aws SUMMIT

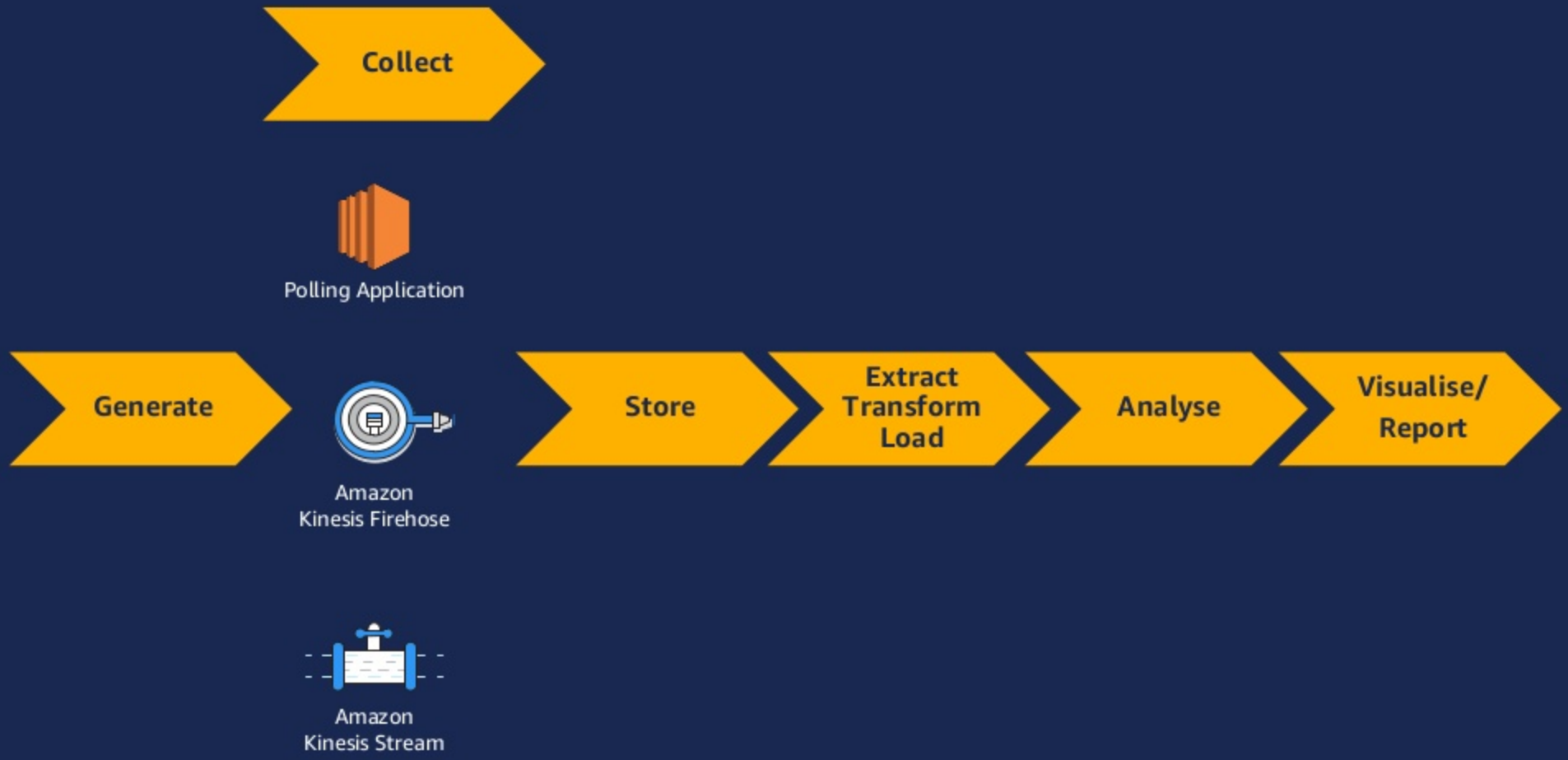
Building Serverless ETL Pipelines With AWS Glue

