CLOUD DAY 2024

improove

Milano, Nov 20

Building APIs for Big Data Services

Riccardo Venturi

Senior Solution Architect @ AWS

Kudos

CLOUD DAY 2024

Milano, Nov 20

improove

Akamai

aruba CLOUD

beSharp. we make IT run.

DATADOG

adesso.it

aws

EssilorLuxottica
Our Vision. Your Future.

software one

.Cleafy

CoNDENSE

Unprecedented Growth of Data

There is more data and more diversity of data than people think

Data growth

>10x

every 5 years

Data platforms needs

To live for

15+

years

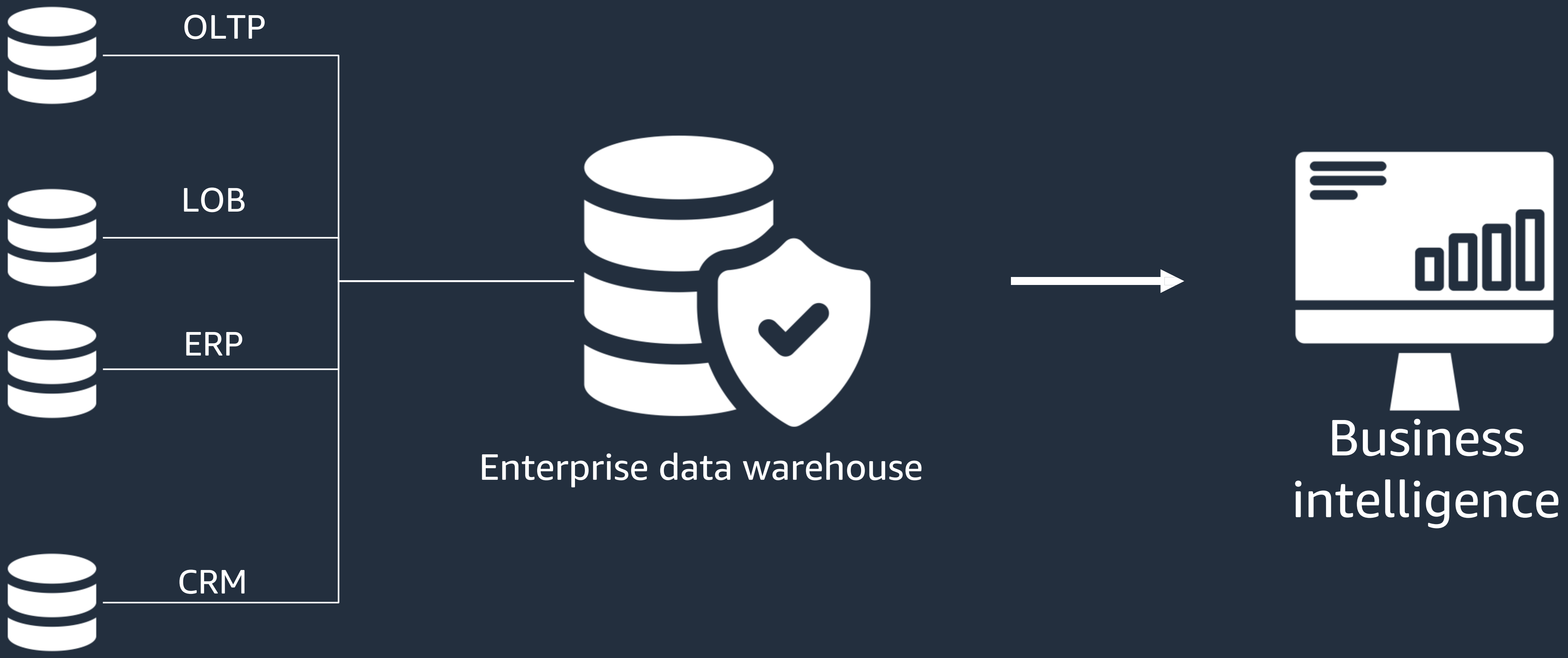
To scale

1,000x

IDC, "Data Age 2025"

Traditional Data Architecture

In the past, decision-making revolved around the **enterprise data warehouse**.



