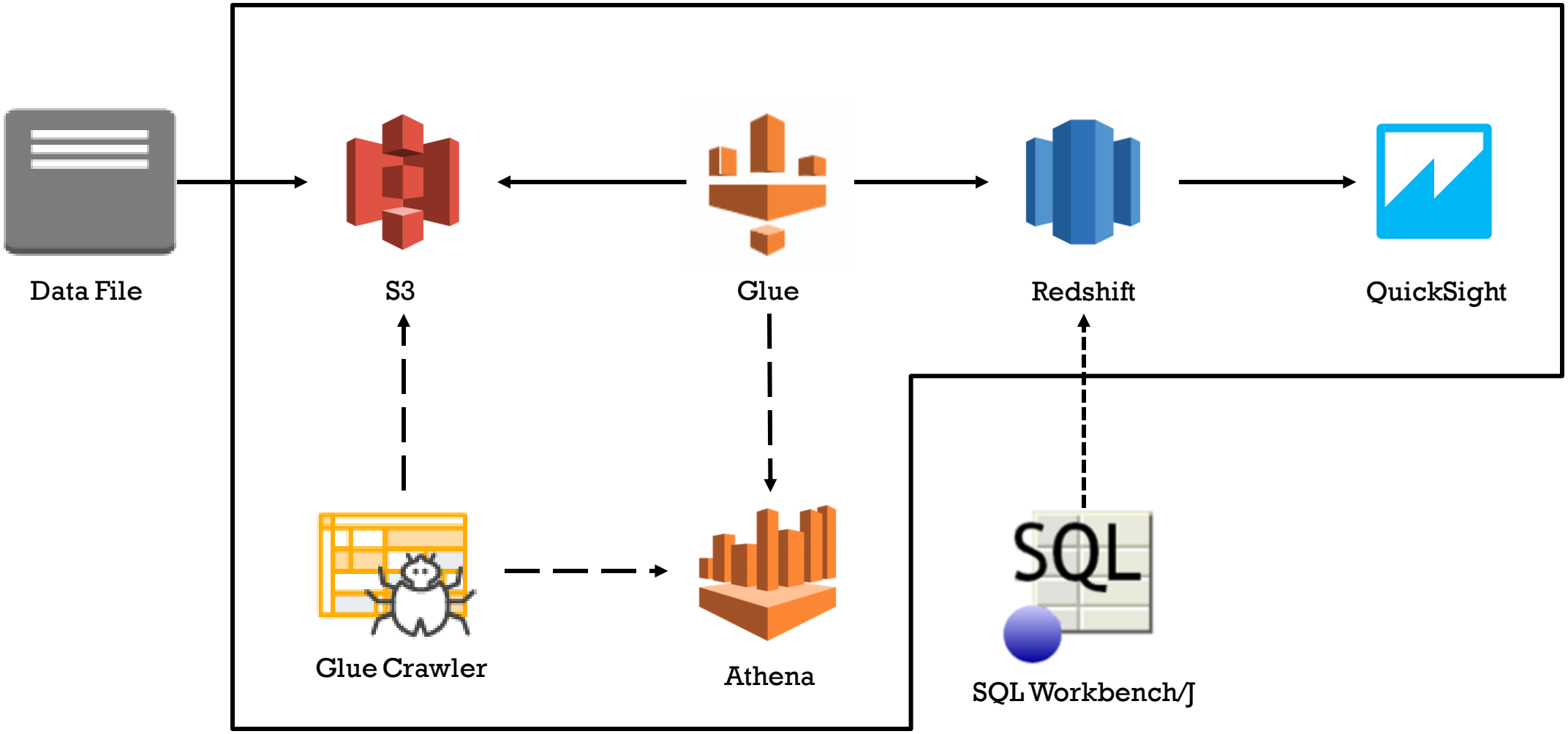
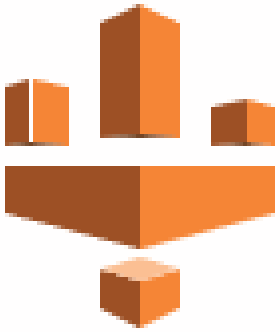
ARTS & CRAFTS WITH AWS GLUE

Lydia White and James Zhang

Amazon Web Services

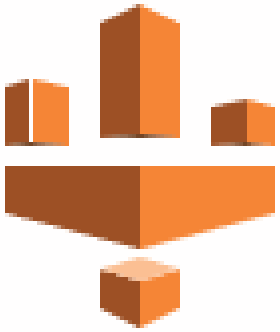


What is Glue?