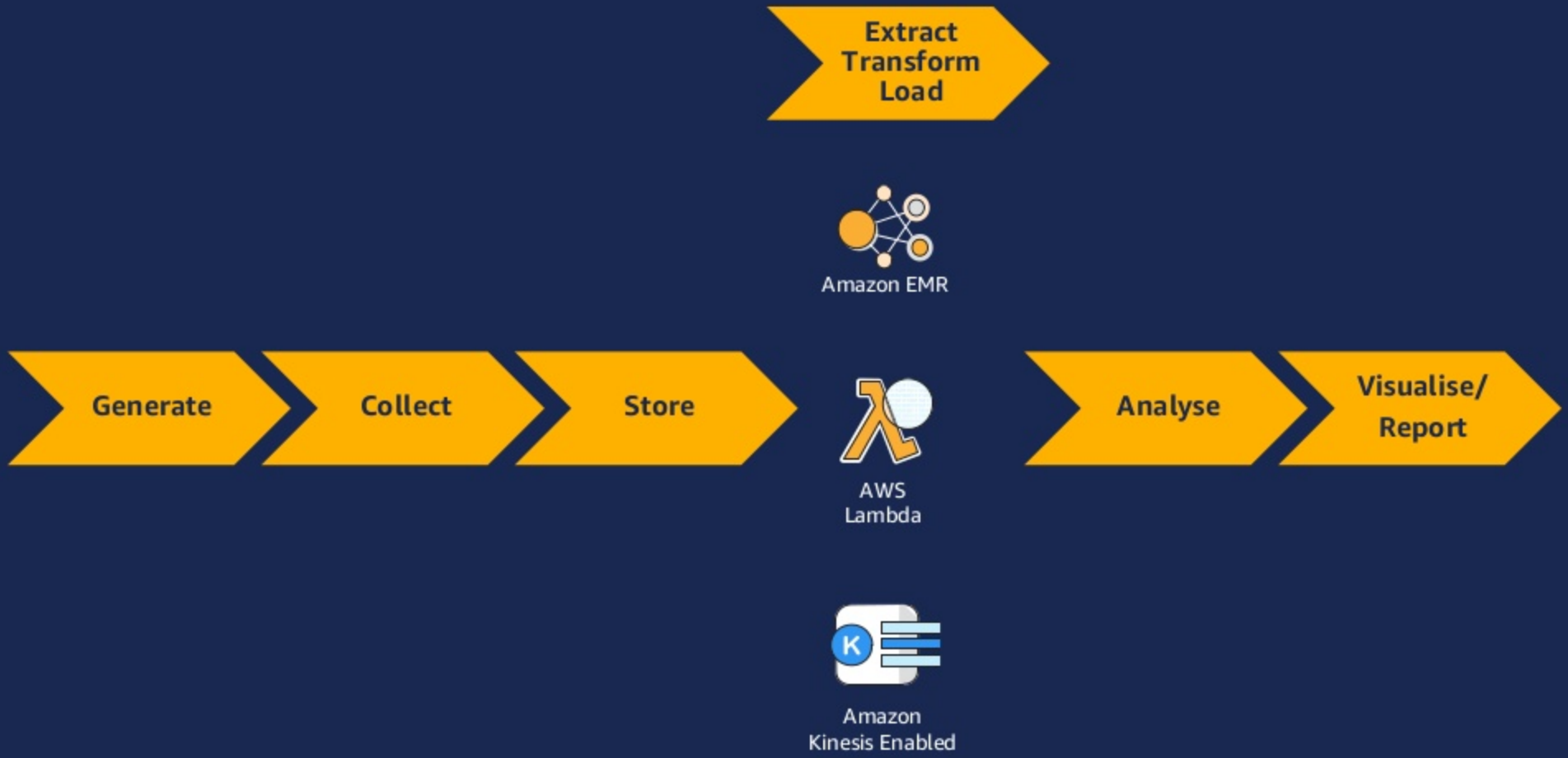
Ben Thurgood

Principal Solutions Architect, Amazon Web Services

Can I get you to go ahead and…

…prepare our data for analysis





No Problem…?

Deal with these Terabytes and Petabytes of data

Simplify querying disparate data sets

Combine existing / legacy data with modern data sets

Prepare data for machine learning

Name	Last modified	Size	Storage class
04dc3925_SO00016	--	--	--
0608917d_SO0002	--	--	--
07a46e08_SO0012	--	--	--
14374589_SO0009	--	--	--
1c1e778f_SO0008	--	--	--
1f6c8c88_SO0007	--	--	--
2fcb2661_SO0015	--	--	--
3225cb4a_SO00010	--	--	--
363eb7cb_SO0018	--	--	--
5a6dfbe5_SO009s	--	--	--
75d6f6be_SO0004	--	--	--

Name	Last modified	Size	Storage class
0051a423-8e40-406e-9db4-d1e88e496101	--	--	--
02c70d2c-8ce4-4d51-be00-c8a5b8270b01	--	--	--
16cee6b1-0eea-4042-9b9f-dcae989e2c1f	--	--	--
16f23b83-95e4-40b4-927a-37713b907266	--	--	--
195f54e8-5f84-44a3-8a9a-c3165211c5bb	--	--	--
1e4d0fb1-0964-46b5-bf9b-0fd09b44c9e4	--	--	--
1f3e86af-f3a6-4cb5-a3ec-a664b6d6c09f	--	--	--
2203aebd-bd7f-42b9-b1e6-90ac11f8d030	--	--	--
25eeff5c-af58-4d52-93f7-a83ab6f5fe80	--	--	--
3334a568-af1a-4c65-a450-219f5282f673	--	--	--
34805536-0d3a-4743-be5d-05776764522d	--	--	--

Name	Last modified	Size	Storage class
0051a423-8e40-406e-9db4-d1e88e496101	--	--	--
02c70d2c-8ce4-4d51-be00-c8a5b8270b01	--	--	--
1f3e86af-f3a6-4cb5-a3ec-a664b6d6c09f	--	--	--
2203aebd-bd7f-42b9-b1e6-90ac11f8d030	--	--	--
25eeff5c-af58-4d52-93f7-a83ab6f5fe80	--	--	--
3334a568-af1a-4c65-a450-219f5282f673	--	--	--
34805536-0d3a-4743-be5d-05776764522d	--	--	--


```
[{"ground_speed": "230", "spi": "0", "lastUpdated": 1510202706.596984,
  "callsign": "QLK476D ", "is_on_ground": "0", "altitude": "3600",
  "alert": "0", "emergency": "0", "message_type": "MSG",
  "transmission_type": "5","lat": "-34.16263", "hex_ident": "7C530C",
  "lon": "151.12625", "vertical_rate": "-768", "track": "77", "squawk": "4263"}]
```

Some extra challenges..

Volumes will grow (the new oil)

