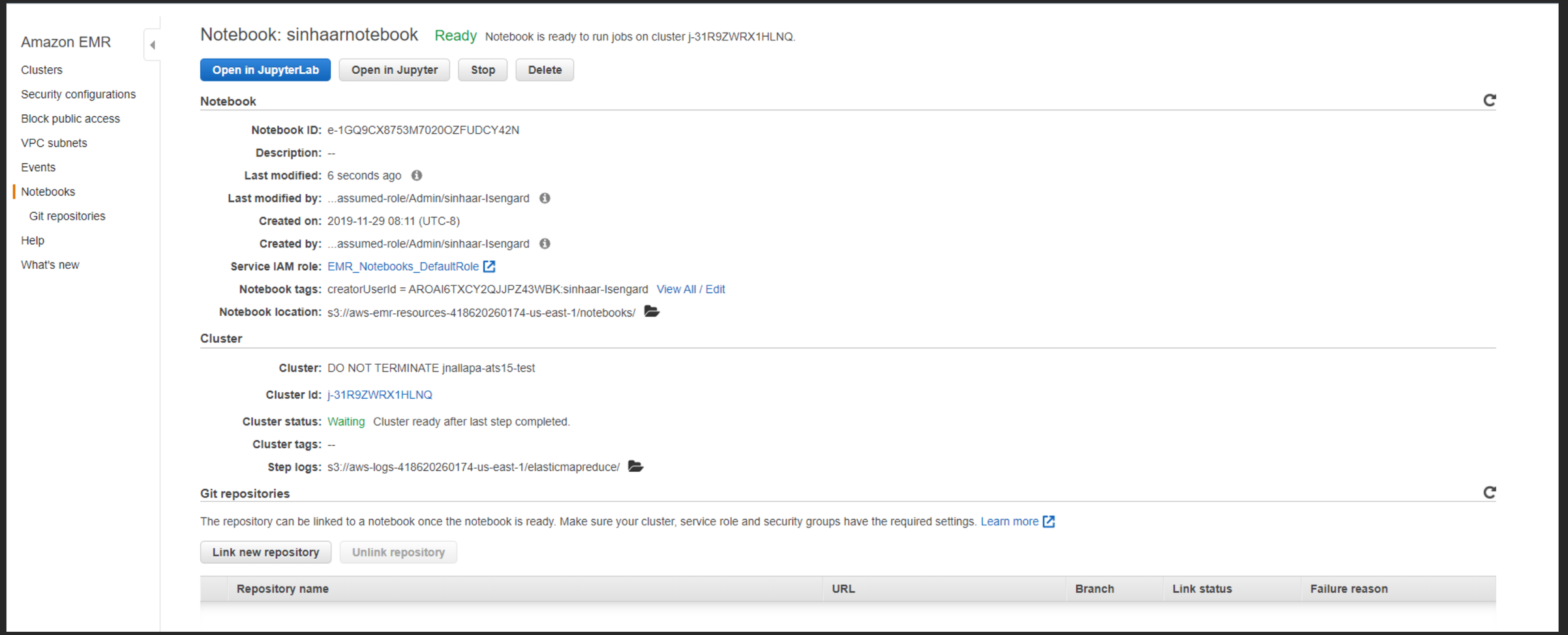
9

Managed notebooks with repository integration and notebook-scoped libraries

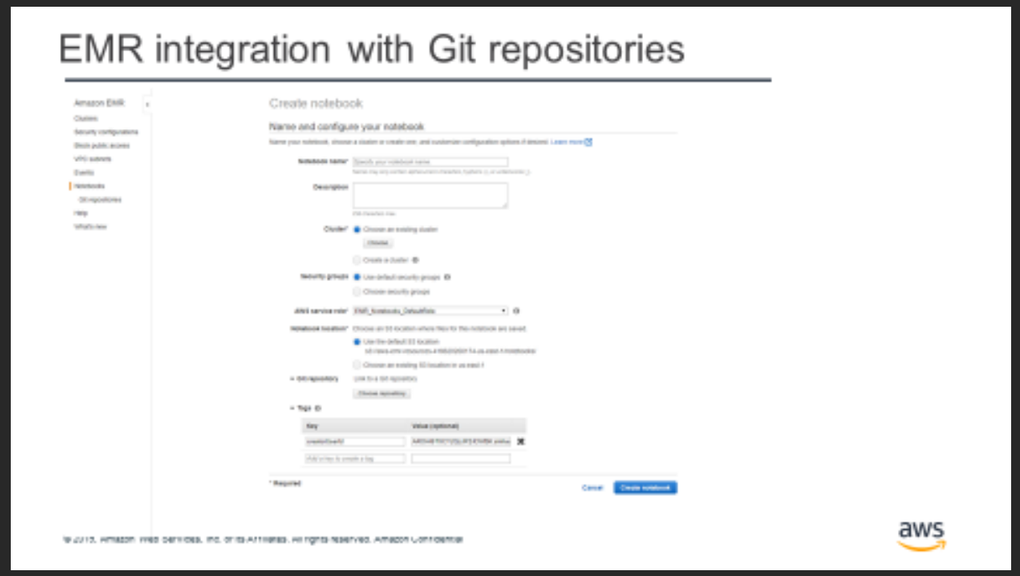
Fully managed notebooks for simplified experience

