



# AWS Data Lake

Syed Jaffry  
AWS Solutions Architect



# Agenda

1. What are data lakes?
2. Data lake optimisation patterns
3. What's hard today in building data lakes?
4. AWS Lake Formation + Demo

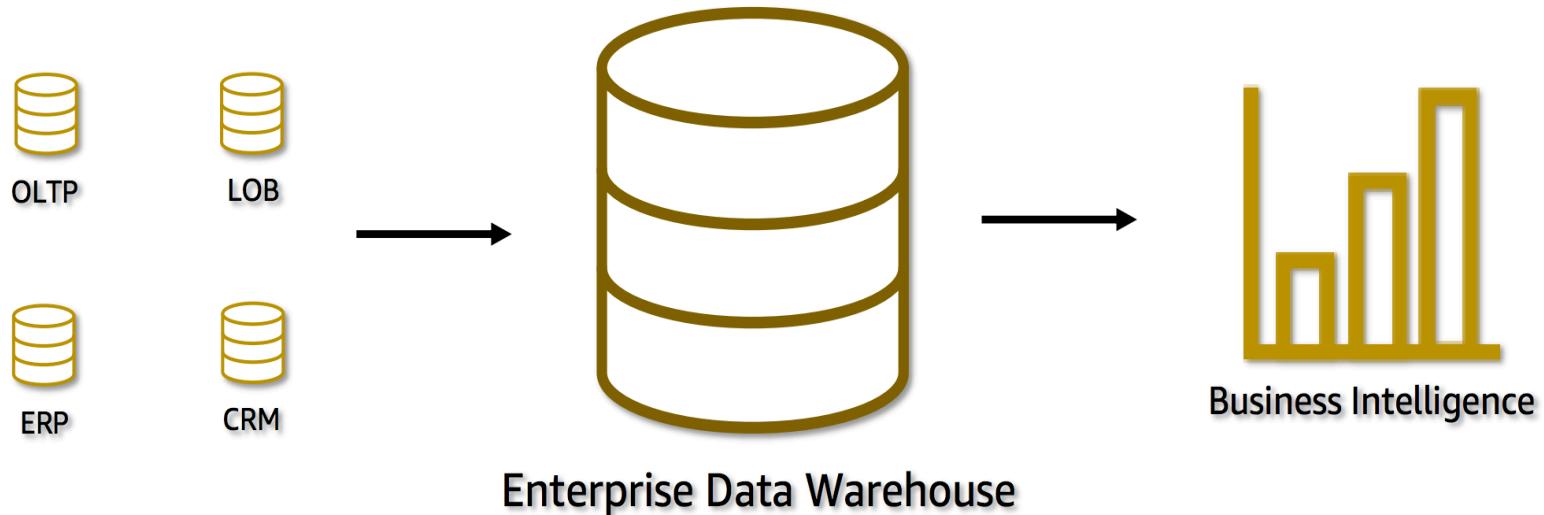
# Contact Me!

Syed Jaffry

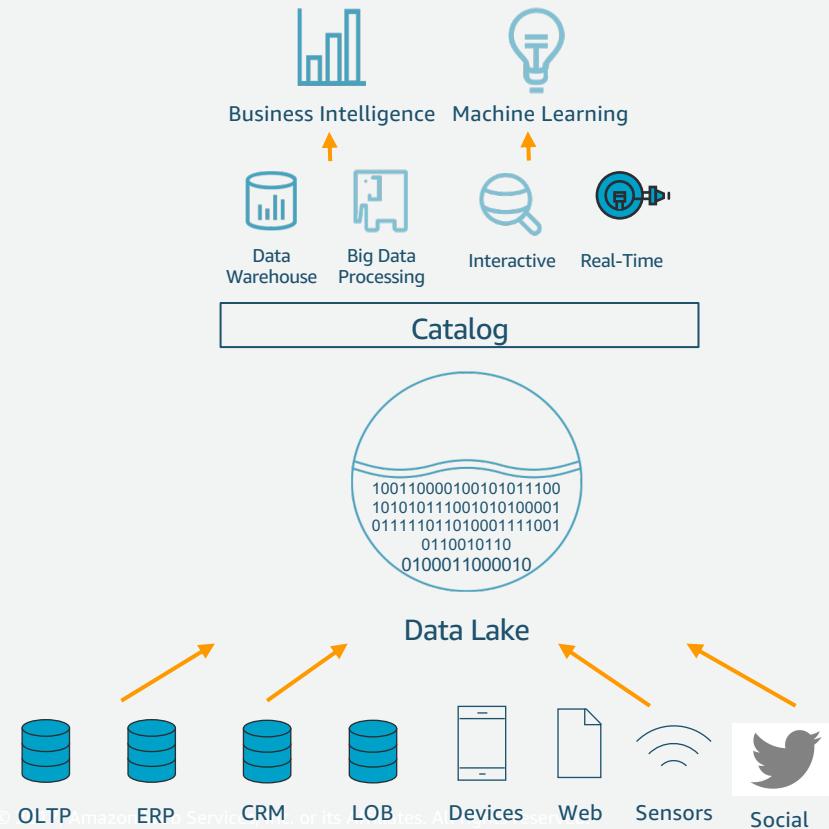
[syejaffr@amazon.com](mailto:syejaffr@amazon.com)

[linkedin.com/in/syed-jaffry-4751114](https://linkedin.com/in/syed-jaffry-4751114)

... revolve around the **Enterprise Data Warehouse** (in the 90s – 00s)