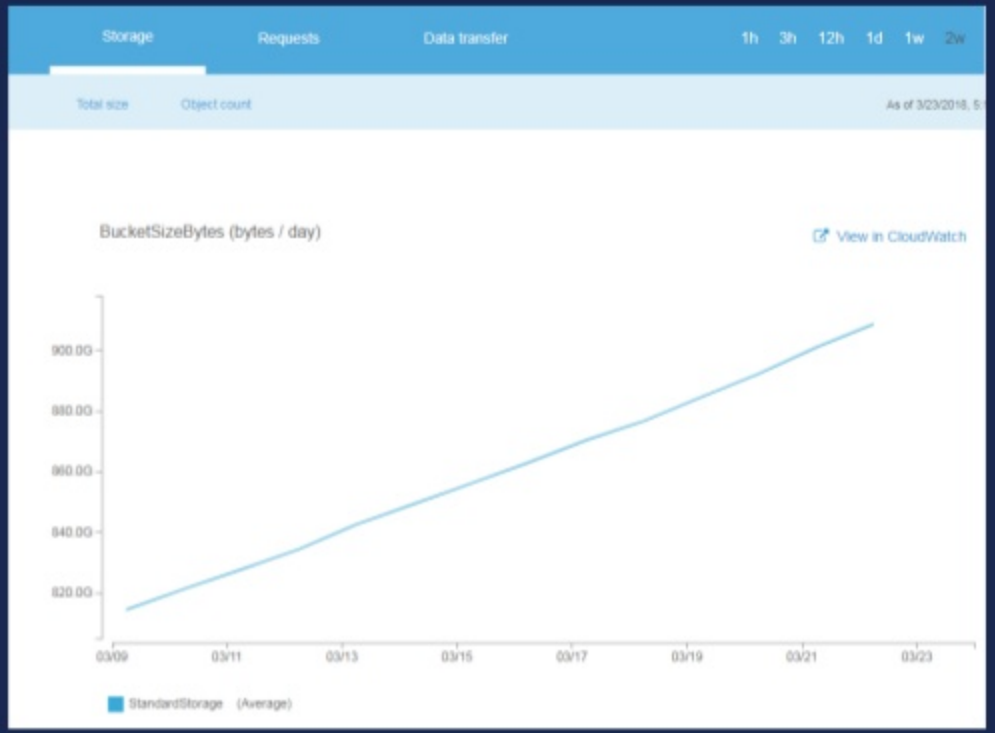
Adding data sources

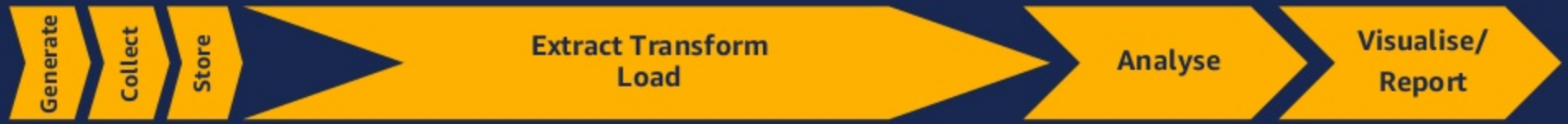
Large proportion of ETL is hand coding

Data formats change over time

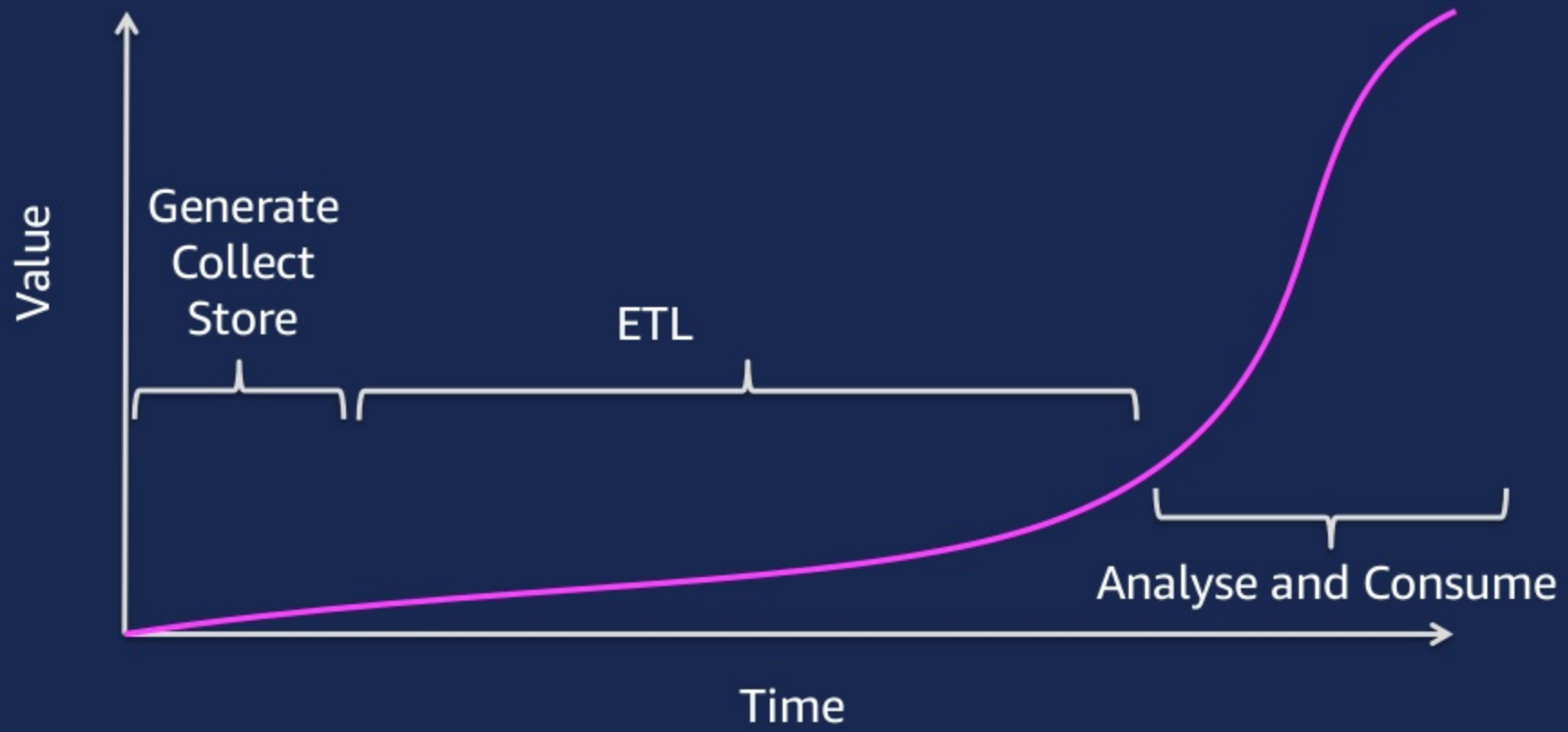
- Within data you already have
- Changes will be coming soon

Target schemas change





And.. ETL Is Not The Rewarding Part



Why AWS Glue?

Automate your ETL

Automatically discover and categorise your data

- Connect to your data sources
- Generate your Data Catalogue

Make it immediately searchable and queryable

- Athena
- Redshift
- EMR

Automate your ETL

Generates your ETL code

- Clean
- Enrich
- Move

Adaptable code

Extension to Spark in Python or Scala

Automate your ETL

Runs your ETL jobs serverless

- Managed
- Control the amount of resources used
- Scales out automatically

Schedule or trigger jobs

Why Glue Examples

How do I ETL my data?