AWS Glue

- Extract, Transform, and Load(ETL) tool by Amazon Web Services
- Used to prepare data for business analytics



ETL

- Extract: Pull data from a source
  - Files
  - Database
  - Reporting Tool
- Transform: Modify the data to fit your needs
  - Add new columns (data source, timestamp, etc.)
  - Remove unwanted data
  - Alter existing data
- Load: Store in your database

ETL

Original Data File

	A	B	C	D	E	F	G	H	I	J	K
1	Retailer country	Order method type	Retailer type	Product line	Product type	Product	Year	Quarter	Revenue	Quantity	Gross margin
2	United States	Fax	Outdoors Shop	Camping Equipment	Cooking Gear	TrailChef Deluxe C	2012	Q1 2012	59628.66	489	0.347548
3	United States	Fax	Outdoors Shop	Camping Equipment	Cooking Gear	TrailChef Double F	2012	Q1 2012	35950.32	252	0.474275
4	United States	Fax	Outdoors Shop	Camping Equipment	Tents	Star Dome	2012	Q1 2012	89940.48	147	0.352772
5	United States	Fax	Outdoors Shop	Camping Equipment	Tents	Star Gazer 2	2012	Q1 2012	165883.4	303	0.282938
6	United States	Fax	Outdoors Shop	Camping Equipment	Sleeping Bags	Hibernator Lite	2012	Q1 2012	119822.2	1415	0.29145
7	United States	Fax	Outdoors Shop	Camping Equipment	Sleeping Bags	Hibernator Extrem	2012	Q1 2012	87728.96	352	0.398146
8	United States	Fax	Outdoors Shop	Camping Equipment	Sleeping Bags	Hibernator Camp C	2012	Q1 2012	41837.46	426	0.335607
9	United States	Fax	Outdoors Shop	Camping Equipment	Lanterns	Firefly Lite	2012	Q1 2012	8268.41	577	0.52896
10	United States	Fax	Outdoors Shop	Camping Equipment	Lanterns	Firefly Extreme	2012	Q1 2012	9393.3	189	0.434205
11	United States	Fax	Outdoors Shop	Camping Equipment	Lanterns	EverGlow Single	2012	Q1 2012	19396.5	579	0.461493
12	United States	Fax	Outdoors Shop	Camping Equipment	Lanterns	EverGlow Butane	2012	Q1 2012	6940.03	109	0.361866
13	United States	Fax	Outdoors Shop	Mountaineering Equip	Rope	Husky Rope 50	2012	Q1 2012	20003.2	133	0.329056
14	United States	Fax	Outdoors Shop	Mountaineering Equip	Rope	Husky Rope 60	2012	Q1 2012	14109.4	79	0.291657
15	United States	Fax	Outdoors Shop	Mountaineering Equip	Rope	Husky Rope 100	2012	Q1 2012	73970.22	227	0.301264

Example Business Requirements:

- Remove the Year from Quarter
- Add a profit column from revenue * gross margin columns
- Add a current date column