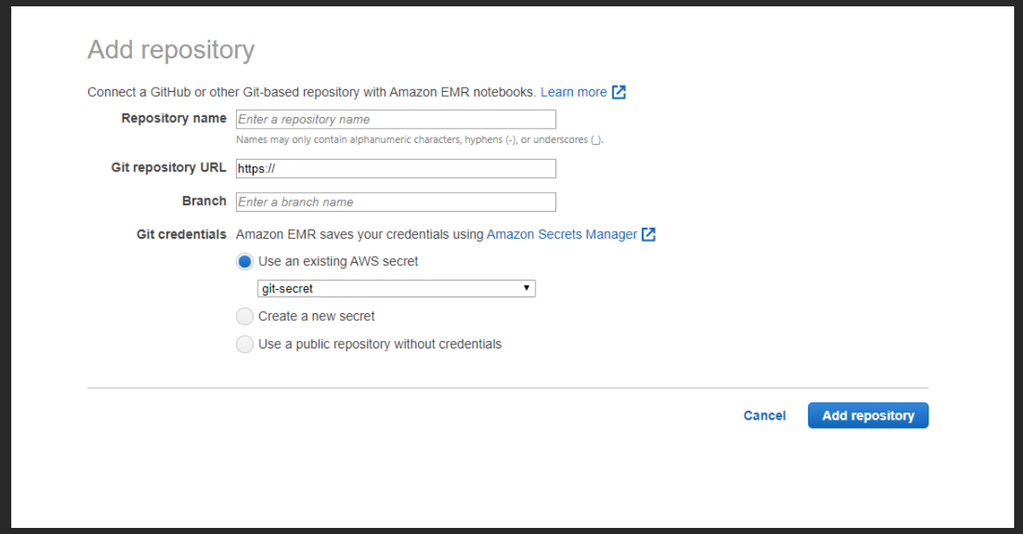


Open in Jupyter or JupyterLab



Associate notebooks with repositories





- Associate notebooks with repositories
- Collaborate and share notebooks
- Integrate with pipelines
- Integration with AWS Secrets Manager

Install notebook-scoped libraries management

- Notebook-scoped libraries are installed in a virtual Python environment
- Scoped to the current notebook session
- Available only during the notebook session
- Libraries are deleted after the session ends

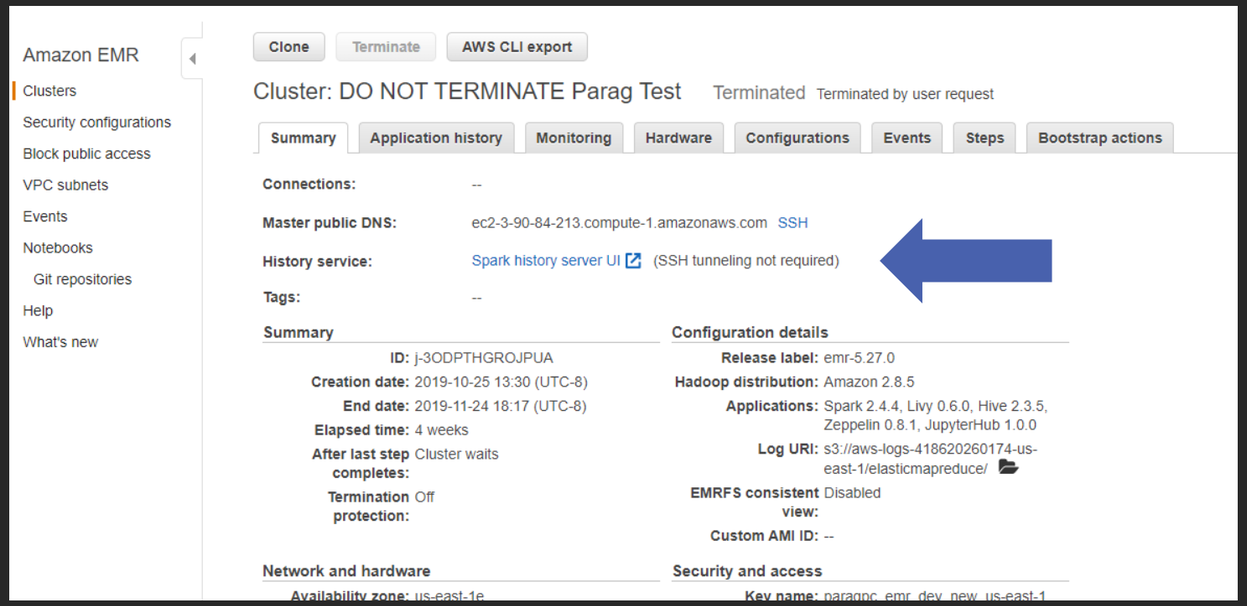
```
sc.install_pypi_package("celery")
import celery sc.range(1,10000,1,100).map(lambda x:celery.__version__).collect()
```