Diversity of Roles and applications

Data scientists

Business users

Analysts

Applications

Machine learning

SQL analytics

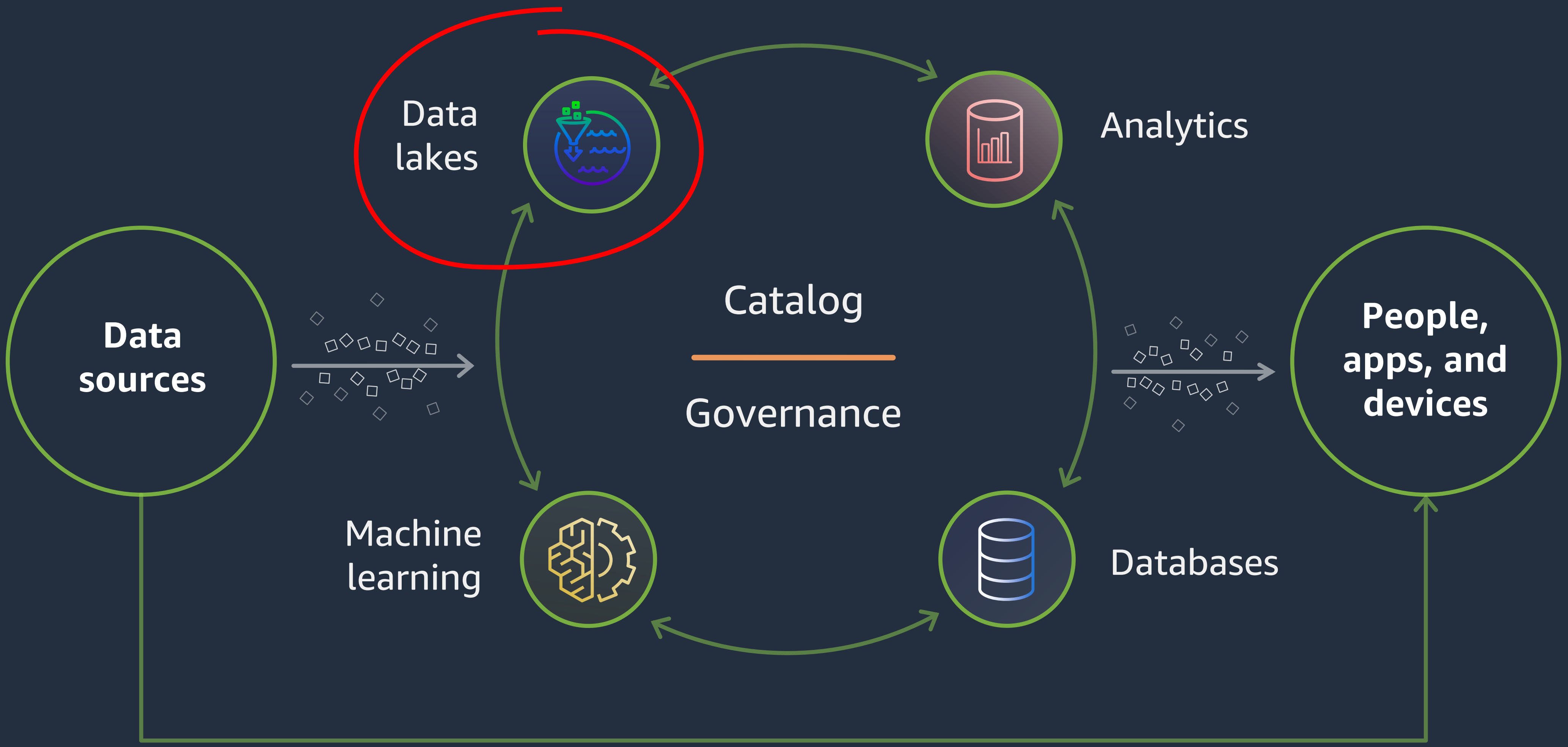
Scientific

Real-time, streaming

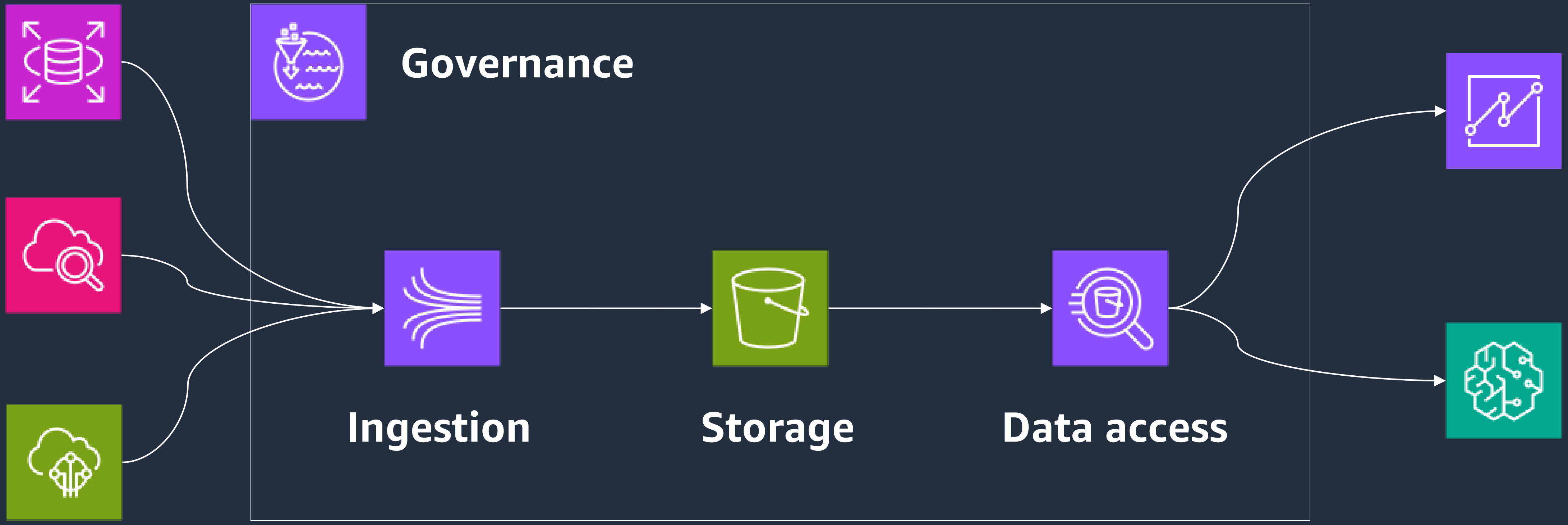
There are **more people** accessing data

And in **different ways**

Modern Data Architecture



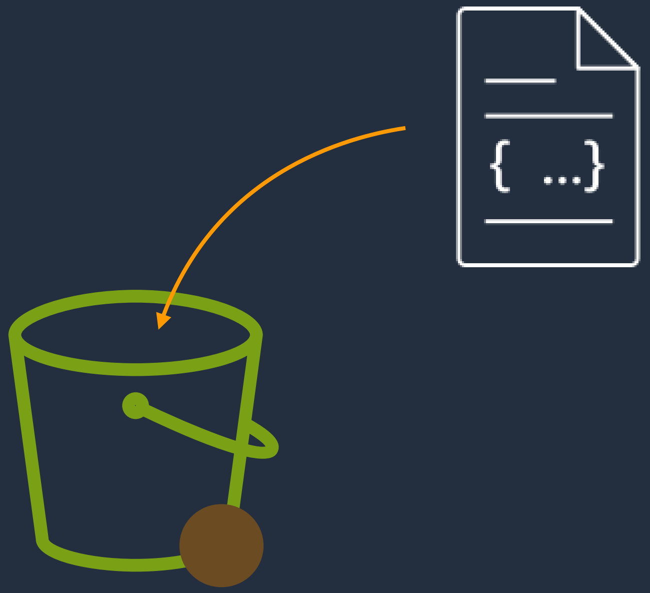
Data Lake Fundamentals



Data sources

Consumers

Ingestion Bucket

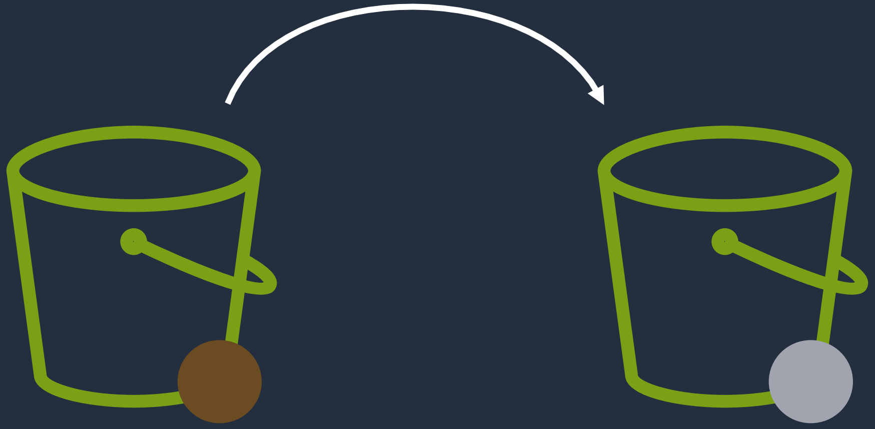


Bronze

Misc formats
(e.g. JSONL, CSV, PDF)

Raw data

Format Transformation



Bronze

Misc formats
(e.g. JSONL, CSV, PDF)

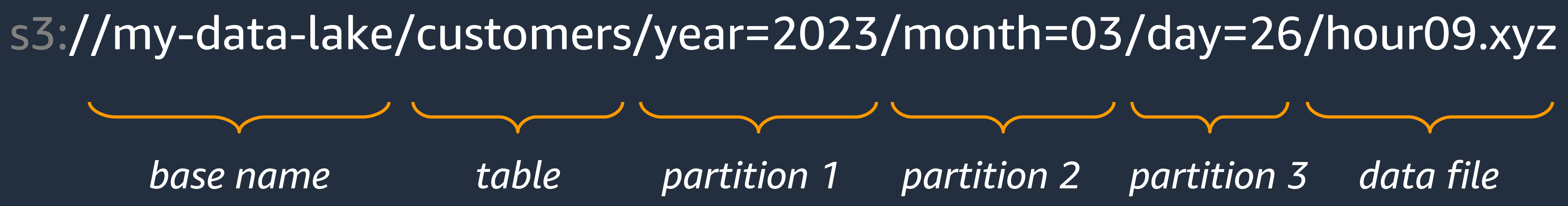
Raw data

Silver

Write Optimised
(e.g. Avro)

Curated data

Data Lake Conventions



s3://my-data-lake/customers/year=2023/month=03/day=26/hour09.xyz

Data Lake Formats

Avro

row-based

Parquet

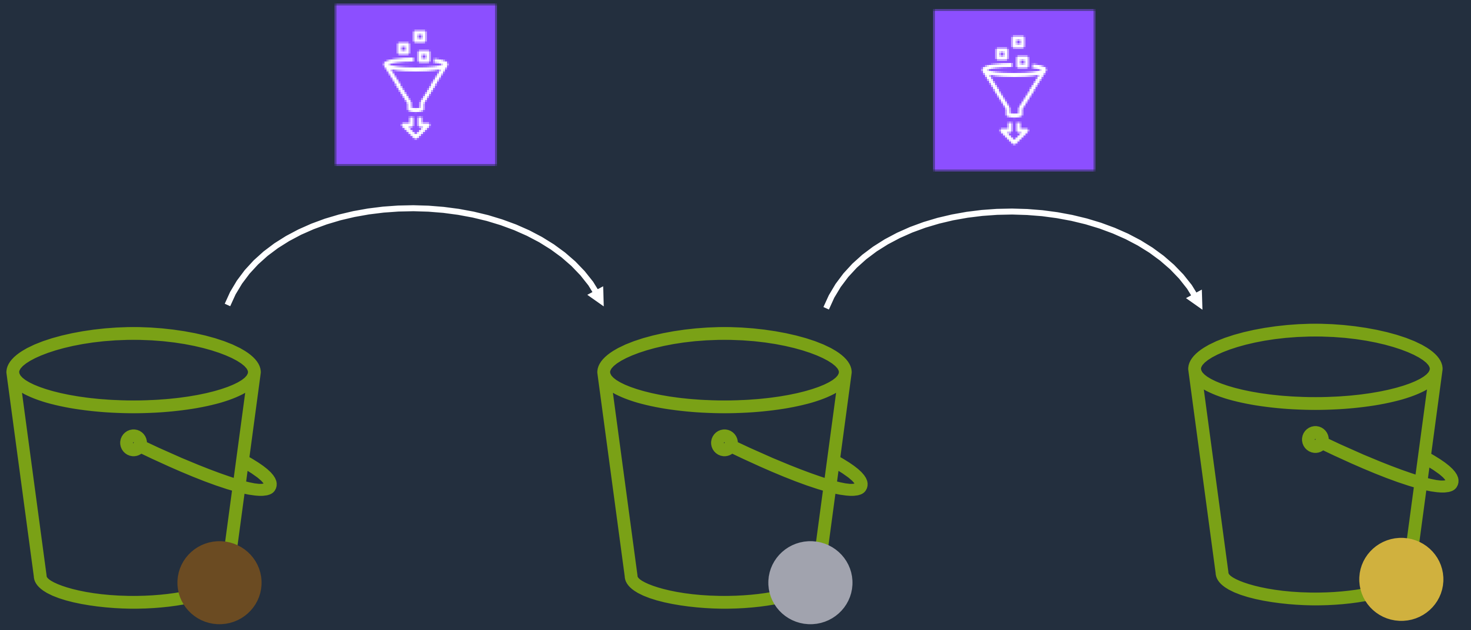
column-based

ICEBERG

transactional

Not in scope for this approach

Three-tier Data Lake



Bronze

Misc formats
(e.g. JSONL, CSV, PDF)