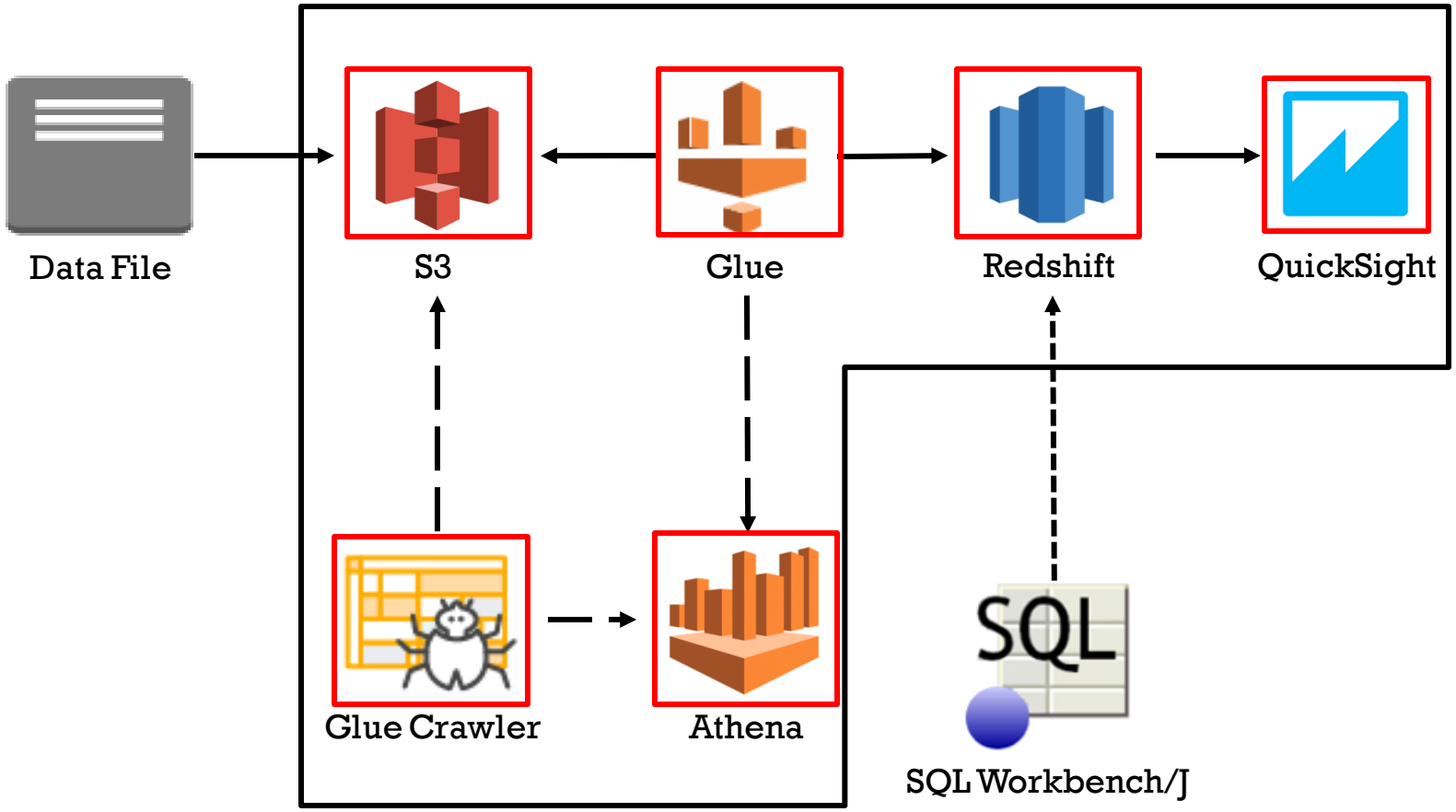
Why use Glue?

- Serverless
  - companies do not have to invest and maintain on premise servers
- Easily scalable
  - adjust storage needs up and down based on need
- Cost Effective – Glue is cheaper than other ETL Services
  - Only pay when being used, where Matillion and Informatica charge hourly or yearly
  - Matillion: $2.74 per hour (m4.large EC2), Informatica $3.66 per hour (m4.large EC2), Glue $0.44 per DPU-Hour
- Code based (Python or Scala) so you can do anything you can program
- Easy integration with other AWS tools
- Automatic error handling and logging

AWS vs. Hadoop

Hadoop – A popular platform used to store and transform big data

- AWS is more flexible – scale up or down storage based on need
- AWS is less complex – no need to set up and maintain servers
- AWS cheaper
  - No start up cost
  - No maintenance cost
  - Pay as you go
- Hadoop has challenges handling a lot of small files
- AWS – End to End solution for data needs
  - Storage
  - Transform
  - Business Intelligence
- ETL(AWS) vs. ELT(Hadoop)
- Durability
  - Data stored in multiple locations within region
  - If a location fails data is still available