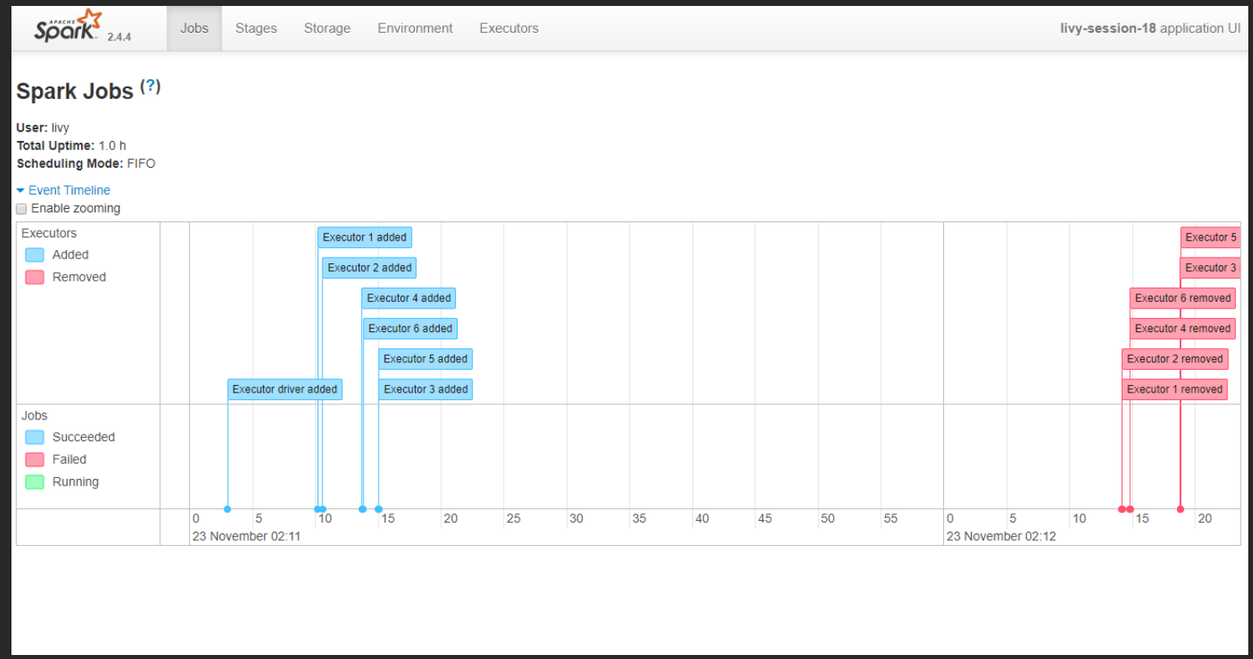
```
sc.install_pypi_package("arrow==0.14.0", "https://pypi.org/simple")
sc.uninstall_package("arrow")
```

10

Off-cluster persistent Spark History Service for simplified debugging

Off-cluster persistent Spark History Service

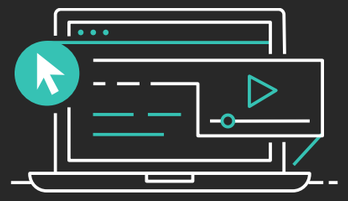




Demo

Learn big data with AWS Training and Certification

Resources created by the experts at AWS to help you build and validate data analytics skills



New free digital course, Data Analytics Fundamentals, introduces Amazon S3, Amazon Kinesis, Amazon EMR, AWS Glue, and Amazon Redshift

Classroom offerings, including Big Data on AWS, feature AWS expert instructors and hands-on labs

Validate expertise with the **AWS Certified Big Data - Specialty** exam or the new **AWS Certified Data Analytics - Specialty** beta exam

Visit aws.amazon.com/training/paths-specialty/

aws training and certification

Thank you!

Abhishek Sinha
sinhaar@amazon.com

Mert Hocanin
hocanint@amazon.com

AWS re:Invent

aws

Please complete the session survey in the mobile app.