Four Steps

Crawl

Map

Edit and Explore

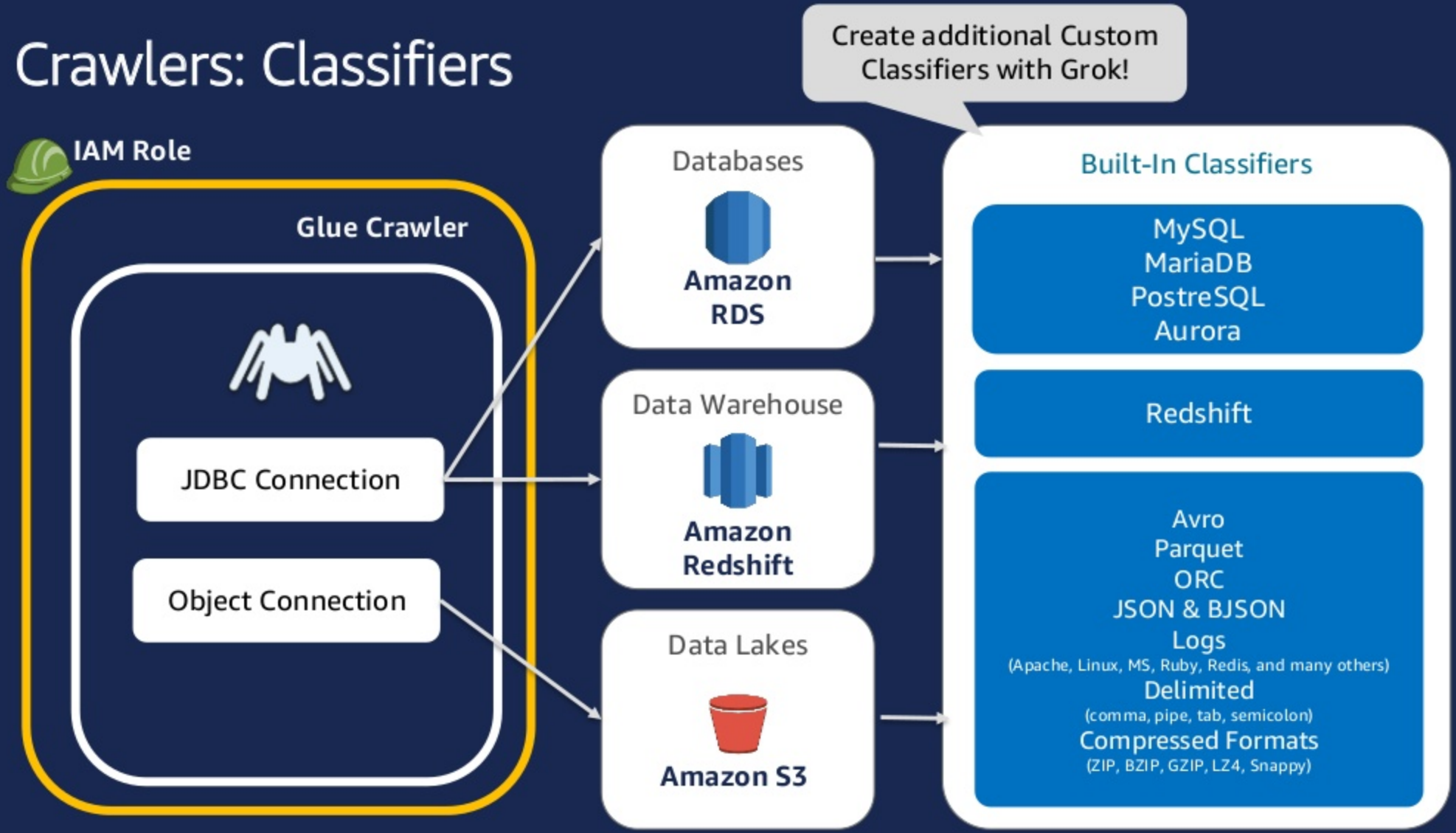
Schedule

How Do I Discover My Data?

Glue Data Catalogue: Crawlers

- Automatically discover new data and extract schema definitions
  - Detect schema changes and version tables
  - Detect Apache Hive style partitions on Amazon S3
- Built-in classifiers for popular data types
  - Custom classifiers using Grok expressions
- Run ad hoc or on a schedule; serverless – only pay when crawler runs

Crawlers: Classifiers



Example Classifier

2018-03-18T01:44:19+00:00 [prefix-p-123-a-7z] WARN: There is a message

Grok expression example:

%{TIMESTAMP_ISO8601:timestamp} \[%{MESSAGEPREFIX:message_prefix}\] %{CRAWLERLOGLEVEL:loglevel} : %{GREEDYDATA:message}

Built in patterns:

TIMESTAMP_ISO8601 %{YEAR}-%{MONTHNUM}-%{MONTHDAY}[T]%{HOUR}:?%{MINUTE}(?::?%{SECOND})?%{ISO8601_TIMEZONE}?

GREEDYDATA .*

Custom patterns

CRAWLERLOGLEVEL (BENCHMARK|ERROR|WARN|INFO|TRACE)

MESSAGEPREFIX .*-.*-.*-.*-.*

Handy Grok debugger:
https://grokdebug.herokuapp.com/

Crawler: Detecting Partitions

S3 bucket hierarchy

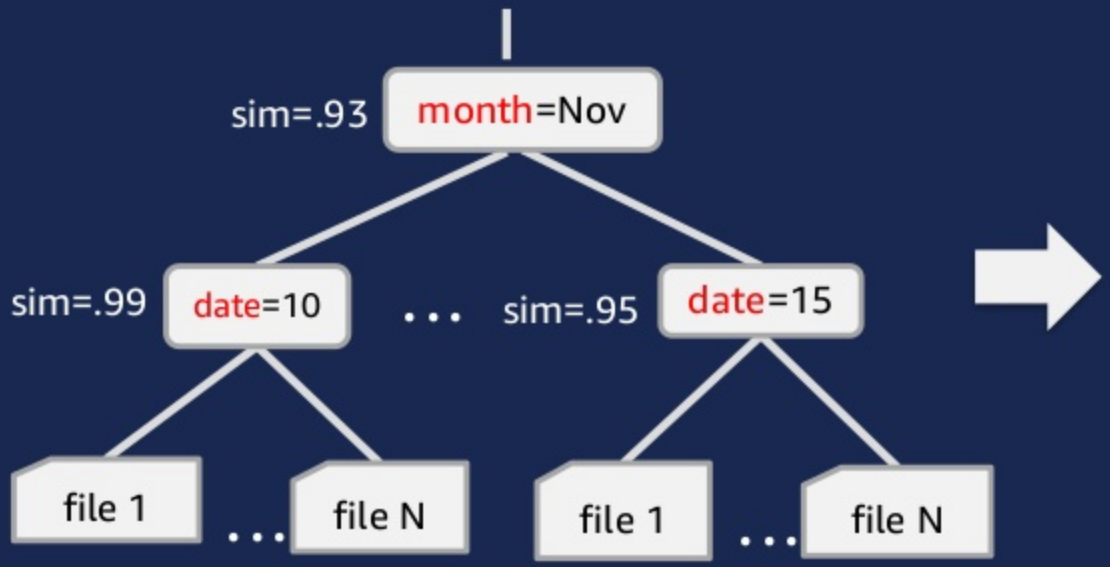


Table definition

Column	Type
month	str
date	str
col 1	int
col 2	float
⋮	⋮

Estimate schema similarity among files at each level to handle semi-structured logs, schema evolution…

Glue Data Catalog

aws Services Resource Groups ben.thurgood @ 1652-2256-46... Sydney Support

AWS Glue

Data catalog
- Databases
  - Tables
  - Connections
- Crawlers
  - Classifiers

ETL
- Jobs
- Triggers
- Dev endpoints

Tutorials
- Add crawler
- Explore table
- Add job
- Resources
- What's new

Tables A table is the metadata definition that represents your data, including its schema. A table can be used as a source or target in a job definition.