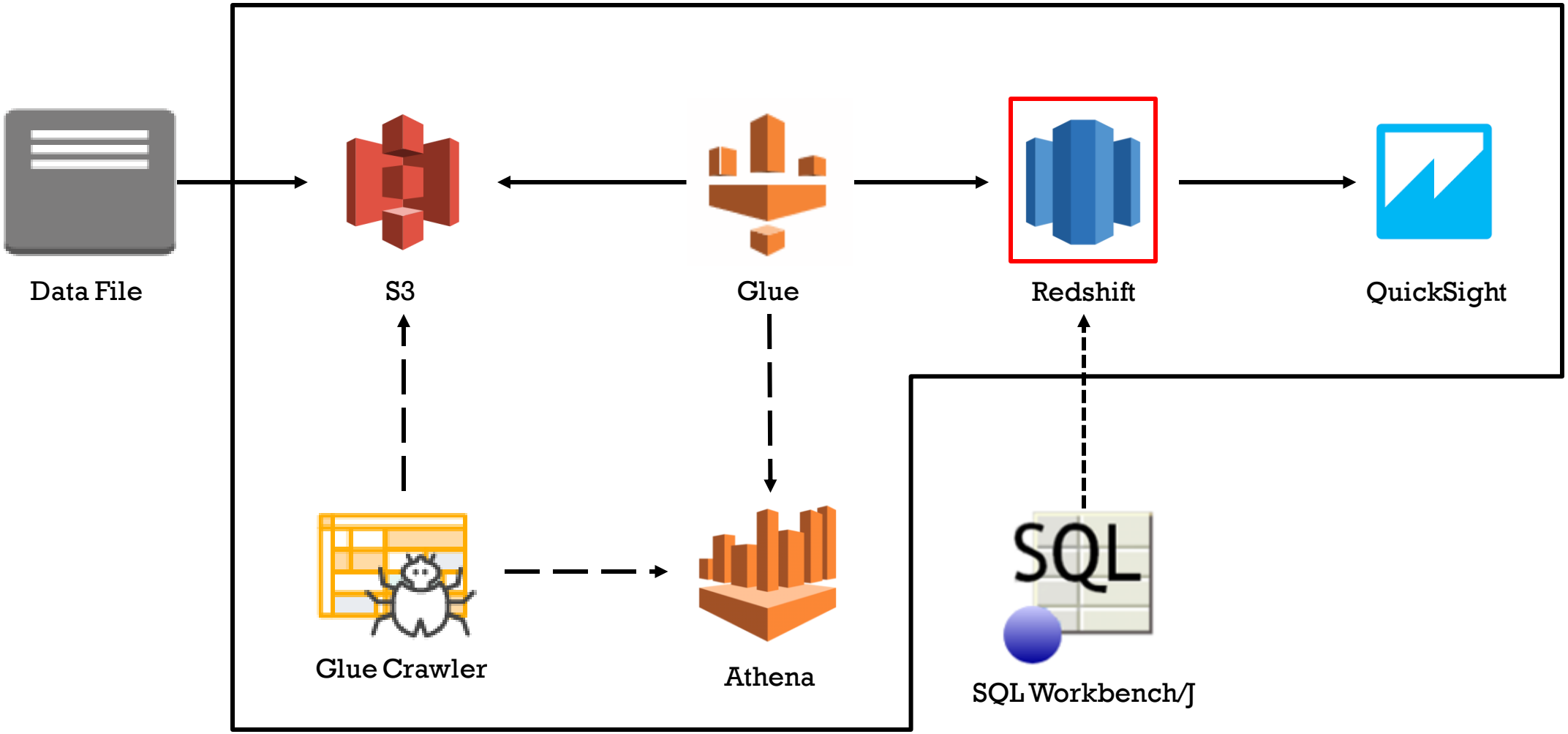


GLUE TUTORIAL OVERVIEW

- Setup Redshift Cluster
- Create Redshift table
- S3 bucket for storing the file
- Athena table to access data in file
- Glue connection
- Glue job
- Run Glue job
- QuickSight

Redshift

Create AWS Data Warehouse



Redshift

Create AWS Data Warehouse

Redshift dashboard

Clusters

Snapshots

Security

Parameter groups

Workload management

Reserved nodes

Advisor Beta

Events

Connect client

What's new

Launch cluster

Amazon Redshift is a powerful, fully managed cloud data warehouse service. Redshift Spectrum extends the power of Redshift to query unstructured data in S3 – without loading your data into Redshift. With a few clicks in the AWS Management Console, you can launch a Redshift cluster and get started analyzing your data.

Quick launch cluster **Launch cluster**

Note: Your cluster will launch in the EU West (Ireland) region

Resources

You are using the following Amazon Redshift resources in the EU West (Ireland) region (used):

Clusters (0)
Increase cluster limit
Snapshots (0)
Manual (0)
Automated (0)

Security
Subnet groups (1)

Parameter groups (0)
Total Reservations (0)
Events (0)
Event subscriptions (0)

Service health

Current Status	Details
Amazon Redshift (Ireland)	Service is operating normally

› View complete service health details

Redshift

Create AWS Data Warehouse

Redshift dashboard
Clusters
Snapshots
Security
Parameter groups
Workload management
Reserved nodes
Events
Connect client
What's new

Launch your Amazon Redshift cluster - Advanced settings | Switch to quick launch

CLUSTER DETAILS ✓ NODE CONFIGURATION ✓ ADDITIONAL CONFIGURATION ✓ REVIEW

You are about to launch a cluster with following the following specifications:

Cluster properties

These attributes specify the name of your cluster, what type of virtual hardware it will run on, how many nodes it will contain, and the availability zone in which it will be located.