# Data Lakes Extend the Traditional Approach



Centralized Repository

Structured and Unstructured data

TB-EBs Scale

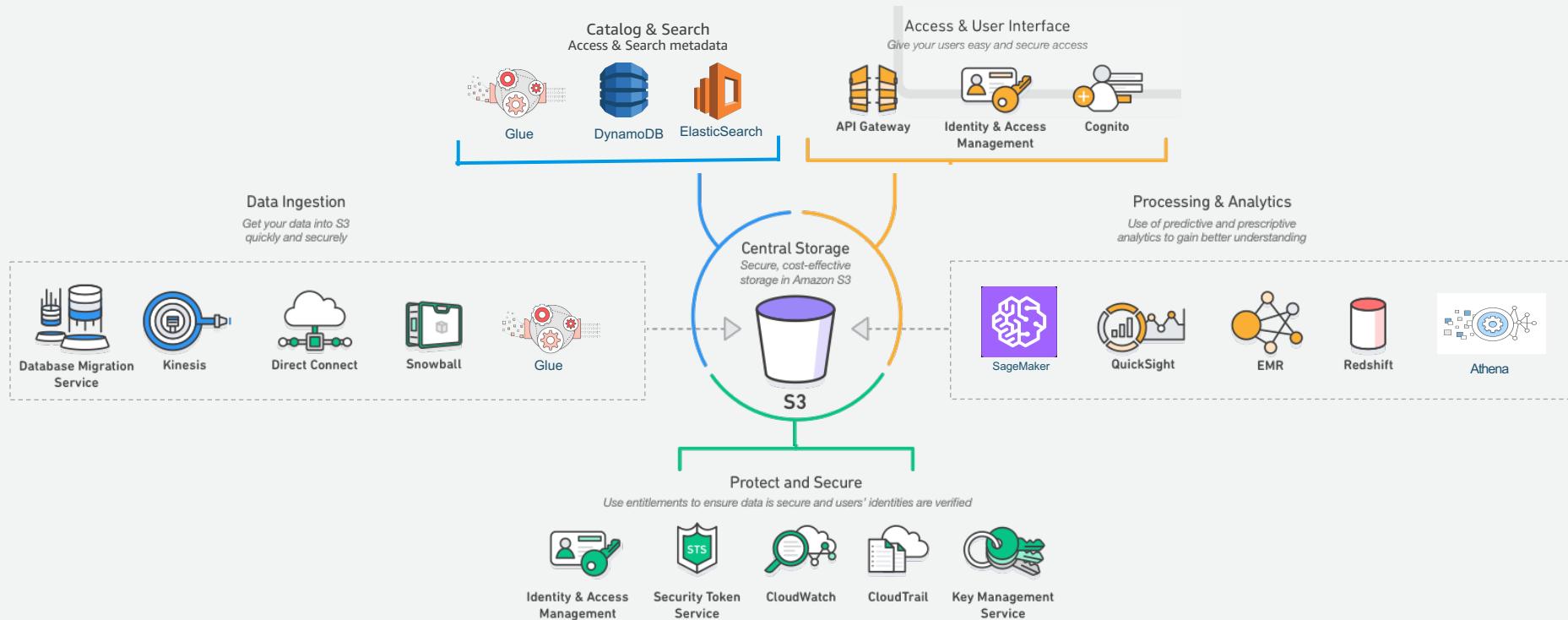
Separation of Storage and Compute

Schema on Read

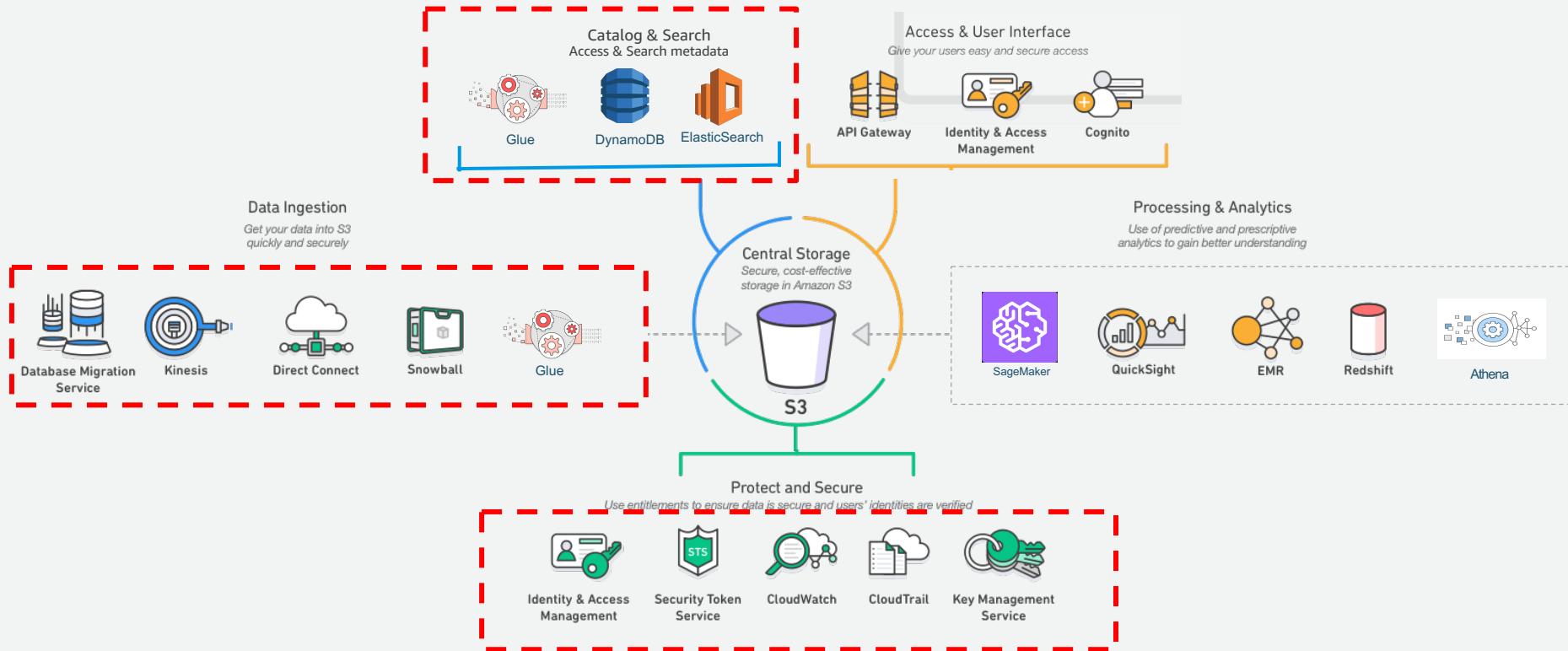
Diverse Analytical Engines



# Data Lake on AWS



# Data Lake on AWS



# Data Lake Optimisation

# Optimizing for Cost and Performance

```
/user/hive/warehouse/logs  
└── dt=2001-01-01/  
    ├── country=GB/  
    │   └── file1  
    │   └── file2  
    └── country=US/  
        └── file3  
└── dt=2001-01-02/  
    ├── country=GB/  
    │   └── file4  
    └── country=US/  
        └── file5  
        └── file6
```

## Partitioning

Pay for data your **query needs**,  
not to scan **all** of your data



## Compression

Pay for what you **store**,  
not for what you **process**



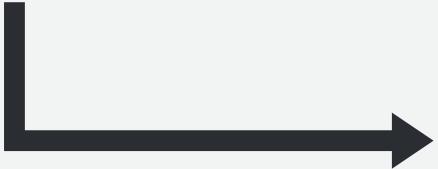
## Managed Services

Pay for what you **use**,  
not for what you **run**

# Partitioning

# Partitioning

```
datalake
└── 20170515T1423-GB-01.tar.gz
└── 20170515T1423-GB-02.tar.gz
└── 20170515T1500-US-01.tar.gz
└── 20170516T1500-US-01.tar.gz
└── 20170516T1600-GB-01.tar.gz
└── 20170516T1600-GB-02.tar.gz
```



```
select * from datalake where
dt=20170515 and country=US
```

```
datalake
└── dt=20170515
    └── country=GB
        └── 20170515T1423-GB-01.tar.gz
        └── 20170515T1423-GB-02.tar.gz
    └── country=US
        └── 20170515T1500-US-01.tar.gz
└── dt=20170516
    └── country=GB
        └── 20170516T1600-GB-01.tar.gz
        └── 20170516T1600-GB-02.tar.gz
    └── country=US
        └── 20170516T1500-US-01.tar.gz
```