Add tables | Action | Filter by attributes or search by keyword | Save view | Showing: 1 - 100

Name	Database	Location	Classification	Last updated	Deprecated
accountcost	[redacted]hydro	s3:[redacted]-poc/	Unknown	3 August 2017 5:55 PM...	
[redacted]	[redacted]	s3:[redacted]-test-sydney-iot/	json	23 March 2018 6:55 P...	
data	nyctaxi	s3://aws-bigdata-blog/artifacts/glue-data-lake/data/	csv	11 March 2018 12:32 P...	
elb_logs	default	s3://athena-examples-us-west-2/elb/plaintext	Unknown		
elb_logs	sampledb	s3://athena-examples-us-west-2/elb/plaintext	Unknown		
nyc_taxi_raw_fhv_tripd...	nyc-taxi-raw	s3://nyc-tlc/trip data/fhv_tripdata_2015-01.csv	csv	18 March 2018 4:46 P...	
nyc_taxi_raw_fhv_tripd...	nyc-taxi-raw	s3://nyc-tlc/trip data/fhv_tripdata_2015-02.csv	csv	18 March 2018 4:46 P...	
nyc_taxi_raw_fhv_tripd...	nyc-taxi-raw	s3://nyc-tlc/trip data/fhv_tripdata_2015-03.csv	csv	18 March 2018 4:46 P...	
nyc_taxi_raw_fhv_tripd...	nyc-taxi-raw	s3://nyc-tlc/trip data/fhv_tripdata_2015-04.csv	csv	18 March 2018 4:46 P...	
nyc_taxi_raw_fhv_tripd...	nyc-taxi-raw	s3://nyc-tlc/trip data/fhv_tripdata_2015-05.csv	csv	18 March 2018 4:46 P...	
nyc_taxi_raw_fhv_tripd...	nyc-taxi-raw	s3://nyc-tlc/trip data/fhv_tripdata_2015-06.csv	csv	18 March 2018 4:46 P...	
nyc_taxi_raw_fhv_tripd...	nyc-taxi-raw	s3://nyc-tlc/trip data/fhv_tripdata_2015-07.csv	csv	18 March 2018 4:46 P...	
nyc_taxi_raw_fhv_tripd...	nyc-taxi-raw	s3://nyc-tlc/trip data/fhv_tripdata_2015-08.csv	csv	18 March 2018 4:46 P...	
nyc_taxi_raw_fhv_tripd...	nyc-taxi-raw	s3://nyc-tlc/trip data/fhv_tripdata_2015-09.csv	csv	18 March 2018 4:46 P...	
nyc_taxi_raw_fhv_tripd...	nyc-taxi-raw	s3://nyc-tlc/trip data/fhv_tripdata_2015-10.csv	csv	18 March 2018 4:46 P...	
nyc_taxi_raw_fhv_tripd...	nyc-taxi-raw	s3://nyc-tlc/trip data/fhv_tripdata_2015-11.csv	csv	18 March 2018 4:46 P...	

Feedback English (US) © 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Glue Data Catalog: Table Properties

Table properties

Data statistics

Table schema

Nested fields

AWS Glue

Data catalog
Databases
Tables
Connections
Crawlers
Classifiers

ETL
Jobs
Triggers
Dev endpoints

Tutorials
Add crawler
Explore table
Add job

Tables > simpletweets_json

Last updated 10 Aug 2017 Table Version (Current version)

Edit table | Delete table | View properties | Compare versions | Edit schema

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

Close

Glue Data Catalog: Version control

Compare schema versions

List of table versions

Last updated 10 Aug 2017 **Table** Version 0

Name	simpletweets_json
Description	
Database	simpletweets_json
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**

Change	Column name	Data type	Key
	entities	struct	
Changed	id	bigint	
	retweeted	boolean	
	text	string	
	user	struct	

Last updated 10 Aug 2017 **Table** Version 1 (Current version)

Name	
Description	
Database	
Classification	
Location	
Connection	
Deprecated	
Last updated	
Properties	compressionType **none** typeOfData **file**