Cluster identifier: glue-tutorial-xxx
Node type: dc2.large
Number of compute nodes: 1 (leader and compute run on a single node)
Availability zone: us-east-2a

Database configuration

These properties specify the database name, port, and username you will use to connect to the database. The parameter group contains configuration values used by the database.

Database name: glue_tutorial_database_xxx
Database port: 5439
Master user name: master
Cluster parameter group: A default parameter group will be created when the cluster is launched.

Security, access, and encryption

These settings control whether your cluster will be created in an existing VPC to allow for simpler integration with other AWS Services, and the security groups which define access rules to your cluster.

Virtual private cloud: vpc-b2fb56da
Cluster subnet group:
Publicly accessible: Yes
Elastic IP: Not used
VPC security groups default (sg-797ba212)
Enhanced VPC Routing: No
Encrypt database: No

CloudWatch alarms

CloudWatch alarms are used to notify if metrics for your cluster are within a certain threshold. All recipients under the SNS topic specified for your alarm will receive notifications once an alarm is triggered.

Basic alarms will not be created for this cluster

Redshift

Create AWS Data Warehouse

Unless you are eligible for the free trial, you will start accruing charges as soon as your cluster is active.

Applicable charges:
The on-demand hourly rate for this cluster will be **$0.30** , or **$0.30** /node. If you have purchased reserved nodes in this region for this node type that are active, your costs will be discounted. Additional nodes will be billed at the on-demand rate.

If you are eligible for a free trial, you will receive 750 hours of free usage for each month of the trial, applied across all running dc2.large nodes across all regions. Regardless of when you start your trial, you will receive two full months of free usage. Once your trial expires or your usage exceeds 750 hours/month, you can shut down your cluster, avoiding any charges, or keep it running at our standard On-demand rate .

For more information, see Amazon Redshift Free Trial FAQ , Amazon Redshift Pricing , and Reserved Nodes Documentation .

Cancel | Previous | Launch cluster

Redshift

Create AWS Data Warehouse

Clusters

Quick launch cluster | Launch cluster | Cluster | Database | Backup | Manage Tags | Manage IAM roles

Cluster	Cluster Status	DB Health	Release Status	In Maintenance	Recent Events
glue-tutorial-xxx	available	healthy	Up to date	no	1

Endpoint glue-tutorial-xxx.c5ytrmxef4xv.us-east-2.redshift.amazonaws.com:5439 (authorized)

Cluster Properties

Cluster Name	glue-tutorial-xxx
Node Type	dc2.large
Nodes	1
Zone	us-east-2a
Cluster Parameter Group	default.redshift-1.0 (in-sync)
Cluster Subnet Group	default
Enhanced VPC Routing	No
IAM Roles	See IAM Roles

Cluster Status

Cluster Status	available
Database Health	healthy
In Maintenance Mode	no
Parameter Group Apply Status	in-sync
Pending Modified Values	None

Cluster Database Properties

Port	5439
Database Name	glue_tutorial_database_xxx
Master Username	master
Encrypted	No

Backup, Audit Logging, and Maintenance

Automated Snapshot Retention Period	1
Cross-Region Snapshots Enabled	No
Audit Logging Enabled	No
Maintenance Window	tue:08:30-tue:09:00
Allow Version Upgrade	Yes

Tags

You have not created any tags. Please add tags using the **Manage Tags** button above.