Raw data

Silver

Write Optimised
(e.g. Avro)

Curated data

Gold

Read Optimised
(e.g. Parquet)

Consumable data

Transformations

```
{ "id": "123", "user": { "id": "987", "name": "Jack
Johnson", "email": "jj@example.com", "city": "Aachen"
}, "items": [ {"p_id": "456", "qty": 1, "price":
37.81}, {"p_id": "567", "qty": 2, "price": 42.35} ],
"datetime": "2024-05-23T18:23:18Z" }
```

```
{ "id": "123", "user_id": "987", "user_city": "Aachen",
"order_total": 122.51, "items": [ {"p_id": "456",
"qty": 1, "price": 37.81}, {"p_id": "567", "qty": 2,
"price": 42.35} ], "datetime": "2024-05-23T18:23:18Z" }
```

AWS Glue

Cost Drivers

With Terabytes of total volumes and Gigabytes per transaction, be aware of typical cost drivers …

Compute

Storage

Data Transfer

Partitioning & Compression Examples

```
select count(*) from datalake where dt >= ‘20170515’ and dt < ‘20170516’
```

Partitioning	Size on S3	Run Time	Data Scanned	Cost
NO	74 GB	10.41 sec	74.1 GB	$0.36
YES	74 GB	2.73 sec	871.39 MB	$0.004
Result	**same**	**4x faster**	**85% less**	**98% cheaper**

```
Process all dataset
```

Compression	Size on S3	Run Time	Data Scanned	Cost
Text	1.15 TB	3m 56s	1.15 TB	$5.75
Parquet	130 GB	6.78s	2.51 GB	$0.013
Result	**87% less**	**34x faster**	**99% less**	**99.7% savings**

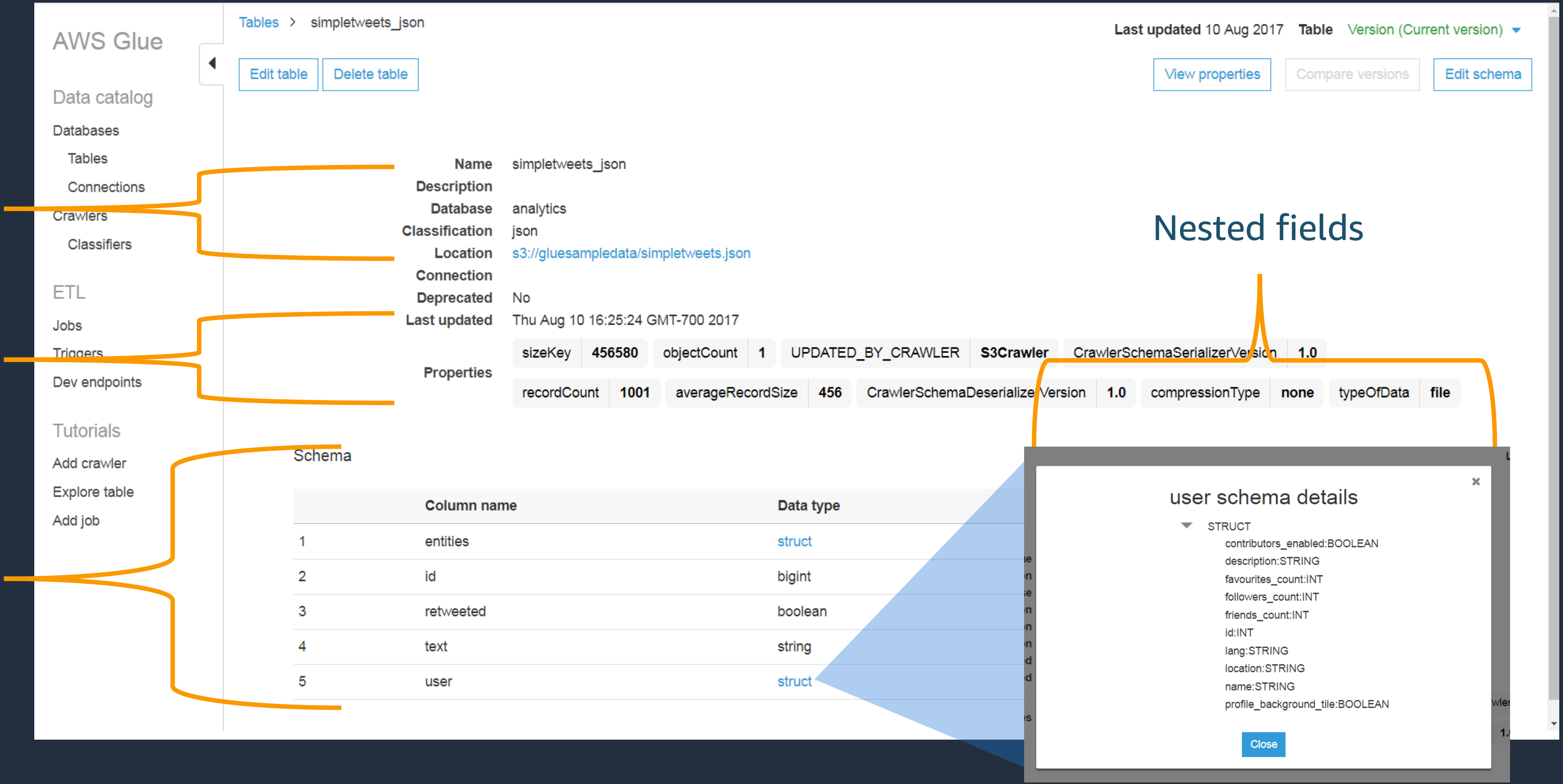
Metadata Catalog

Table properties

Data statistics

Table schema
& Partitions

Nested fields



Last updated 10 Aug 2017 **Table** Version (Current version)

Name simpletweets_json
Description
Database analytics
Classification json
Location s3://gluesampledata/simpletweets.json
Connection
Deprecated No
Last updated Thu Aug 10 16:25:24 GMT-700 2017
Properties sizeKey **456580** objectCount **1** UPDATED_BY_CRAWLER **S3Crawler** CrawlerSchemaSerializerVersion **1.0** recordCount **1001** averageRecordSize **456** CrawlerSchemaDeserializerVersion **1.0** compressionType **none** typeOfData **file**

Schema

	Column name	Data type
1	entities	struct
2	id	bigint
3	retweeted	boolean
4	text	string
5	user	struct

user schema details

STRUCT
contributors_enabled:BOOLEAN
description:STRING
favourites_count:INT
followers_count:INT
friends_count:INT
id:INT
lang:STRING
location:STRING
name:STRING
profile_background_tile:BOOLEAN

APIs and Query Semantics for Big Data

```
{ "id": "123", "user": { "id": "987", "name": "Jack Johnson", "email":
"jj@example.com", "city": "Aachen" }, "items": [ {"p_id": "456", "qty": 1, "price":
37.81}, {"p_id": "567", "qty": 2, "price": 42.35} ], "datetime": "2024-05-
23T18:23:18Z" }

{ "id": "124", "user": { "id": "988", "name": "John Jackson", "email":
"jj@example.net", "city": ”Milan" }, "items": [ { "p_id": "678", "qty": 3, "price":
43.19 }, { "p_id": "789", "qty": 25, "price": 2.88 } ], "time": "2024-11-
20T09:43:10Z" }

…
```

Task: “Get orders from Milan in Nov. 2024 or later”

Query Semantics: REST

/orders?customerCity=Berlin&orderDateFrom=2023-03-01

Query Semantics: OData

```
/orders?$filter=City eq 'Berlin'&$expand=Orders($filter=OrderDate ge 2023-03-01;$expand=OrderItems)
```