# Partitioning - Advantages

	select count(*) from datalake where dt='20170515'		select count(*) from datalake where dt >= '20170515' and dt < '20170516'	
	Non-Partitioned	Partitioned	Non-Partitioned	Partitioned
Run Time	9.71 sec	<b>2.16 sec</b>	10.41 sec	<b>2.73 sec</b>
Data Scanned	74.1 GB	<b>29.06 MB</b>	74.1 GB	<b>871.39 MB</b>
Cost	\$0.36	<b>\$0.0001</b>	\$0.36	<b>\$0.004</b>
<b>Results</b>	<b>77% faster, 99% cheaper</b>		<b>73% faster, 98% cheaper</b>	

# Compression

# Compression

- Compressing your data can speed up your queries significantly
- Splittable formats enable parallel processing across nodes

Algorithm	Splittable	Compression Ratio	Algorithm Speed	Good For
Gzip (DEFLATE)	No	High	Medium	Raw Storage
bzip2	Yes	Very High	Slow	Very Large Files
LZO	Yes	Low	Fast	Slow Analytics
Snappy	Yes and No *	Low	Very Fast	Slow & Fast Analytics

\* Depends on if the source format is splittable and can output each record into a Snappy Block

# Compression - Example

## Snappy Compression with Parquet File Format

Format	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
<b>Result</b>	<b>87% less</b>	<b>34x faster</b>	<b>99% less</b>	<b>99.7% savings</b>

# Compression – File Counts

Fewer, larger files are better than many, smaller files (when splittable)

- Faster Listing Operations
- Fewer Requests to Amazon S3
- Less Metadata to Manage
- Faster Query Performance

Query	# files	Run Time
select count(*) from datalake	5000 files	8.4 sec
select count(*) from datalake	1 file	2.31 sec
<b>Result</b>		<b>72% Faster</b>

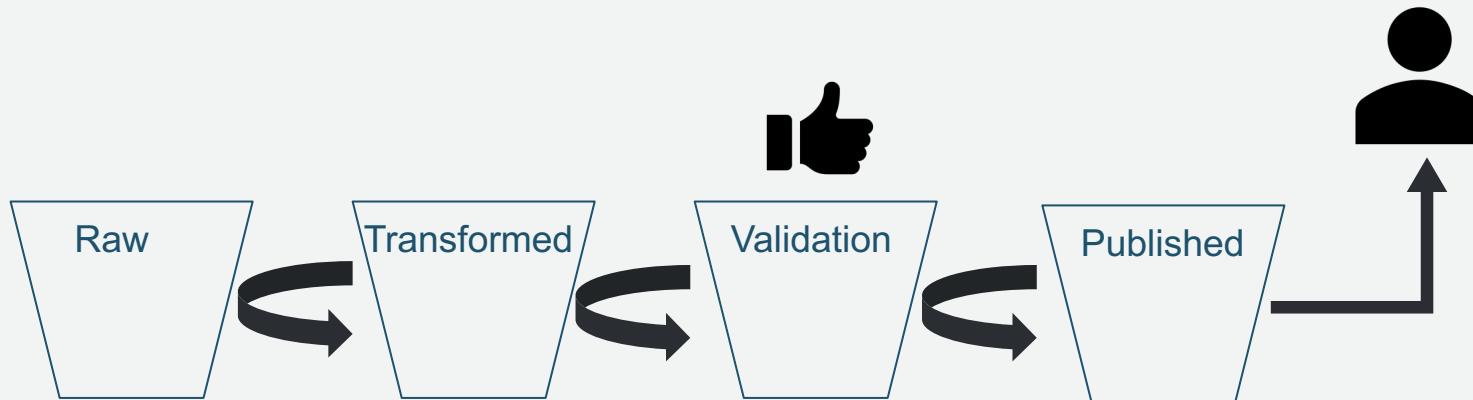
# Validation before publishing

# Huh... why?

# Remember Read-Committed Database Isolation?

*Read-committed: DB queries do not read data that is un-committed because it may be rolled back by the writer if some validation fails.*

Same concept in a data lake. Improve quality of insights by validating transformational integrity before publishing data to consumers