Redshift

Create table

Create empty table in the Redshift database

SQL Workbench/J GlueTutorial - Default.wksp

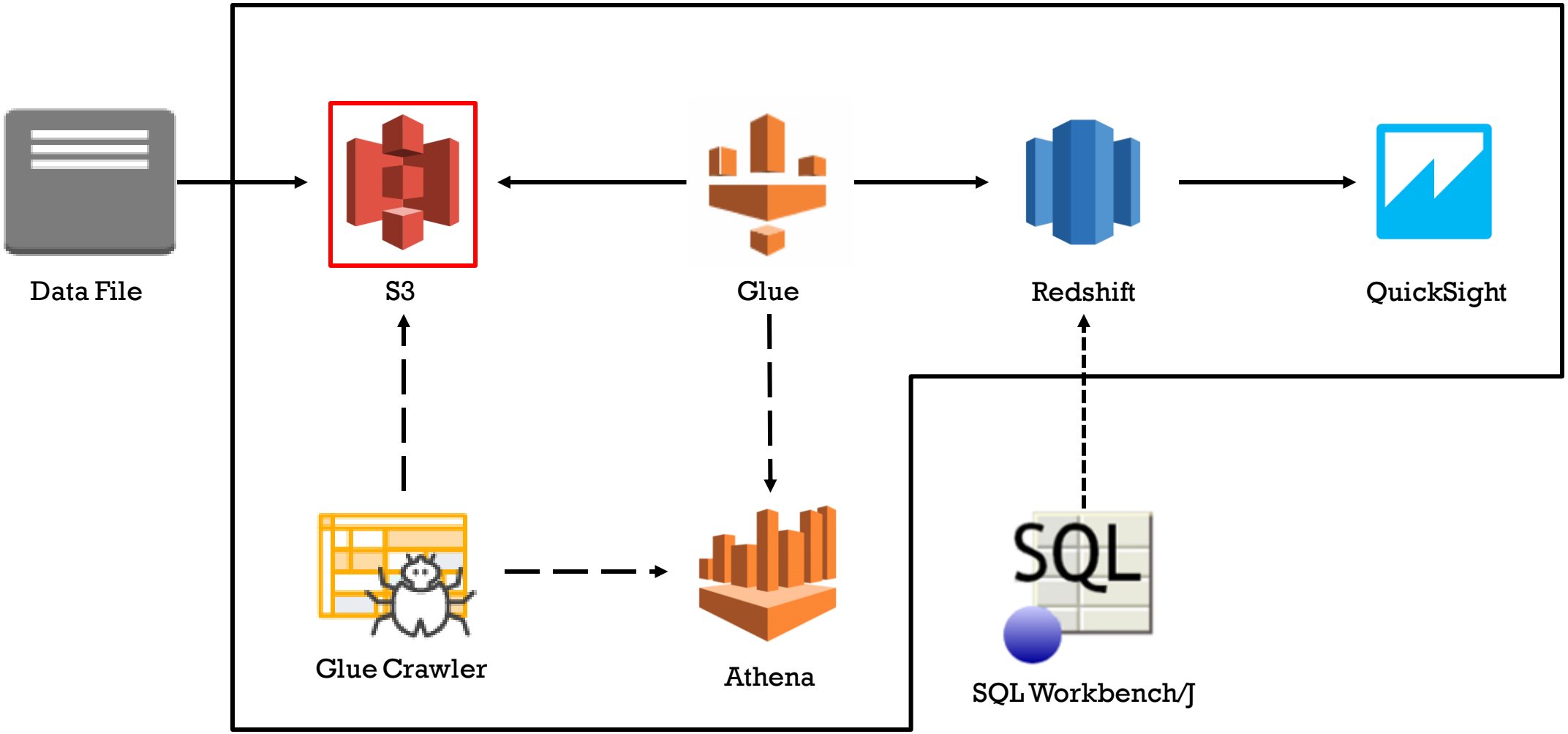
File Edit View Data SQL Macros Workspace Tool

Statement 1 Database Explorer 2

```
CREATE SCHEMA sales_XXX;

CREATE TABLE sales_XXX.products_XXX
(
   retailer_country   varchar(20),
   order_method_type  varchar(15),
   retailer_type      varchar(30),
   product_line       varchar(30),
   product_type       varchar(30),
   product            varchar(50),
   year               varchar(4),
   quarter            varchar(2),
   revenue            numeric(15,2),
   quantity           integer,
   gross_margin       numeric(15,10),
   profit             numeric(15,2),
   timestamp          date
);
```