Query Semantics: GraphQL

```
query {
  customers(filter: { city: "Berlin" }) {
    id
    name
    orders(filter: { datetime: { gte: "2023-03-01" } }) {
      id
      date
      orderItems {
        p_id
        qty
        price
      }
    }
  }
}
```

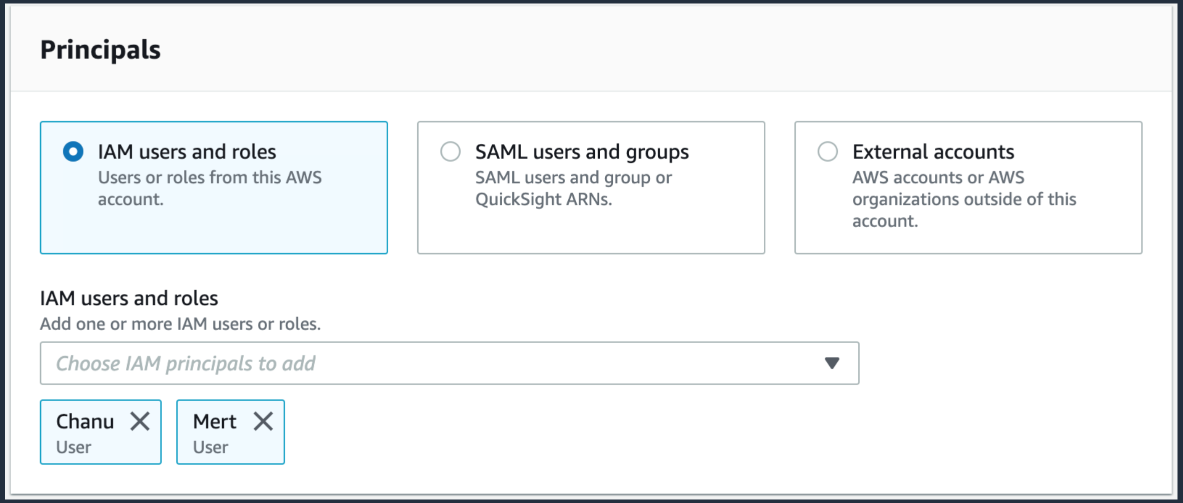
Query Semantics: SQL

```
SELECT c.*, o.*, oi.*
FROM customers c
JOIN orders o ON c.customer_id = o.customer_id
JOIN order_items oi ON o.order_id = oi.order_id
WHERE c.city = 'Berlin'
   AND o.order_date >= DATE '2023-03-01'
```

trino

Amazon
Athena

Data API Security



Principals

● IAM users and roles
Users or roles from this AWS account.

○ SAML users and groups
SAML users and group or QuickSight ARNs.

○ External accounts
AWS accounts or AWS organizations outside of this account.

IAM users and roles
Add one or more IAM users or roles.

Choose IAM principals to add ▼

Chanu ✕
User

Mert ✕
User

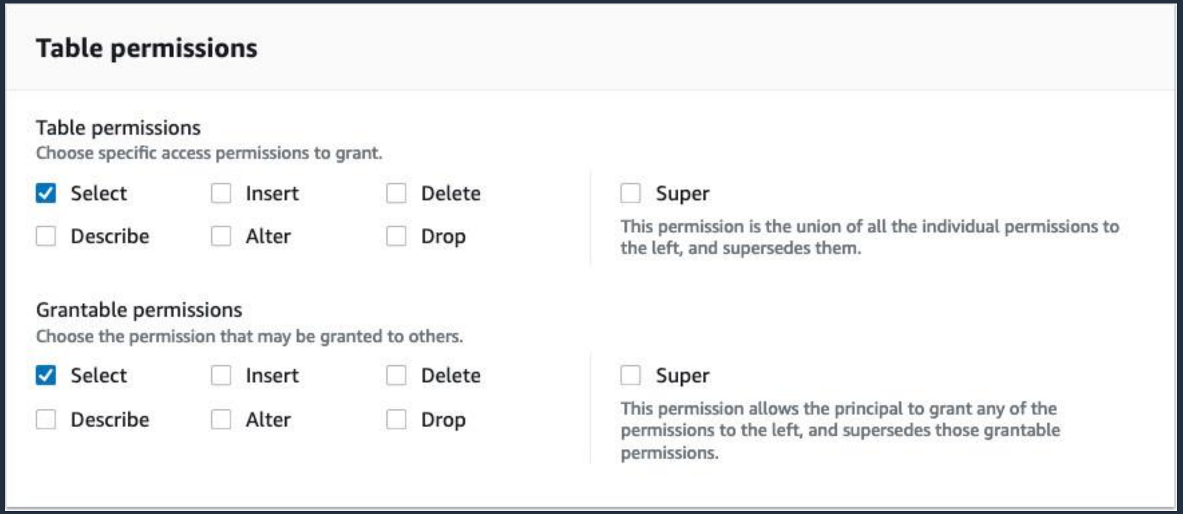


Table permissions

Table permissions
Choose specific access permissions to grant.

☑ Select □ Insert □ Delete
□ Describe □ Alter □ Drop

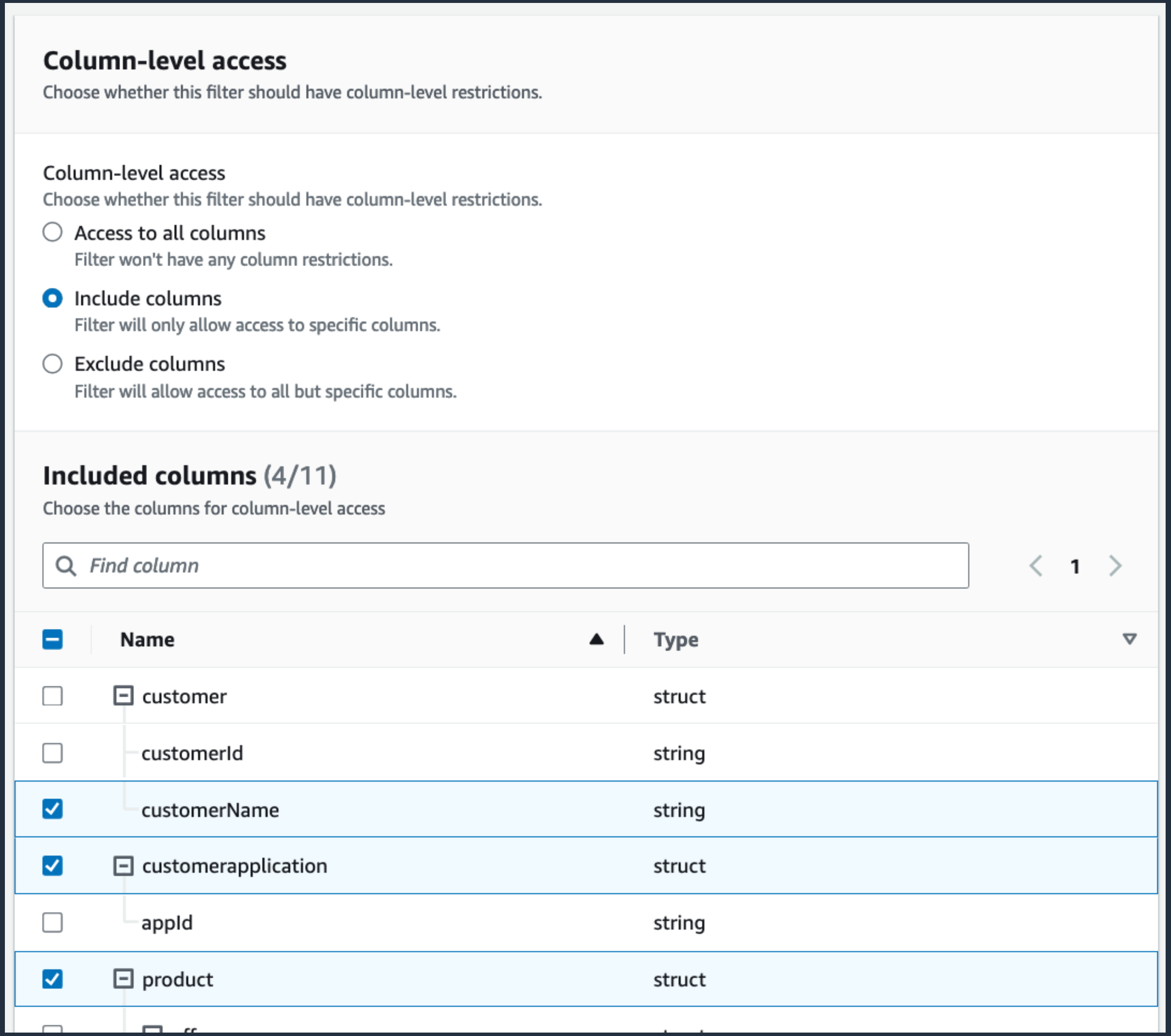
□ Super
This permission is the union of all the individual permissions to the left, and supersedes them.