# AWS Lake Formation

Build a secure data lake in days

---



Identify, ingest, clean, and transform data

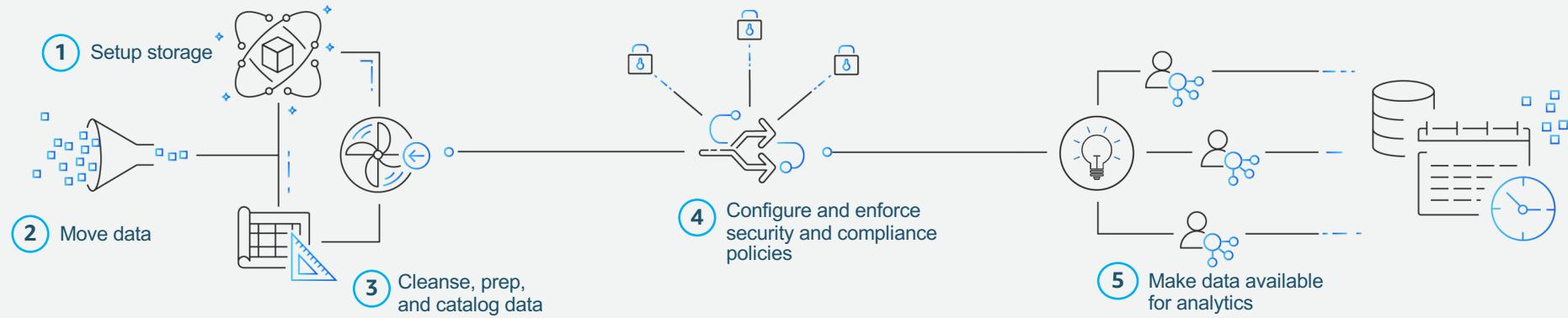


Enforce security policies across multiple services



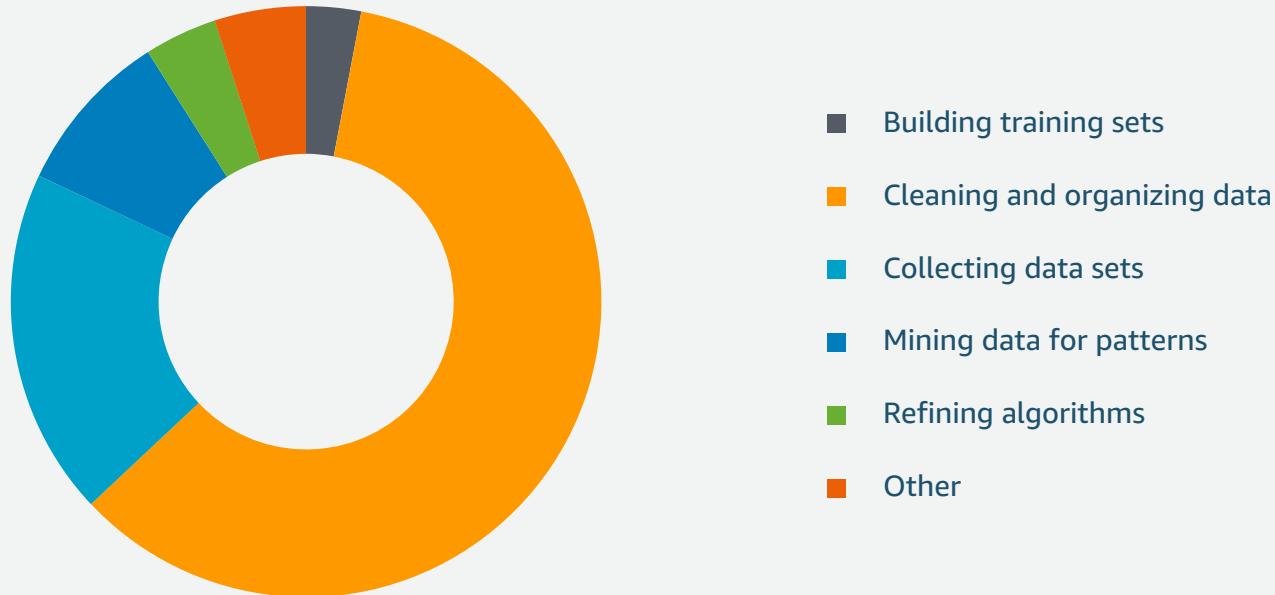
Gain and manage new insights

# Typical steps of building a data lake



Building data lakes can still take **months**

# Data preparation accounts for ~80% of the work

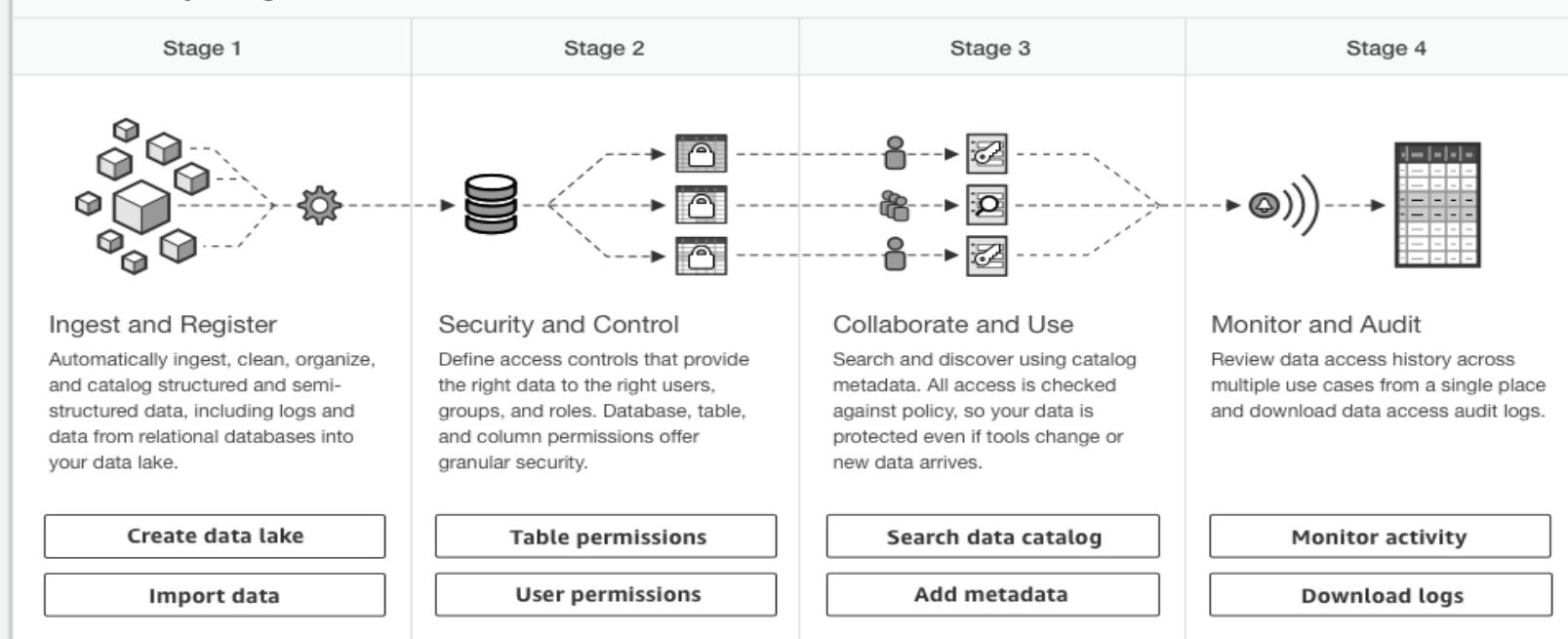


# How it works



## Overview

Data lake lifecycle stages and activities



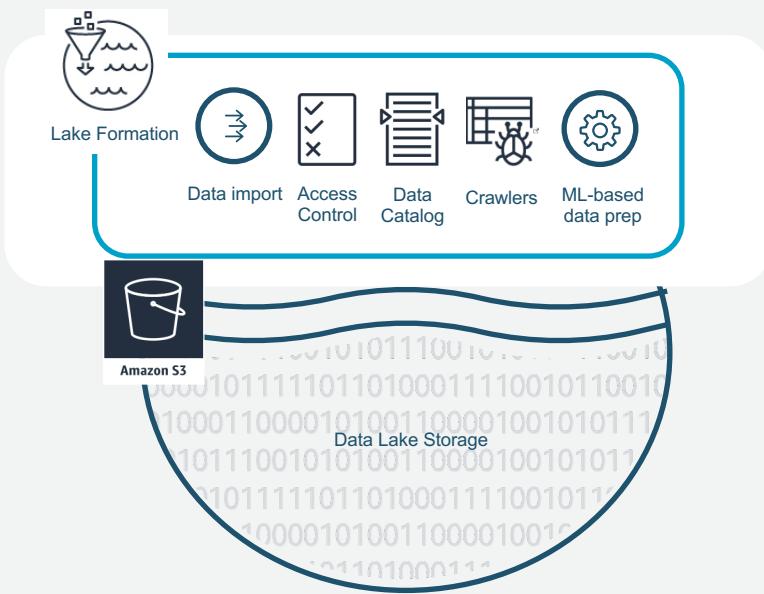
# Register existing data or import new

Amazon S3 forms the storage layer for Lake Formation

Register existing S3 buckets that contain your data

Ask Lake Formation to create required S3 buckets and import data into them

Data is stored in your account. You have direct access to it. No lock-in.