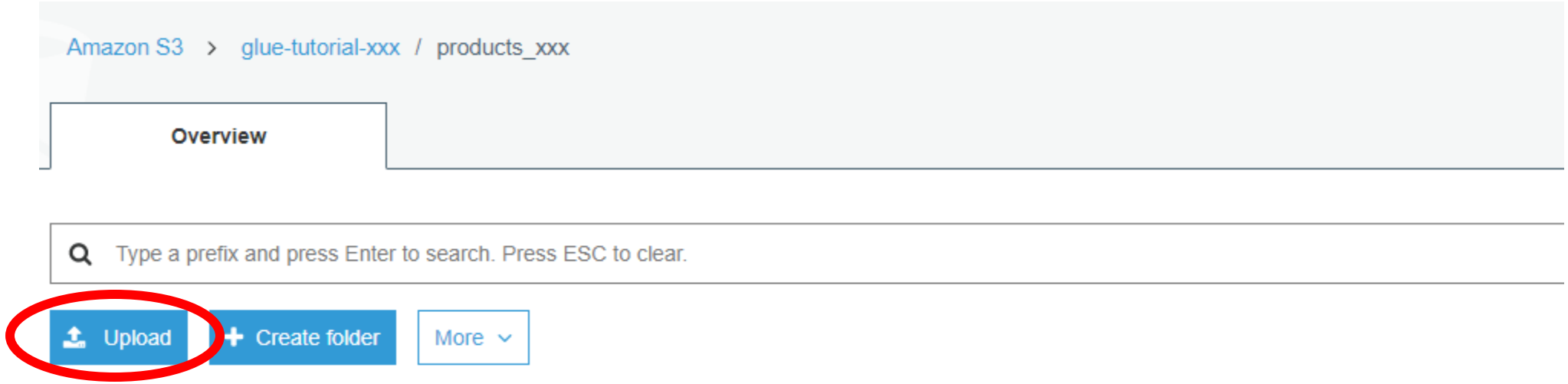
S3

Create S3 bucket with AWS Console



S3

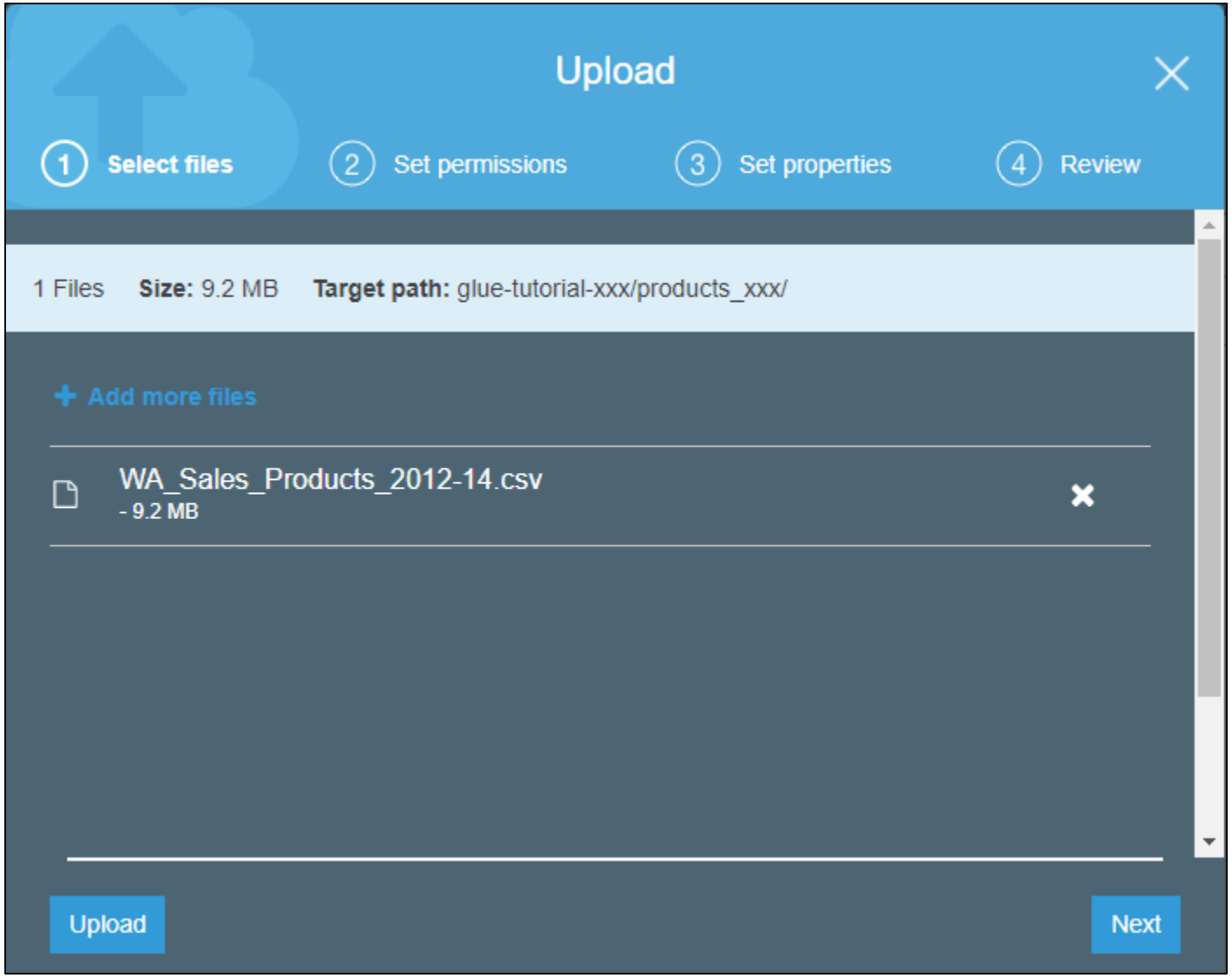
Add file to S3 bucket with AWS Console



S3

Add file to S3 bucket with AWS Console

Add file from repository called "WA_Sales_Products_2012-14"