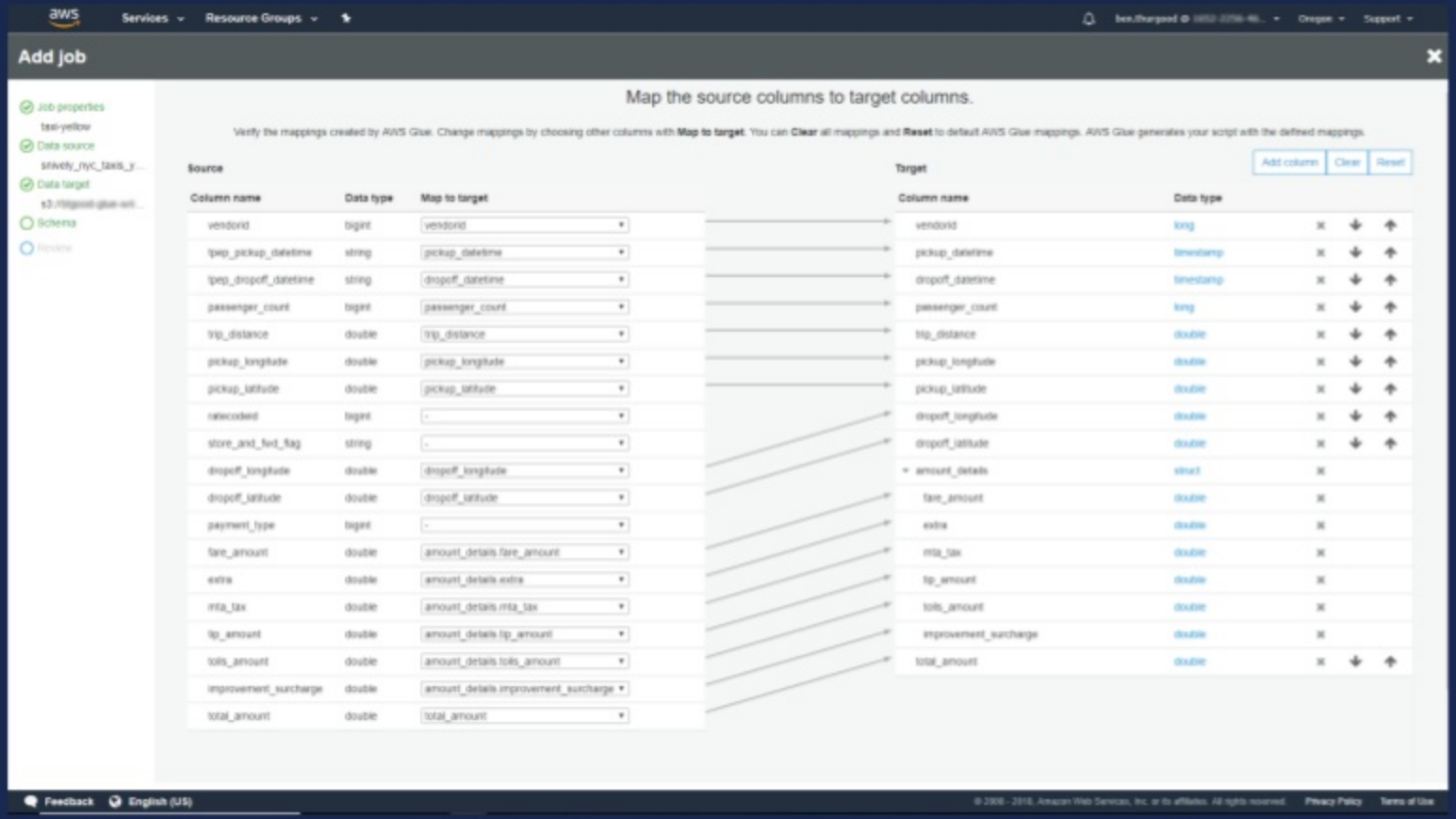
Showing: 1 - 2

Version	Created:	Created by:
1	10 August 2017 ...	
0	10 August 2017 ...	arn:aws:sts::413... role/Glue_Default... Crawler

Change	Column name	Data type	Key
	entities	struct	
Changed	id	STRING	
	retweeted	boolean	
	text	string	
	user	struct	

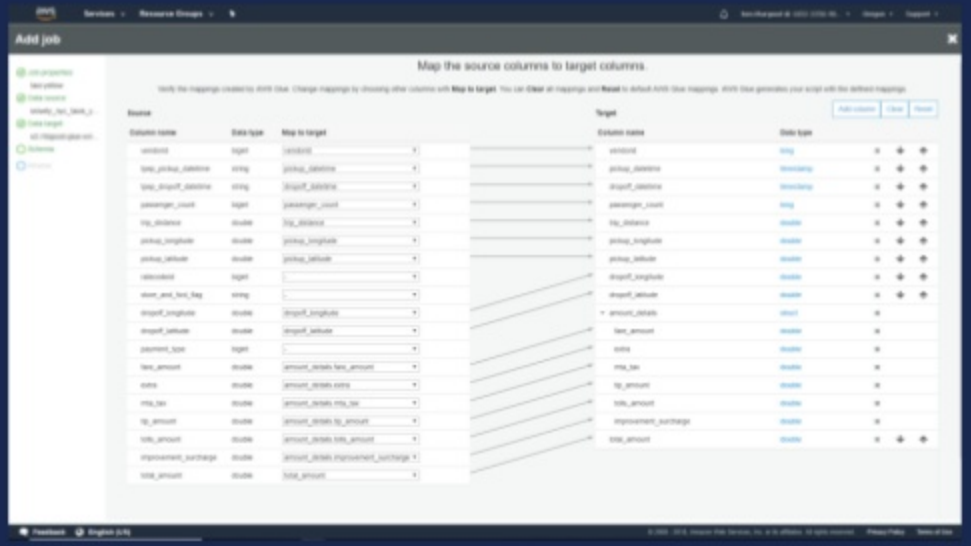
How Do I Build The ETL?

Job Authoring: Automatic Code Generation



aws SUMMIT

Job Authoring: Automatic Code Generation

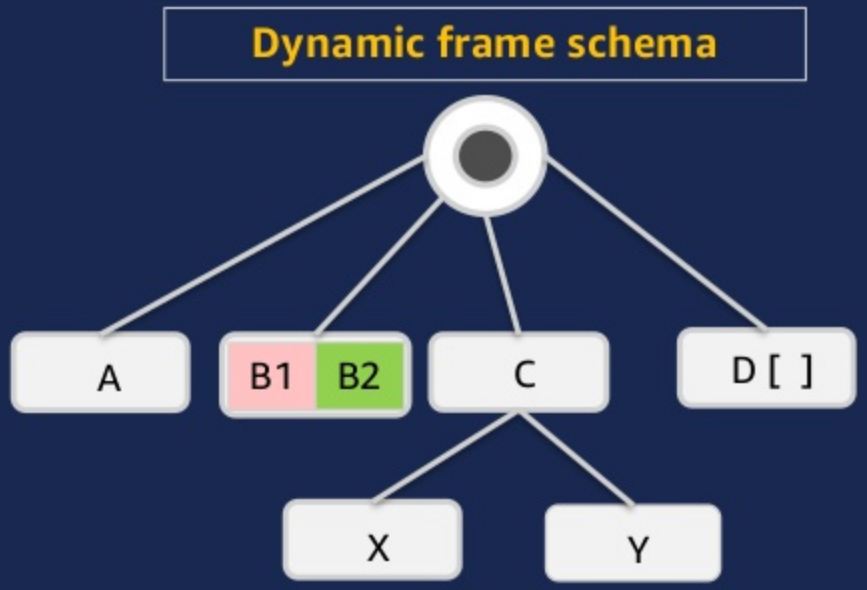


Job Authoring: ETL Code

- **Human-readable**, editable, and portable PySpark code

```
sc = SparkContext()
glueContext = GlueContext(sc)
spark = glueContext.spark_session
job = Job(glueContext)
job.init(args['JOB_NAME'], args)
## @type: DataSource
## @args: [database = "snively-nyc-taxis-db", table_name = "snively_nyc_
## @return: datasource0
## @inputs: []
datasource0 = glueContext.create_dynamic_frame.from_catalog(database = '
```

Job Authoring: Glue Dynamic Frames



Like Apache Spark's Data Frames, but better for:

- Cleaning and (re)-structuring **semi-structured** data sets, e.g. JSON, Avro, Apache logs …

No upfront schema needed:

- Infers schema on-the-fly, enabling transformations in a **single pass**

Easy to handle the unexpected:

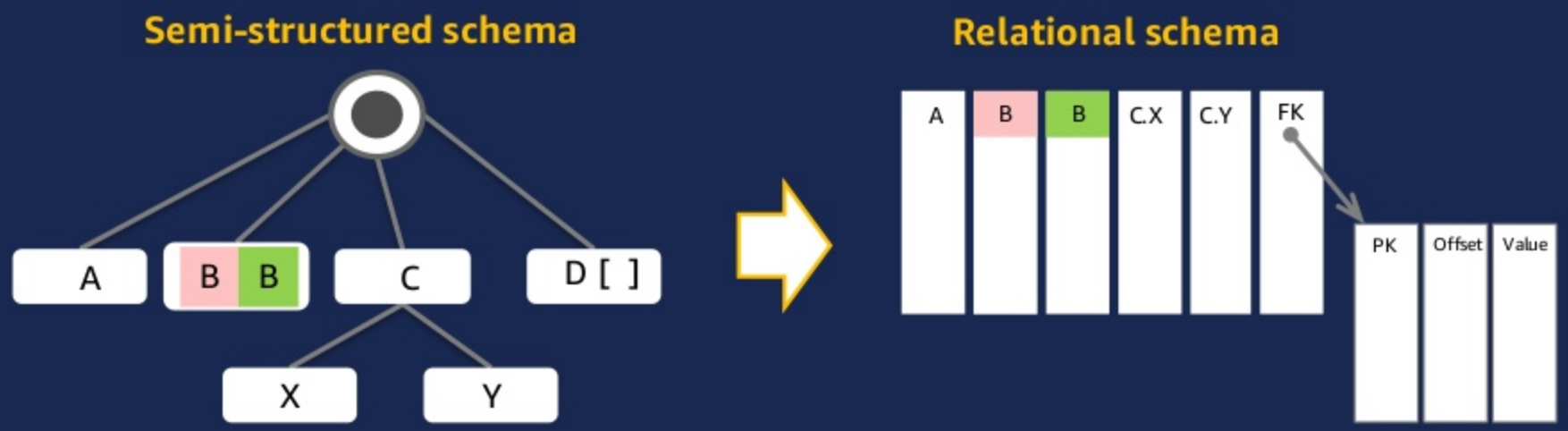
- Tracks new fields, and inconsistent changing data types with **choices**, e.g. integer or string
- Automatically mark and separate error records

Job Authoring: Glue Transforms

ResolveChoice()

Apply Mapping()

Job Authoring: Relationalize() Transform



Job Authoring: ETL Code

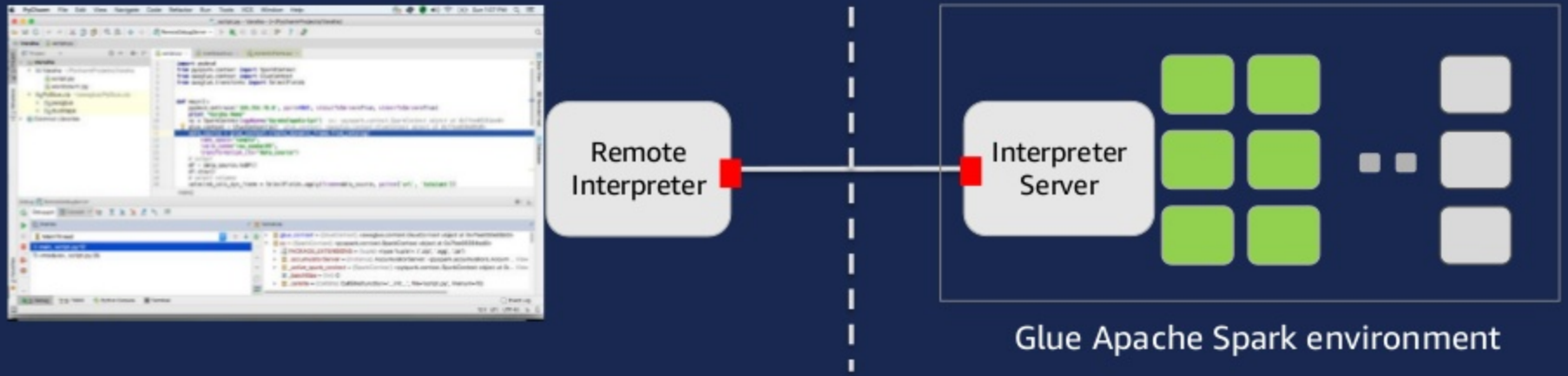
- **Human-readable**, editable, and portable PySpark code
- **Flexible**: Glue's ETL library simplifies manipulating complex, semi-structured data
- **Customisable:** Use native PySpark, import custom libraries, and/or leverage Glue's libraries

```
#convert to a Spark DataFrame...
customDF = dropnullfields3.toDF()

#add a new column for "type"
customDF = customDF.withColumn("type", lit('yellow'))

# Convert back to a DynamicFrame for further processing.
customDynamicFrame = DynamicFrame.fromDF(customDF, glueContext, "customDF_df")
#----------------------------------
```

Job Authoring: Developer Endpoints



- Environment to **iteratively develop** and test ETL code.
- Connect your IDE or notebook (e.g. **Zeppelin**) to a Glue development endpoint.
- When you are satisfied with the results you can create an ETL job that runs your code.

Job Authoring: ETL Code

- **Human-readable**, editable, and portable PySpark code
- **Flexible**: Glue's ETL library simplifies manipulating complex, semi-structured data
- **Customisable:** Use native PySpark, import custom libraries, and/or leverage Glue's libraries
- **Collaborative:** share code snippets via GitHub, reuse code across jobs

How do I run ETL jobs?

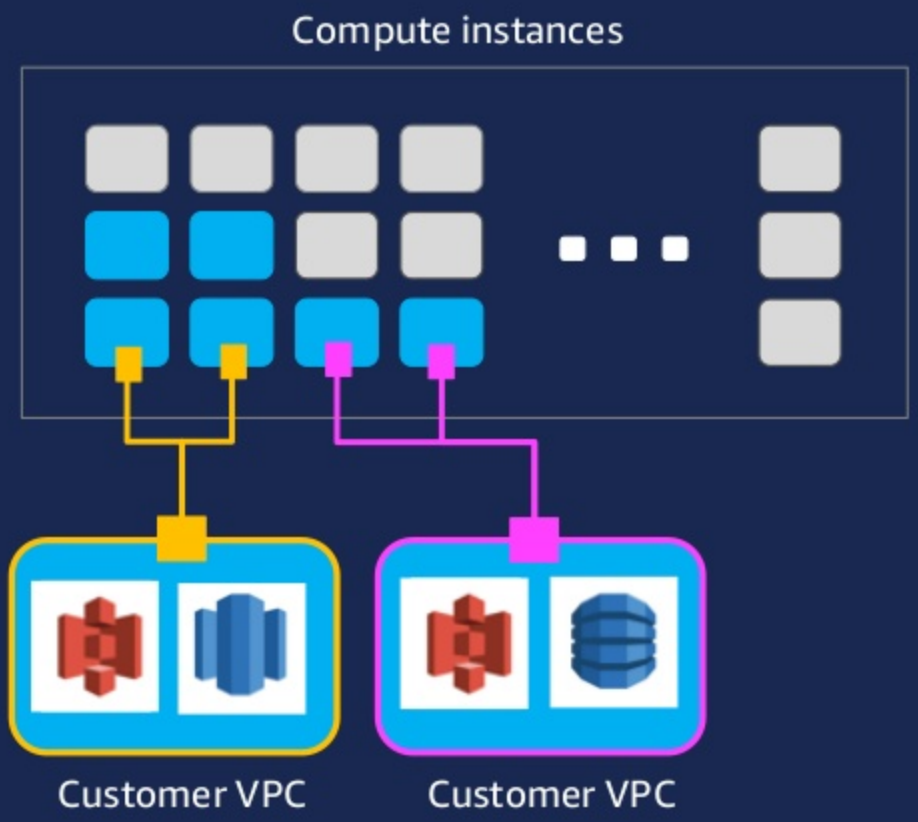
Serverless Job Execution

Auto-configure VPC & role-based access
security & isolation preserved

Customers can specify job capacity
using Data Processing Units (DPU)