# Ingest & Transformations

# Easily load data to your data lake



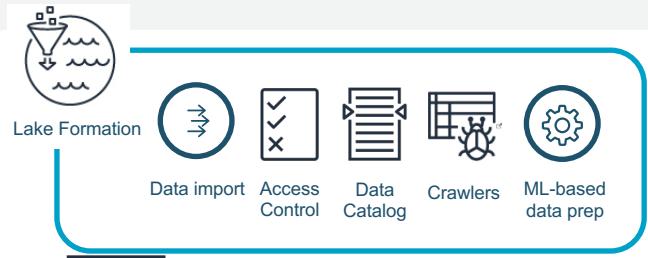
DBs

logs

## Blueprints

one-shot

incremental



# Blueprints & Workflows



Services

Resource Groups



galbarn @ galbarn

N. Virginia

Support

AWS Glue

Data catalog

Databases

Tables

Connections

Crawlers

Classifiers

Settings

ETL

Workflows

Jobs

ML Transforms

Triggers

Dev endpoints

Notebooks

Security

Security configurations

Tutorials

Add crawler

Explore table

Add job

Resources

What's new

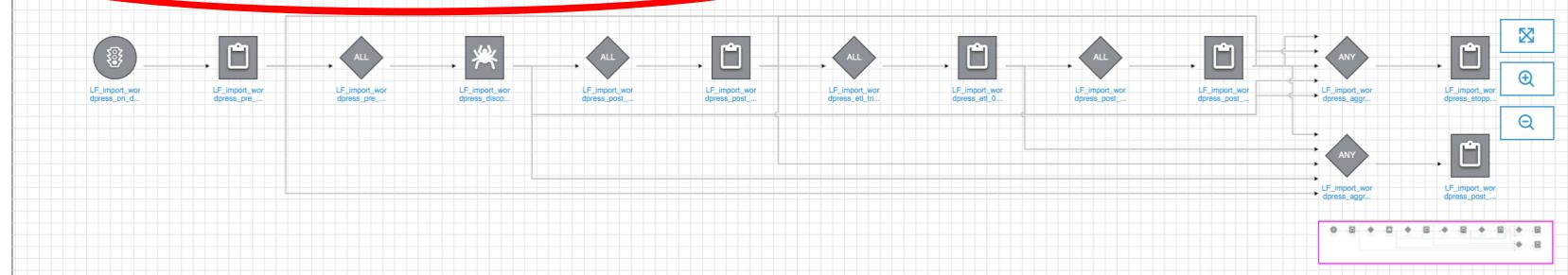
## Workflows (1)

A workflow is an orchestration used to visualize and manage the relationship and execution of multiple triggers, jobs and crawlers.

[Add workflow](#)[Actions](#) Filter workflows

&lt; 1 &gt;

Name	Last run	Last run status	Last modified
LF_import_wordpress	Sat, 06 Jul 2019 17:5...	Completed	Fri, 05 Jul 2019 14:53:31 GMT

[Graph](#) [Details](#) [History](#)**Legend:** ● Start ◆ Trigger Job Crawler Incomplete Error Deleting[Remove](#) [Action](#)

# Blueprints & Workflows

AWS Services Resource Groups ⚙

galbarn @ galbarn N. Virginia Support

AWS Lake Formation X

AWS Lake Formation > Blueprints > LF\_import\_wordpress

LF\_import\_wordpress

Start Delete View graph

Workflow details

Name	Last updated
LF_import_wordpress	Fri, 05 Jul 2019 14:53:31 GMT
Last run status	Created on
COMPLETED	Fri, 05 Jul 2019 14:53:31 GMT

Workflow runs (4)

Name	Started on	Run ID
LF_import_wordpress	Sat, 06 Jul 2019 17:13:02 GMT	wr_c9340b8a50bfd677a5cc292ea1fe223a41742f67d7a7beba9098762cab58760
LF_import_wordpress	Fri, 05 Jul 2019 16:19:13 GMT	wr_7226d0c9b470a5121e8a0155fd6aafe303c1a3bd81fe95153fb7fd7a9faf144
LF_import_wordpress	Fri, 05 Jul 2019 16:04:08 GMT	wr_6c6cff77bcddeb1fe6c519d61db23d0f0d647125ee5f54b6c931b3c77192bc201
LF_import_wordpress	Fri, 05 Jul 2019 14:55:17 GMT	wr_e901e893755318658b8731739d9c46d7bb16aba184c65c6ad8fe09ad495ba658

Filter Workflow runs < 1 > ⚙

# Blueprints & Workflows

Workflows > LF\_import\_wordpress > Run: wr\_c9340b8a50bfd677a5cc292ea1fe223a41742f67d7a7beba9098762cabc58760 > Graph