Grantable permissions
Choose the permission that may be granted to others.

☑ Select □ Insert □ Delete
□ Describe □ Alter □ Drop

□ Super
This permission allows the principal to grant any of the permissions to the left, and supersedes those grantable permissions.

Column-level Access



Column-level access

Choose whether this filter should have column-level restrictions.

Column-level access

Choose whether this filter should have column-level restrictions.

○ Access to all columns
Filter won't have any column restrictions.

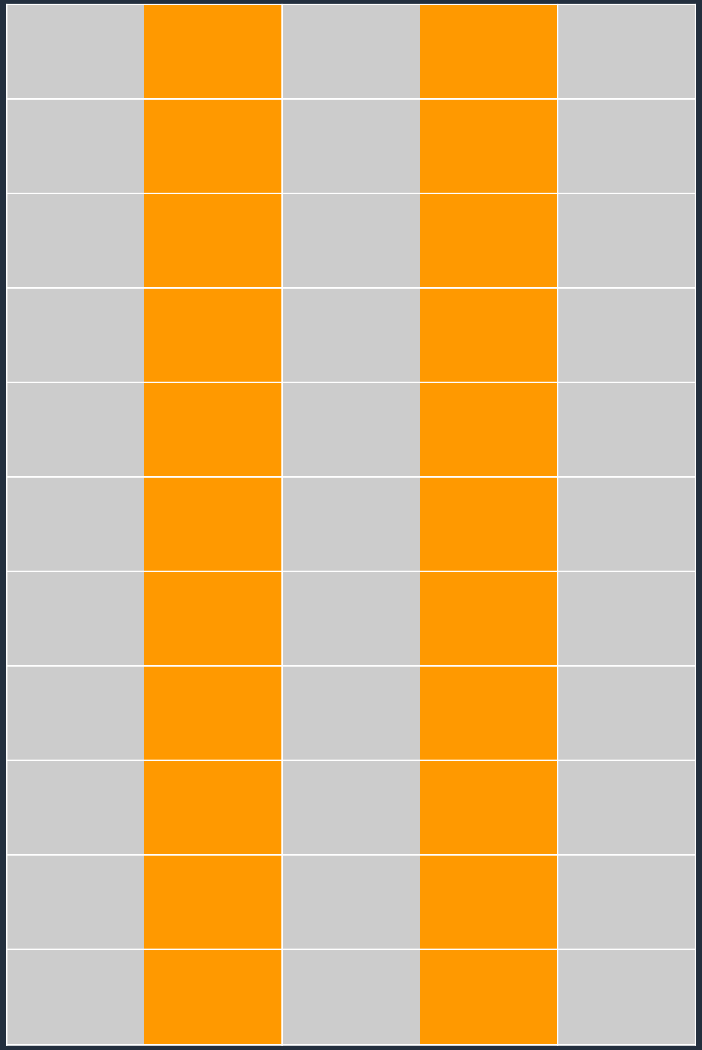
● Include columns
Filter will only allow access to specific columns.

○ Exclude columns
Filter will allow access to all but specific columns.

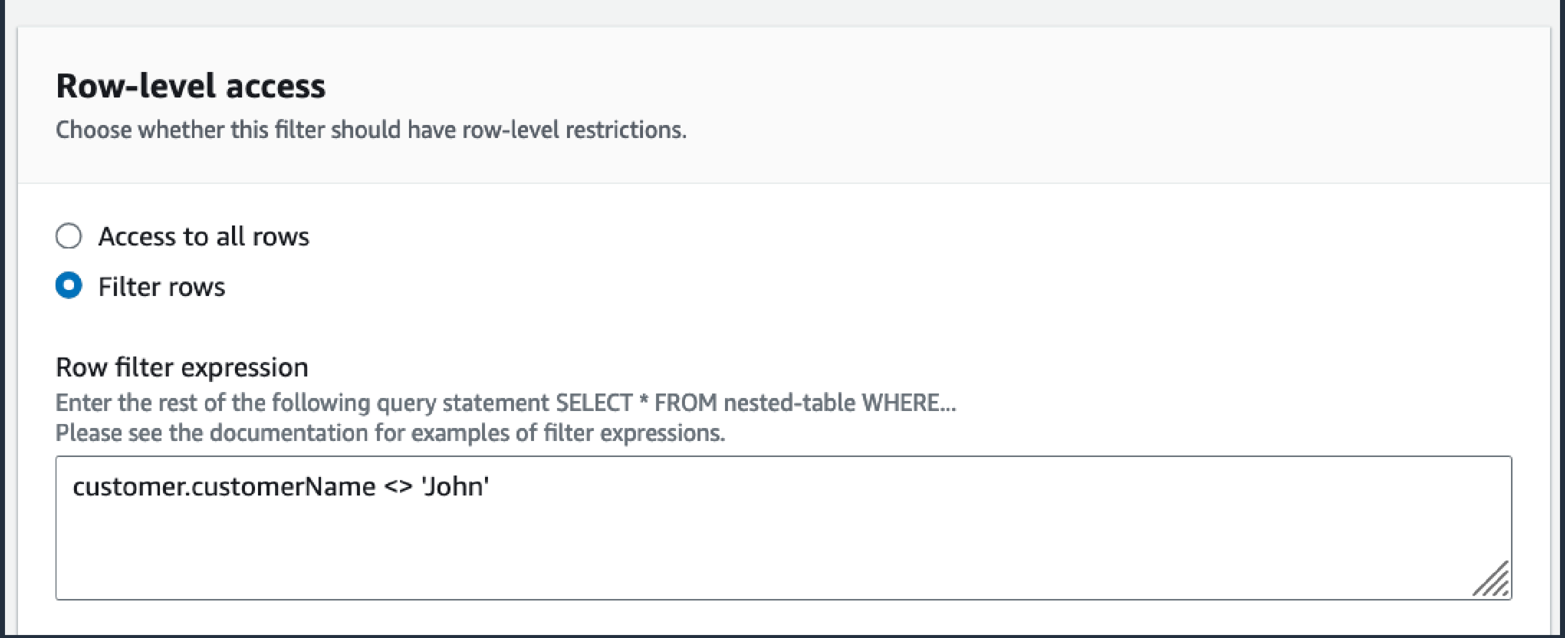
Included columns (4/11)

Choose the columns for column-level access

☐	Name	Type
☐	customer	struct
☐	customerId	string
☑	customerName	string
☑	customerapplication	struct
☐	appId	string
☑	product	struct



Row-level Access



Row-level access

Choose whether this filter should have row-level restrictions.

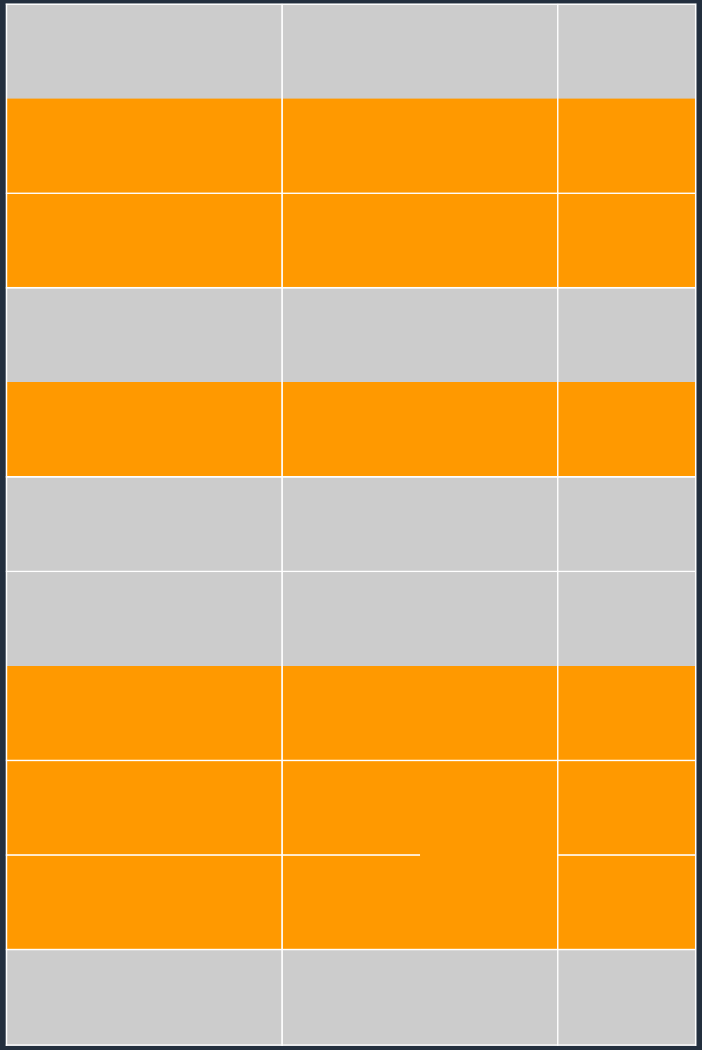
○ Access to all rows

● Filter rows

Row filter expression

Enter the rest of the following query statement SELECT * FROM nested-table WHERE...
Please see the documentation for examples of filter expressions.