Automatically scale resources

Only pay for the resources you consume
per-second billing (10-minute min)



Data Processing Units (DPUs)

1 DPU = 4 vCPU + 16GB RAM

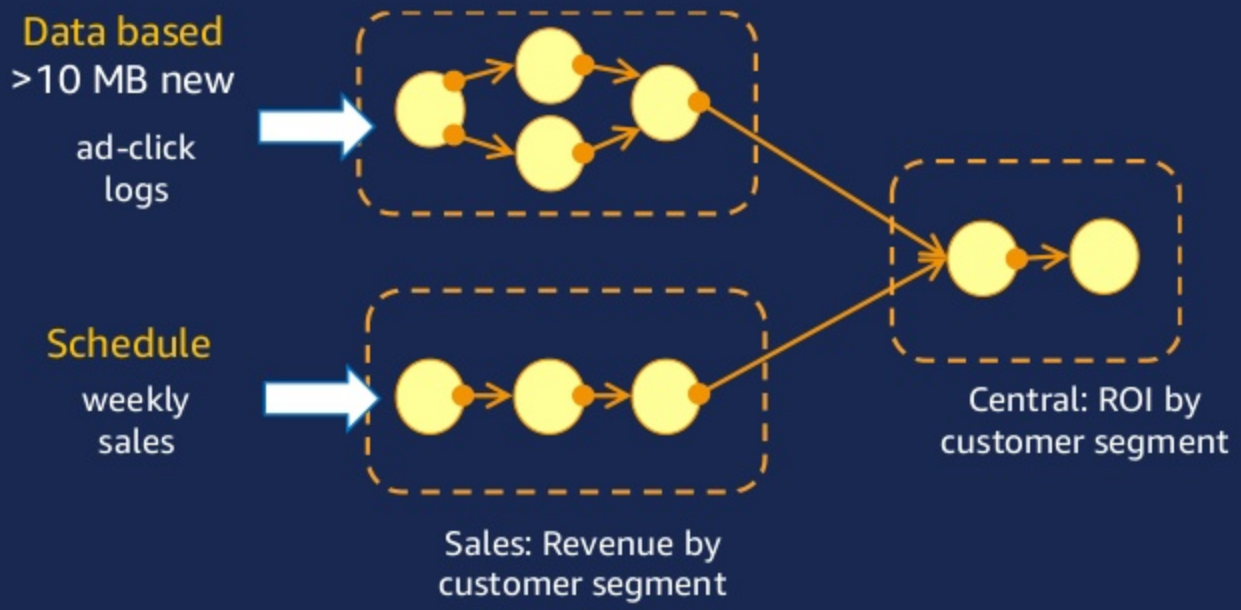
Storage:

- Free for the first million objects stored
- $1 per 100,000 objects stored above 1M, per month

Requests:

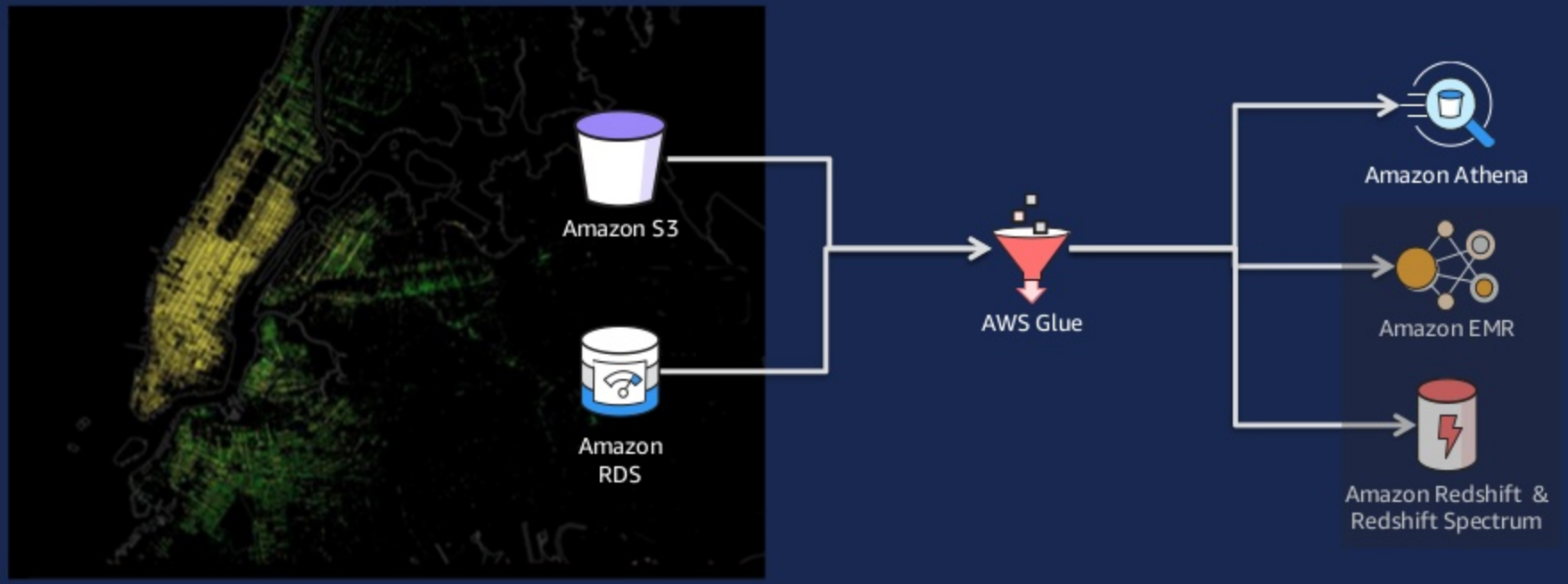
- Free for the first million requests per month
- $1 per million requests above 1M in a month

Job Composition: Example



Sounds Good In Theory…
What's It Really Like?

Demo context



How AWS Glue Helps with ETL

Automatically discover your data

Generate ETL code

Run your ETL jobs serverless

Speaker Contact

Ben Thurgood
Principal Solutions Architect
btgood@amazon.com

aws SUMMIT

Thank You