Workflow LF\_import\_wordpress

name

Run ID wr\_c9340b8a50bfd677a5cc292ea1fe223a41742f67d7a7beba9098762cabc58760

Run status Completed

Start time Sat, 06 Jul 2019 17:13:02 GMT

End time Sat, 06 Jul 2019 17:51:05 GMT

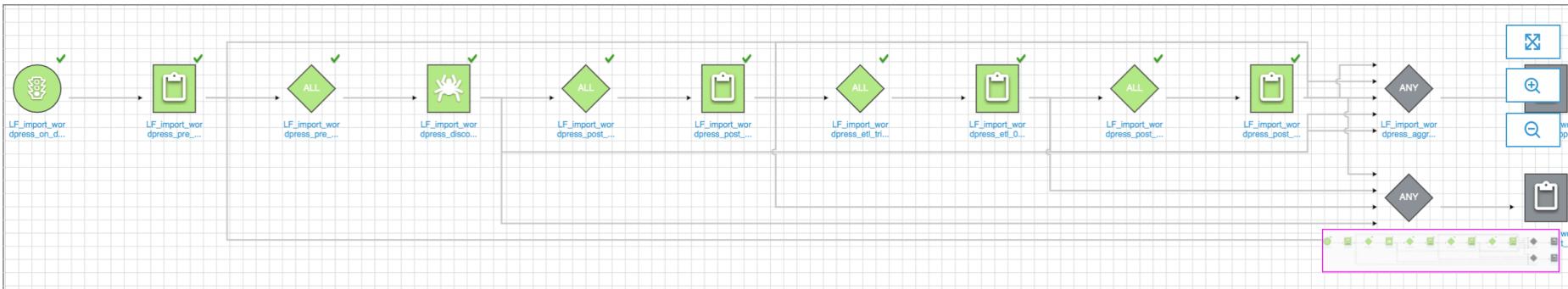
Execution time 38 Minutes

Run properties

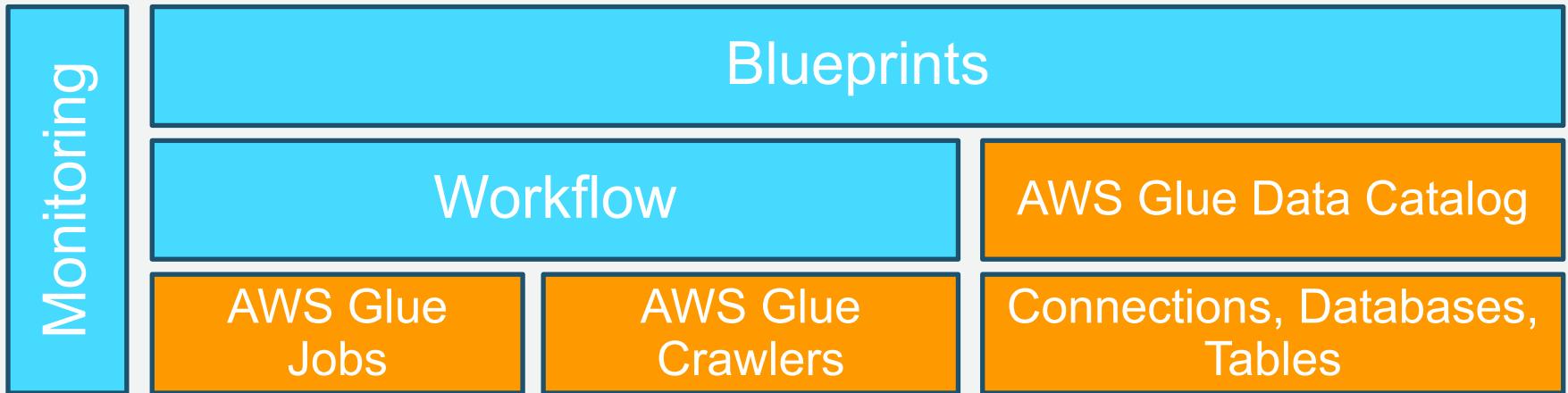
run\_state COMPLETED

Graph

Legend: ✓ Completed ⚡ Running ⚠ Warning ✖ Error ✎ Deleting

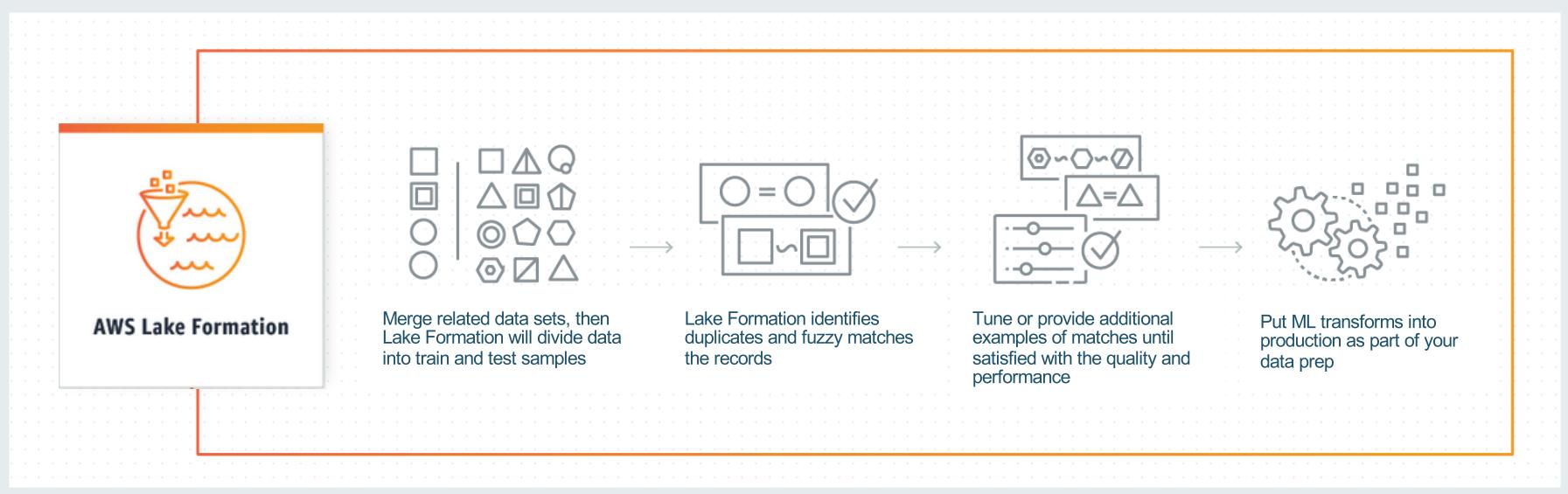


# Blueprints build on AWS Glue



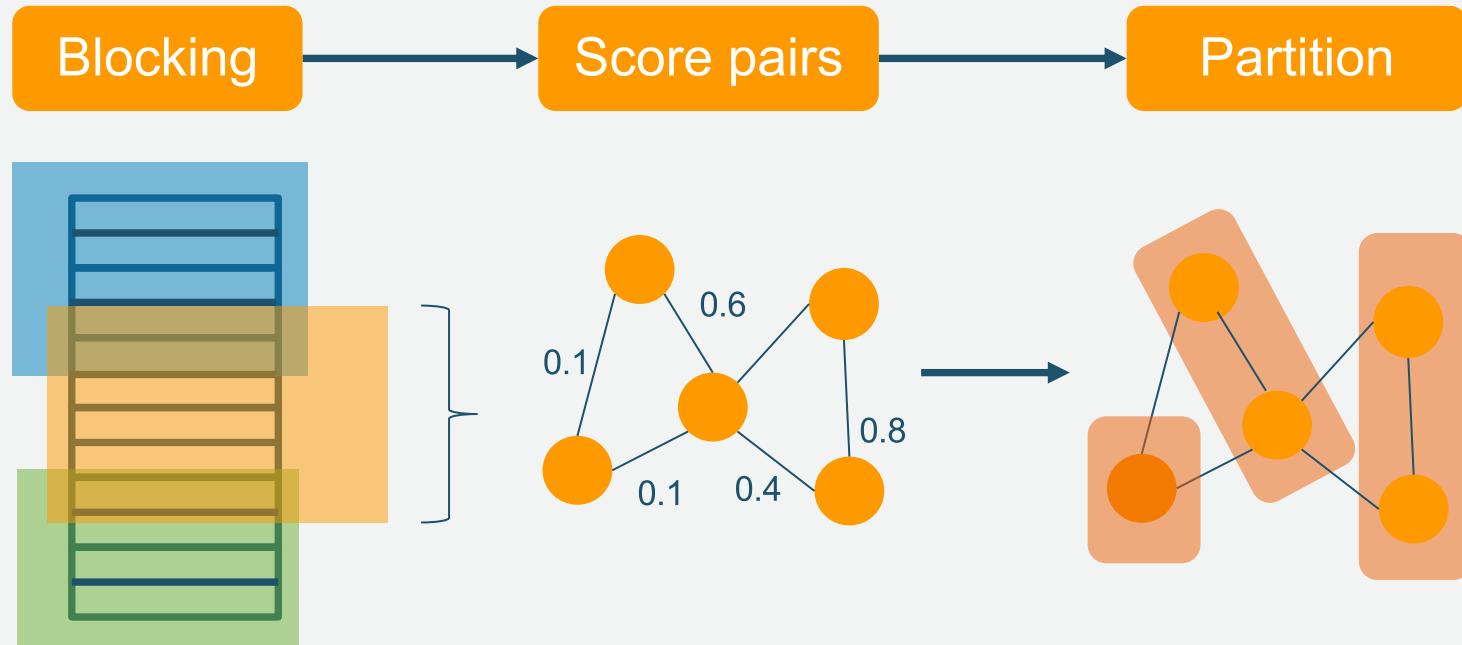
# ML Transformations

# Easily de-duplicate your data with ML transforms



# Fuzzy de-duplication – under the hood

Naïve: look at all pairs,  $N^2$  – state-of-the-art:



# Fuzzy de-duplication – Innovations

## Intersection Dynamic Blocking (VLDB 2008)

parallelizable & performant  
blocks on dynamic mix of columns

**400M** rows

**7.5B+** candidate pairs

**2.5** hours

## SuperPart

partitions based on customer-  
provided ground-truth

gives confidence of grouping

effective without tuning knobs

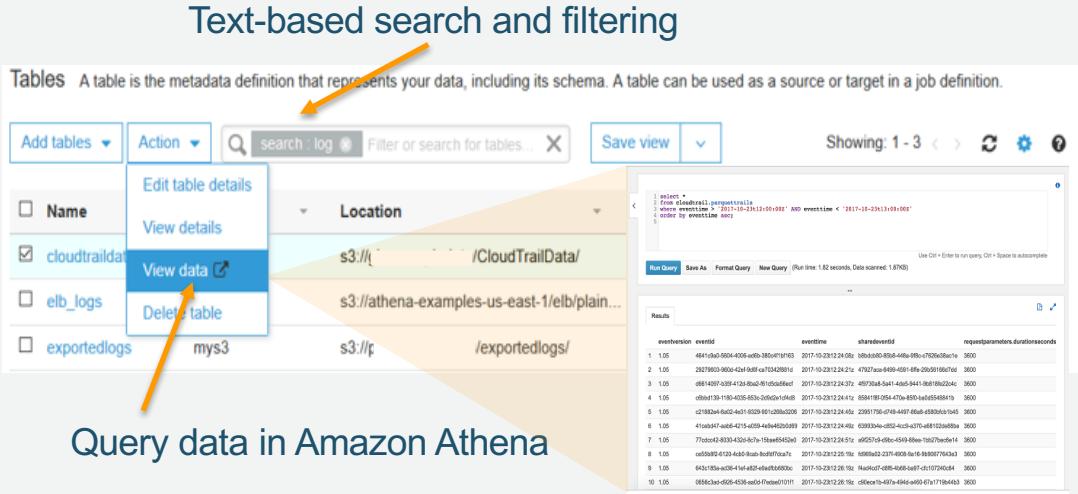
# Data Catalog

# Enhanced Data Catalog

Text-based, faceted search  
across all metadata

Add attributes like Data owners, stewards, and other as table properties

Add data sensitivity level, column definitions, and others as column properties



# Enhanced Data Catalog

AWS Lake Formation X

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 > wp\_\_wordpress\_wp\_posts

wp\_\_wordpress\_wp\_posts

Version 3 (Current version)

Actions ▾

Compare versions

Drop table

View properties

Edit table

## Table details

Table name

wp\_\_wordpress\_wp\_posts

Description

-

Database

lakeformation\_techsummit

Classification

PARQUET

Location

[s3://datalake-galbarn-us-east/wp\\_\\_wordpress\\_wp\\_posts/version\\_1/](s3://datalake-galbarn-us-east/wp__wordpress_wp_posts/version_1/) 

Connection

-

Last updated

Mon, 08 Jul 2019 03:00:19 GMT

Input format

# Enhanced Data Catalog

## ▼ Table properties

Key	Value	
CreatedByJob	LF_import_wordpress_etl_0_dacaccc0	<button>Remove</button>
CreatedByJobRun	jr_3c0502dd7fd0b09e9a7ea3f36974a	<button>Remove</button>
LastTransformCompletedOn	2019-07-06 17:41:49.958976	<button>Remove</button>
LastUpdatedByJob	LF_import_wordpress_etl_0_dacaccc0	<button>Remove</button>
LastUpdatedByJobRun	jr_13e5c00273cab7374a754c9dbeb7;	<button>Remove</button>
SourceConnection	wp_rds_1	<button>Remove</button>
SourceTableName	wordpress_wp_posts	<button>Remove</button>
SourceType	JDBC	<button>Remove</button>
TableVersion	1	<button>Remove</button>
TransformTime	0:01:07.909511	<button>Remove</button>
classification	PARQUET	<button>Remove</button>

Add

## ▼ Table properties

Key	Value	
CreatedByJob	LF_import_wordpress_etl_0_dacaccc0	<button>Remove</button>
CreatedByJobRun	jr_3c0502dd7fd0b09e9a7ea3f36974a	<button>Remove</button>
LastTransformCompletedOn	2019-07-06 17:41:49.958976	<button>Remove</button>
LastUpdatedByJob	LF_import_wordpress_etl_0_dacaccc0	<button>Remove</button>
LastUpdatedByJobRun	jr_13e5c00273cab7374a754c9dbeb7;	<button>Remove</button>
PII	no	<button>Remove</button>
SourceConnection	wp_rds_1	<button>Remove</button>
SourceTableName	wordpress_wp_posts	<button>Remove</button>
SourceType	JDBC	<button>Remove</button>
TableVersion	1	<button>Remove</button>
TransformTime	0:01:07.909511	<button>Remove</button>
classification	PARQUET	<button>Remove</button>
sensitivity	high	<button>Remove</button>