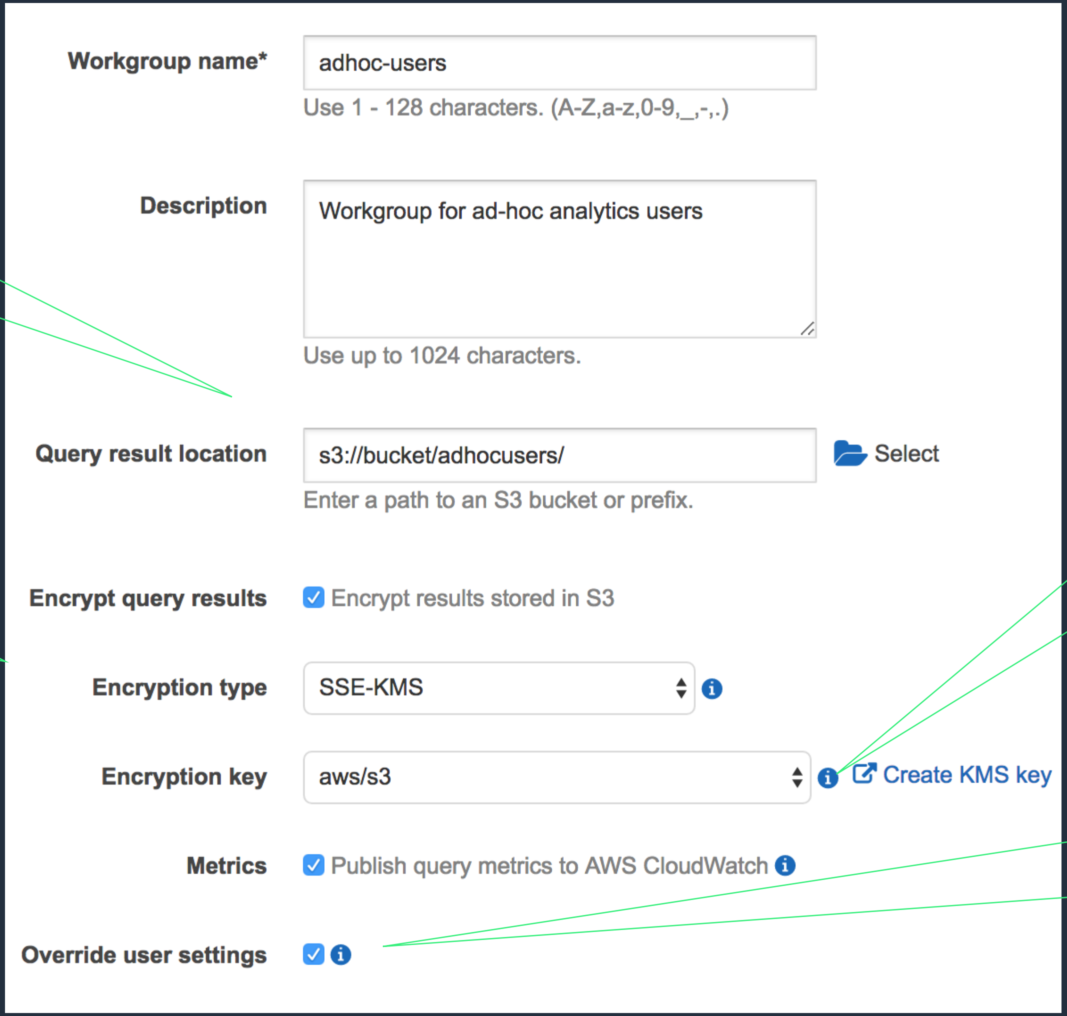
```
customer.customerName <> 'John'
```



Workloads Control - Athena Workgroups

Unique query output location per Workgroup

Encrypt results with unique AWS KMS key per Workgroup

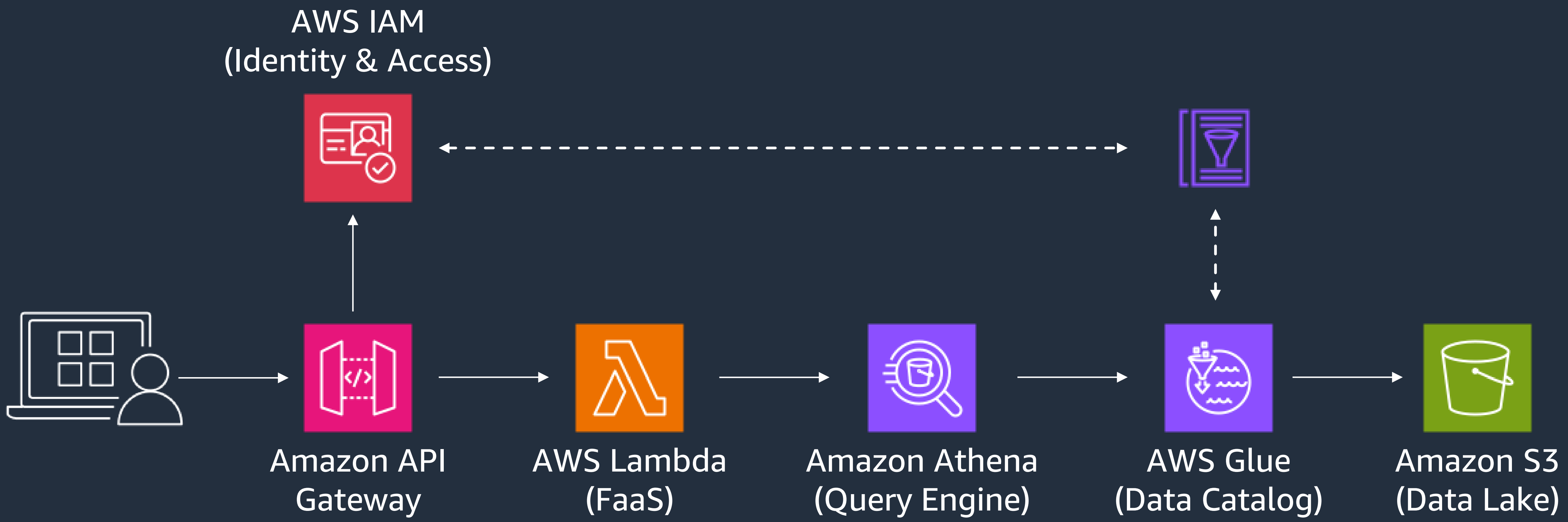


Workgroup name*	adhoc-users Use 1 - 128 characters. (A-Z,a-z,0-9,_,-,.)
Description	Workgroup for ad-hoc analytics users Use up to 1024 characters.
Query result location	s3://bucket/adhocusers/ Select Enter a path to an S3 bucket or prefix.
Encrypt query results	☑ Encrypt results stored in S3
Encryption type	SSE-KMS
Encryption key	aws/s3 Create KMS key
Metrics	☑ Publish query metrics to AWS CloudWatch
Override user settings	☑

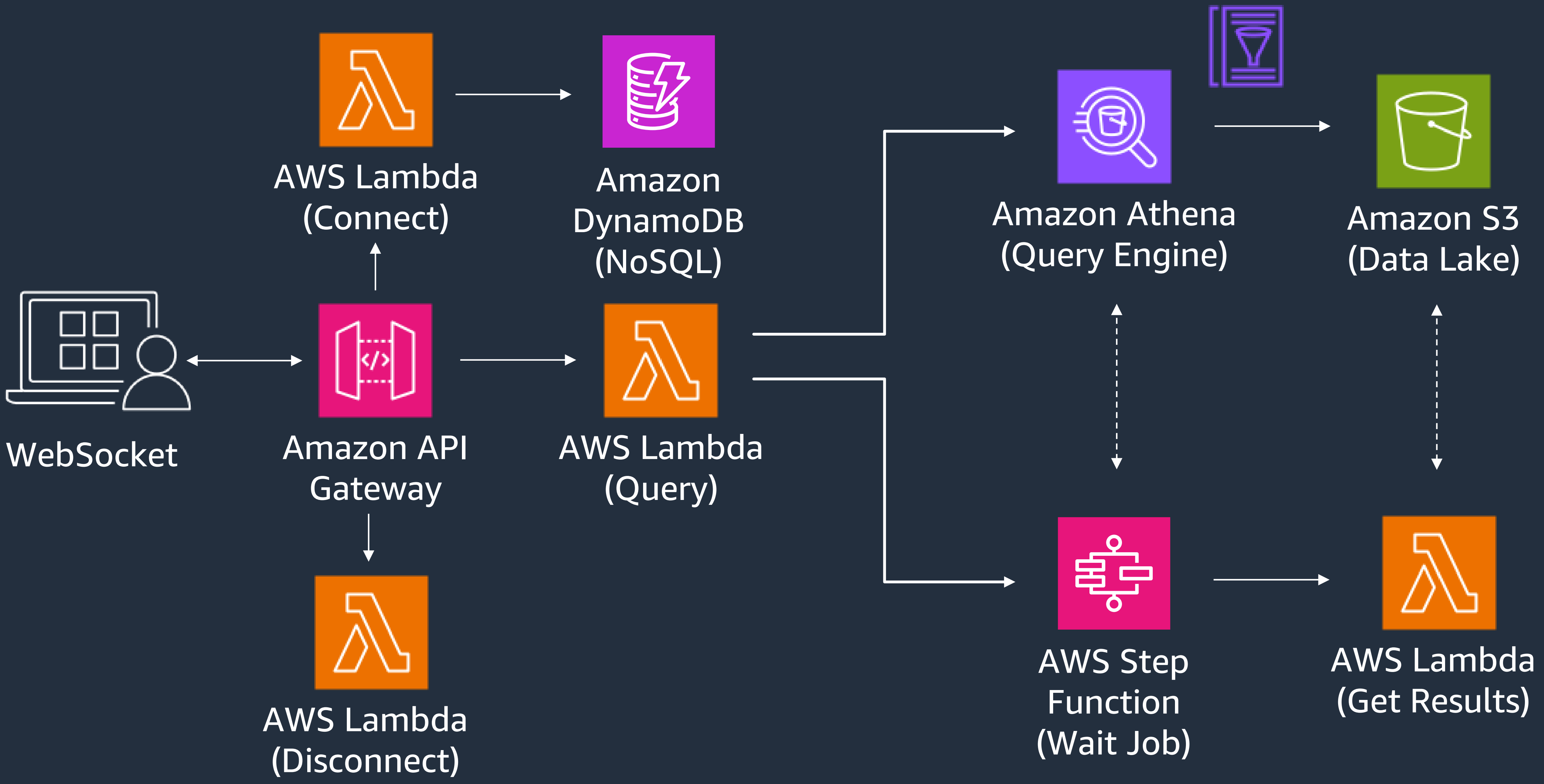
Collect and publish aggregated metrics per Workgroup to AWS CloudWatch

Use Workgroup settings eliminating need to configure individual users

Example 1: Sync Query API



Example 2: Async Query API



WebSocket

Amazon API Gateway

AWS Lambda (Connect)

Amazon DynamoDB (NoSQL)

AWS Lambda (Disconnect)

AWS Lambda (Query)

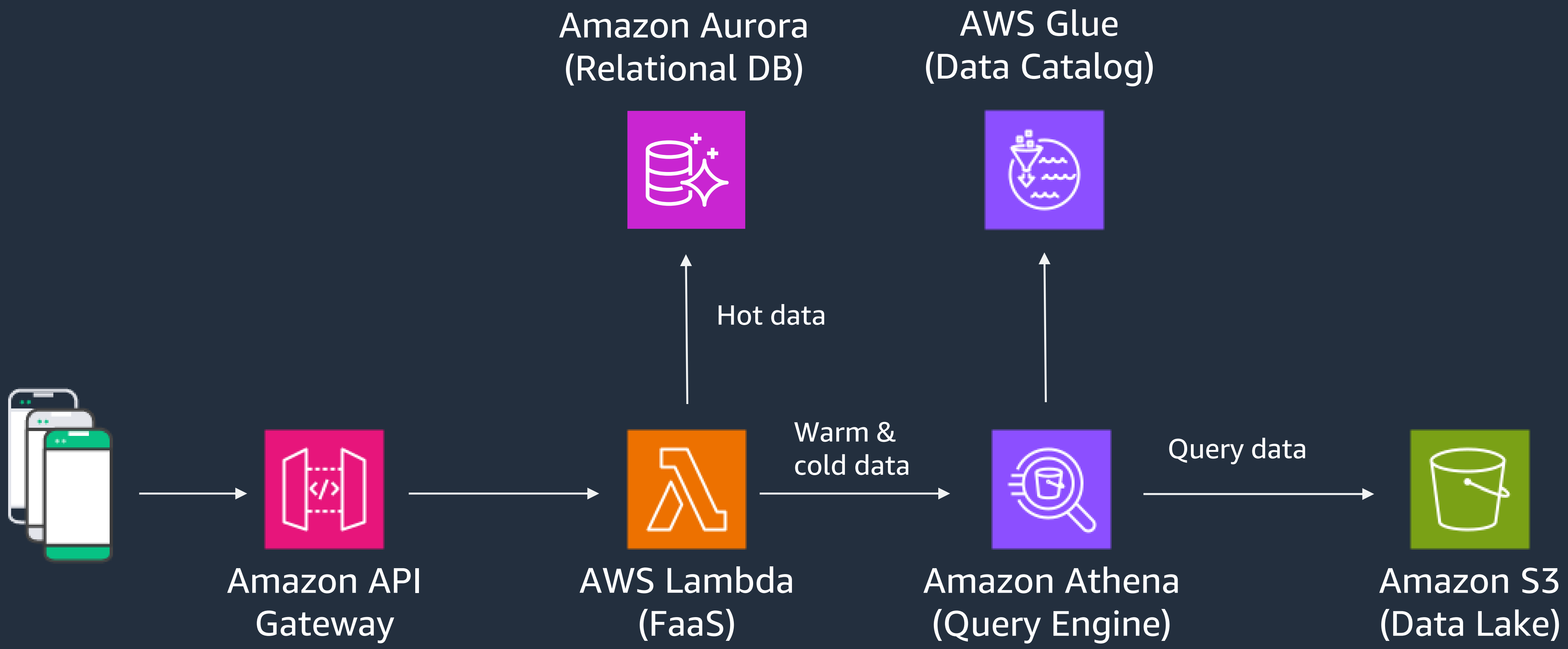
Amazon Athena (Query Engine)

Amazon S3 (Data Lake)

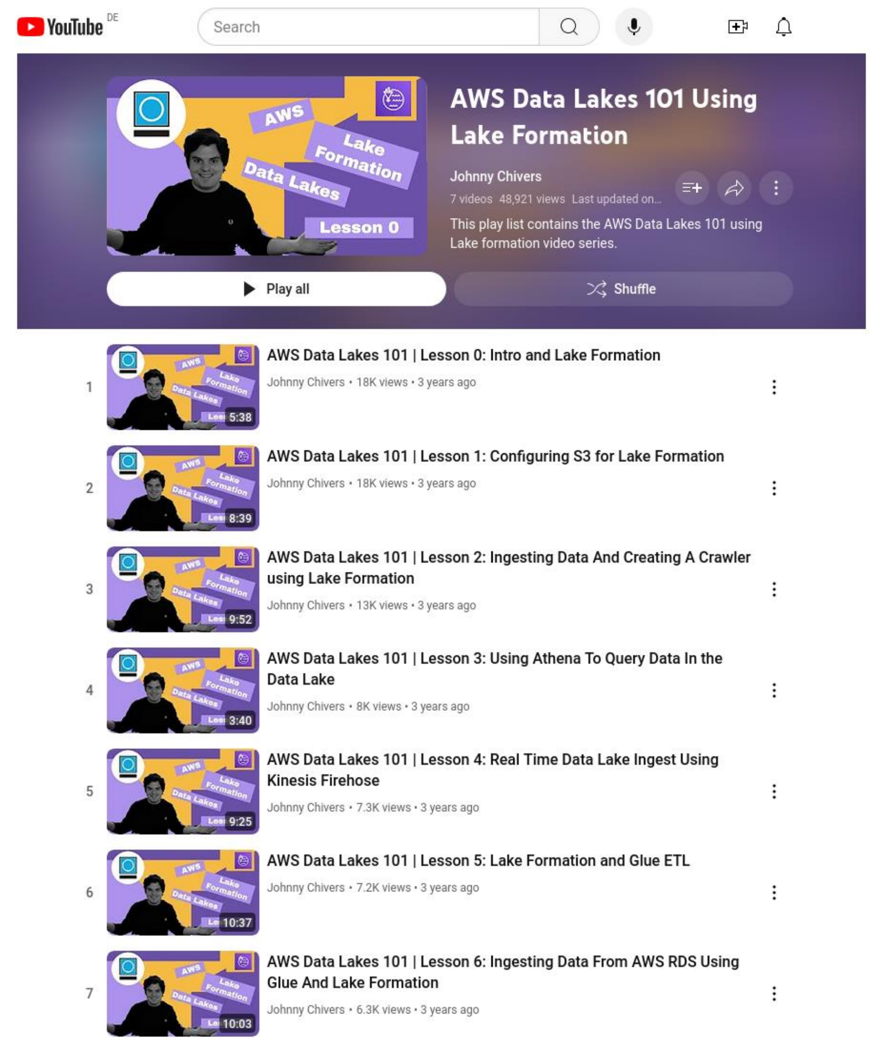
AWS Step Function (Wait Job)