Add

# Enhanced Data Catalog

AWS Lake Formation X

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

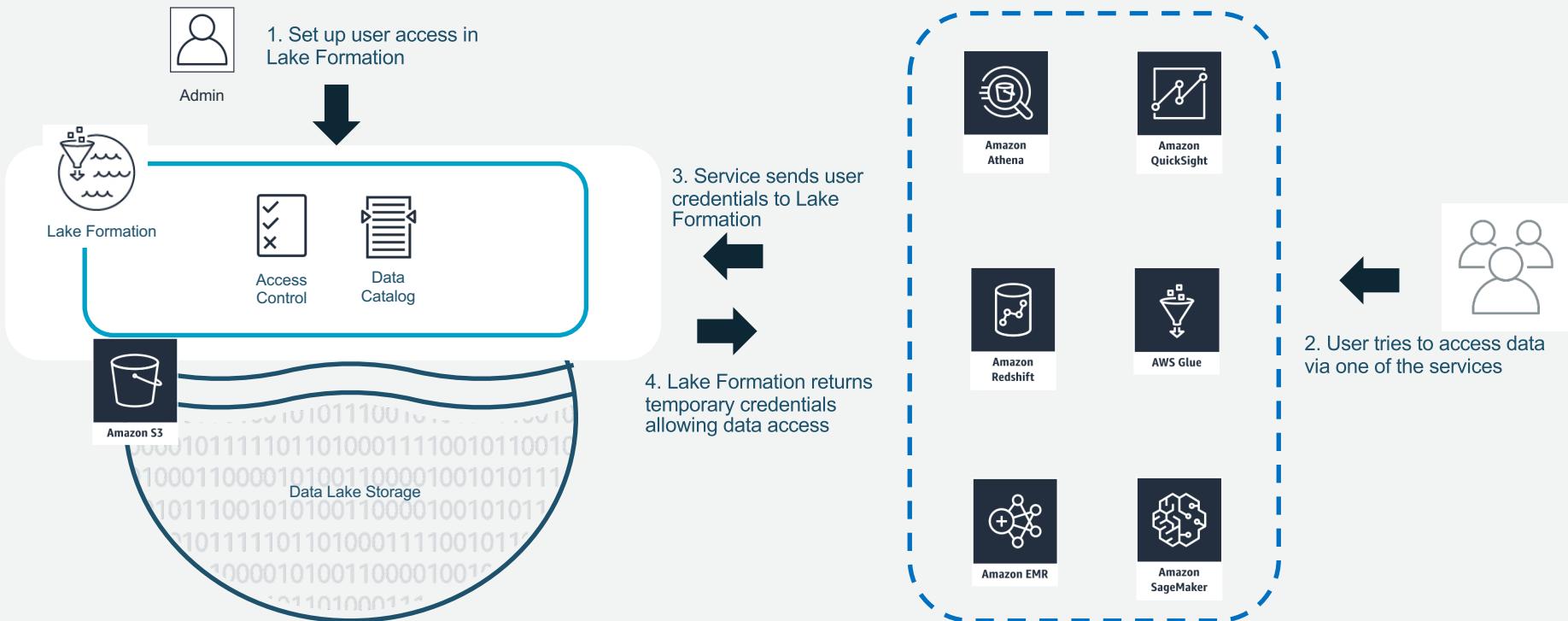
Tables (2)

keyword : sensitivity:high X

Name	Database	Location	Classification	Last updated
wp__wordpress_wp_comment s	lakeformation_techsummit	s3://datalake-galbarn-us-east/wp__wordpress_wp_comments/version_1/	PARQUET	Mon, 08 Jul 2019 02:55:27 GMT
wp__wordpress_wp_posts	lakeformation_techsummit	s3://datalake-galbarn-us-east/wp__wordpress_wp_posts/version_1/	PARQUET	Mon, 08 Jul 2019 03:00:19 GMT

# Security & Governance

# Secure once, access in multiple ways



# Security permissions in Lake Formation

Control data access with simple grant and revoke permissions

Specify permissions on tables and columns rather than on buckets and objects

Easily view policies granted to a particular user

Audit all data access at one place

The screenshot shows the AWS Lake Formation 'Tables' page with two tables listed: 'reviews' and 'orders'. A context menu is open over the 'reviews' table, with the 'Grant' option highlighted. The menu also includes options like 'Edit', 'Copy', 'Delete', 'Revoke', and 'Verify permissions'. Below this, a modal dialog titled 'Grant permissions to table orders' is displayed, showing a list of IAM users ('john', 'Salesgrp', 'analyst') selected for granting permissions. It provides options to 'Grant all' or 'Specific permissions' (selected), and lists checkboxes for various database actions. At the bottom of the dialog are 'Cancel' and 'Save' buttons.

# Security permissions in Lake Formation

Search and view permissions granted to a user, role, or group in one place

Verify permissions granted to a user

Easily revoke policies for a user

The screenshot illustrates the AWS Lake Formation User permissions interface. It shows a hierarchical tree of permissions under a user named 'johnd'. The 'sales' database has 'Create, Select, Insert' permissions, while 'reviews' and 'orders' tables have 'Administrator' and 'Create, Drop' permissions respectively. A modal window titled 'Verify permissions for database sales' allows selecting IAM users, groups, or roles for permission verification. Another modal shows 'Permissions for selected users' for 'johnd' and 'salesgrp'. A final modal, 'Revoke all permissions for johnd', contains a confirmation field and 'Cancel'/'Revoke' buttons.

AWS Lake Formation > User permissions

User permissions (3)

Name	Type	Permissions	Last modified
johnd	User	3 permissions	11/28/2018 14:24
<input checked="" type="checkbox"/> sales	Database	Create, Select, Insert	11/28/2018 13:37
<input type="checkbox"/> reviews	Table	Administrator	11/28/2018 13:37
<input type="checkbox"/> orders	Table	Create, Drop	11/28/2018 12:44
<input type="checkbox"/> salesgrp			
<input type="checkbox"/> sales			
<input type="checkbox"/> analyst			
<input type="checkbox"/> reviews			

Verify permissions for database sales

IAM user, group, role(s) Info  
Choose one or more IAM users, groups, and roles to verify access permission

Add user  
johnd X salesgrp X

Permissions for selected users (2)

Name	Type	Permissions
<input checked="" type="checkbox"/> johnd	User	Create, Select, Insert
<input type="checkbox"/> salesgrp	Group	Administrator

Revoke all permissions for johnd

To confirm, type `revoke all` into the field.

Cancel Revoke

# Grant table and column-level permissions



# Audit and monitor in real time

See detailed alerts in the console

Download audit logs for further analytics

Data ingest and catalog notifications also published to Amazon CloudWatch events

The screenshot shows the AWS Lake Formation Dashboard. The top navigation bar includes 'Services', 'Resource Groups', and 'Support'. The left sidebar has sections for 'Dashboard', 'Ingest and register', 'Data catalog', 'Users and permissions', and 'Monitor and audit' (with 4 notifications). The main area displays the 'Overview' of data lake lifecycle stages and activities across four stages: Stage 1 (Ingest and Register), Stage 2 (Security and Control), Stage 3 (Collaborate and Use), and Stage 4 (Monitor and Audit). Below each stage is a brief description and several buttons: 'Create data lake', 'Import data', 'Table permissions', 'User permissions', 'Search data catalog', 'Add metadata', 'Monitor activity', and 'Download logs'. A red oval highlights the 'Recent activity' section at the bottom right. This section shows a table of recent events with columns for 'Description', 'Type', 'Resource', and 'Alert time'. The events listed are:

Description	Type	Resource	Alert time
Create access was granted to JohnD	Grant	data_science:trend_extract	November 28, 2018 14:24
<input checked="" type="checkbox"/> Select access was revoked for TimK	Revoke	sales:invoices:customer, address	November 28, 2018 13:57
Data was accessed via Athena by DaveM	Access	finance:2018_monthly	November 28, 2018 13:06
A new table was added by DaveM	New table	finance:2019_projections	November 28, 2018 12:44

# Demo!