AWS Lambda (Get Results)

Example 3: Data Temperature Routing



Do it Yourself!



AWS Data Lakes 101 Using Lake Formation

Johnny Chivers

7 videos 48,921 views Last updated on...

This play list contains the AWS Data Lakes 101 using Lake formation video series.

1 AWS Data Lakes 101 | Lesson 0: Intro and Lake Formation
Johnny Chivers • 18K views • 3 years ago
5:38

2 AWS Data Lakes 101 | Lesson 1: Configuring S3 for Lake Formation
Johnny Chivers • 18K views • 3 years ago
8:39

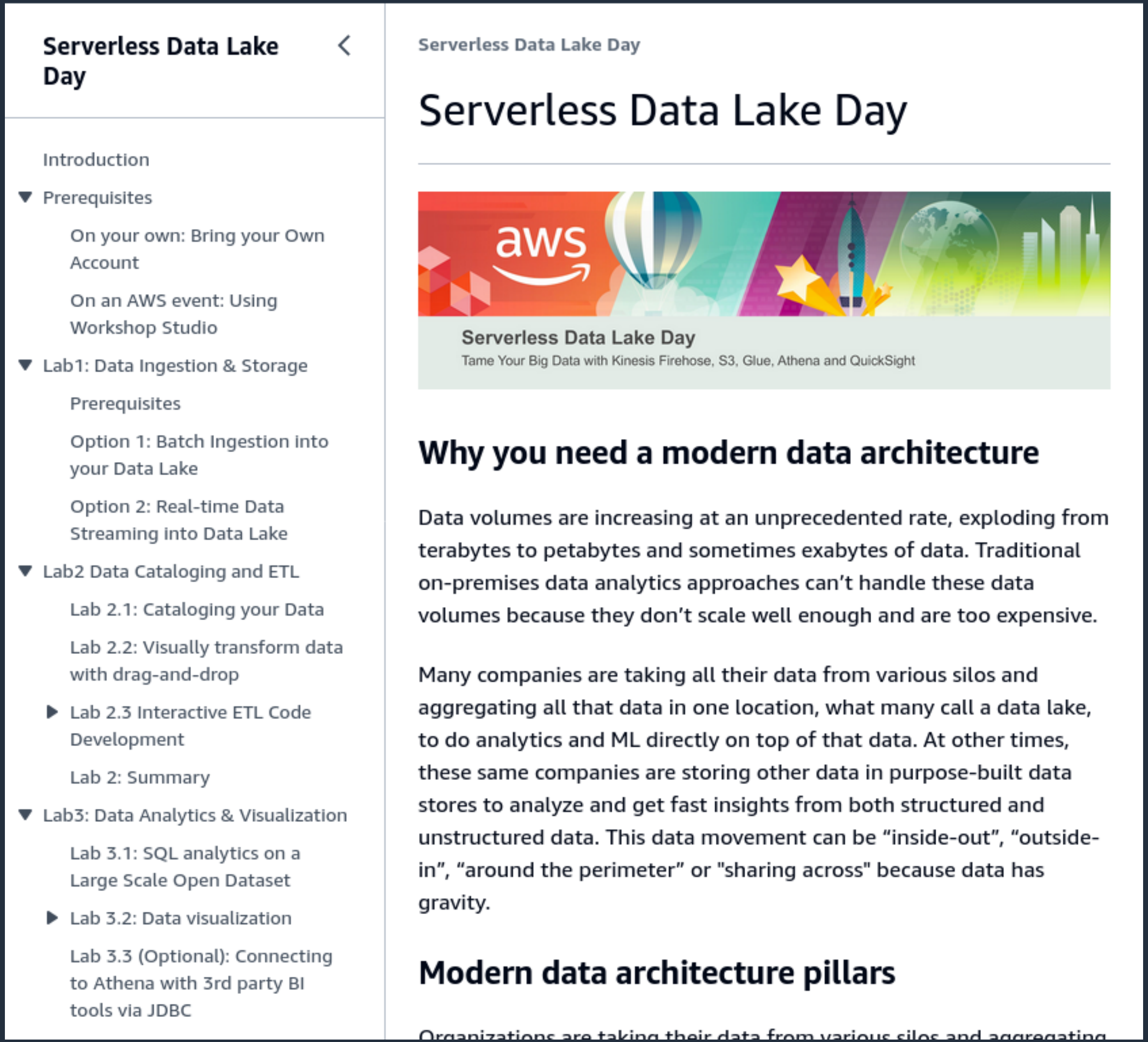
3 AWS Data Lakes 101 | Lesson 2: Ingesting Data And Creating A Crawler using Lake Formation
Johnny Chivers • 13K views • 3 years ago
9:52

4 AWS Data Lakes 101 | Lesson 3: Using Athena To Query Data In the Data Lake
Johnny Chivers • 8K views • 3 years ago
3:40

5 AWS Data Lakes 101 | Lesson 4: Real Time Data Lake Ingest Using Kinesis Firehose
Johnny Chivers • 7.3K views • 3 years ago
9:25

6 AWS Data Lakes 101 | Lesson 5: Lake Formation and Glue ETL
Johnny Chivers • 7.2K views • 3 years ago
10:37

7 AWS Data Lakes 101 | Lesson 6: Ingesting Data From AWS RDS Using Glue And Lake Formation
Johnny Chivers • 6.3K views • 3 years ago
10:03


aws
Serverless Data Lake Day
Tame Your Big Data with Kinesis Firehose, S3, Glue, Athena and QuickSight

Serverless Data Lake Day

- Introduction
- ▼ Prerequisites
  - On your own: Bring your Own Account
  - On an AWS event: Using Workshop Studio
- ▼ Lab1: Data Ingestion & Storage
  - Prerequisites
  - Option 1: Batch Ingestion into your Data Lake
  - Option 2: Real-time Data Streaming into Data Lake
- ▼ Lab2 Data Cataloging and ETL
  - Lab 2.1: Cataloging your Data
  - Lab 2.2: Visually transform data with drag-and-drop
  - ▶ Lab 2.3 Interactive ETL Code Development
  - Lab 2: Summary
- ▼ Lab3: Data Analytics & Visualization
  - Lab 3.1: SQL analytics on a Large Scale Open Dataset
  - ▶ Lab 3.2: Data visualization
  - Lab 3.3 (Optional): Connecting to Athena with 3rd party BI tools via JDBC

Serverless Data Lake Day

Serverless Data Lake Day

Why you need a modern data architecture

Data volumes are increasing at an unprecedented rate, exploding from terabytes to petabytes and sometimes exabytes of data. Traditional on-premises data analytics approaches can't handle these data volumes because they don't scale well enough and are too expensive.

Many companies are taking all their data from various silos and aggregating all that data in one location, what many call a data lake, to do analytics and ML directly on top of that data. At other times, these same companies are storing other data in purpose-built data stores to analyze and get fast insights from both structured and unstructured data. This data movement can be "inside-out", "outside-in", "around the perimeter" or "sharing across" because data has gravity.

Modern data architecture pillars

Organizations are taking their data from various silos and aggregating

Additional resources

All resources can be found here:
https://de3n2axe8vokl.cloudfront.net

- 📑 Presentation Slides
- 🔗 Blog Links
- 📚 Suggested Training Material
- 📊 Survey

Linkedin @venturir

Thanks!

Riccardo Venturi

Senior Solution Architect @ AWS

CLOUD DAY 2024

improove

Milano, Nov 20