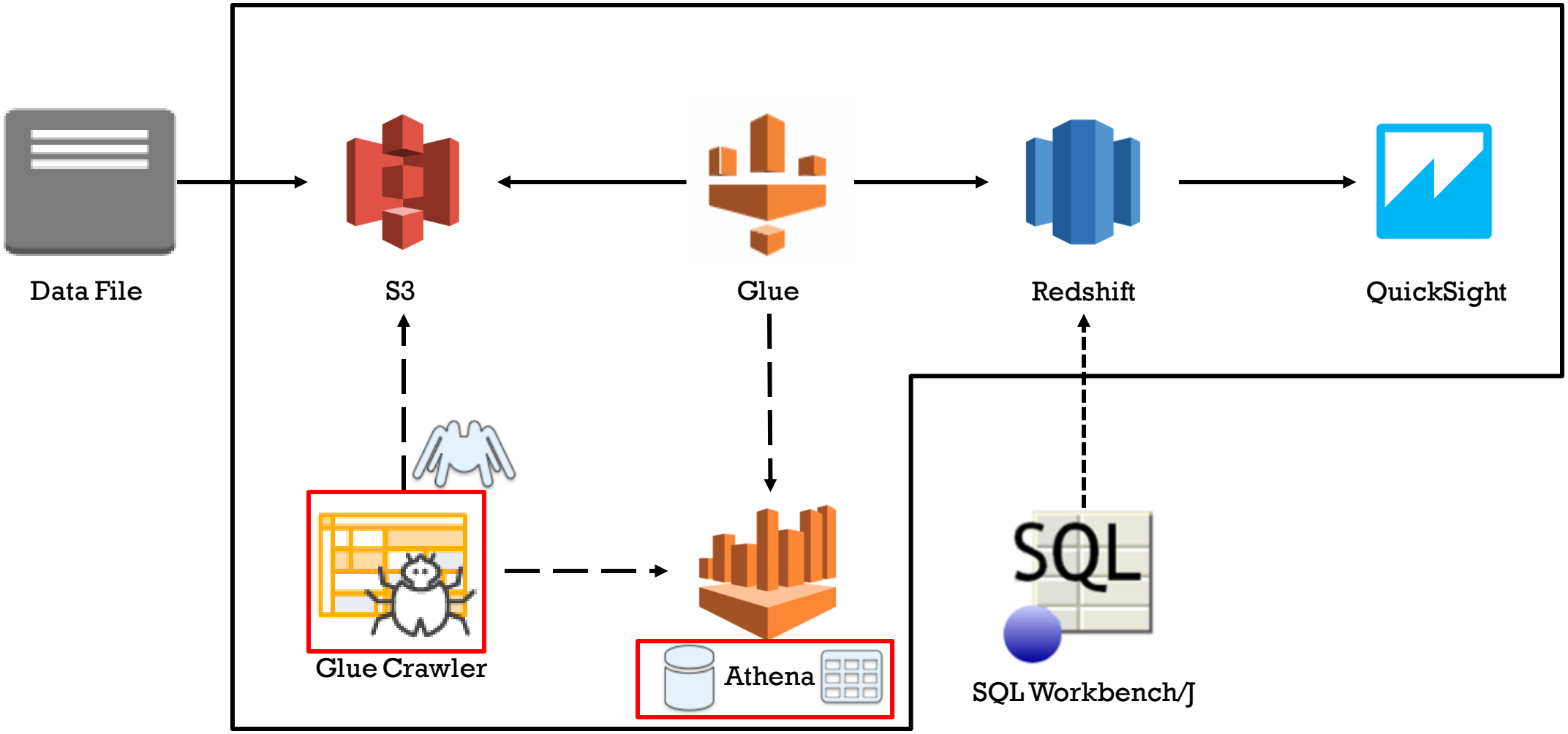
S3

Add file to S3 bucket with AWS CLI* (Alternative)

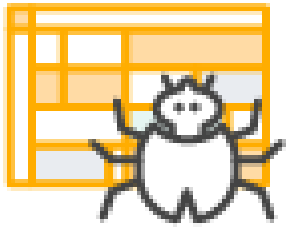
```
$ aws s3 cp <your-file-path>/aws-glue-tutorial/WA_Sales_Products_2012-14.csv s3://glue-tutorial-xxx/products_xxx/
```

* Must install and set up AWS CLI in order to use this

Glue Crawler



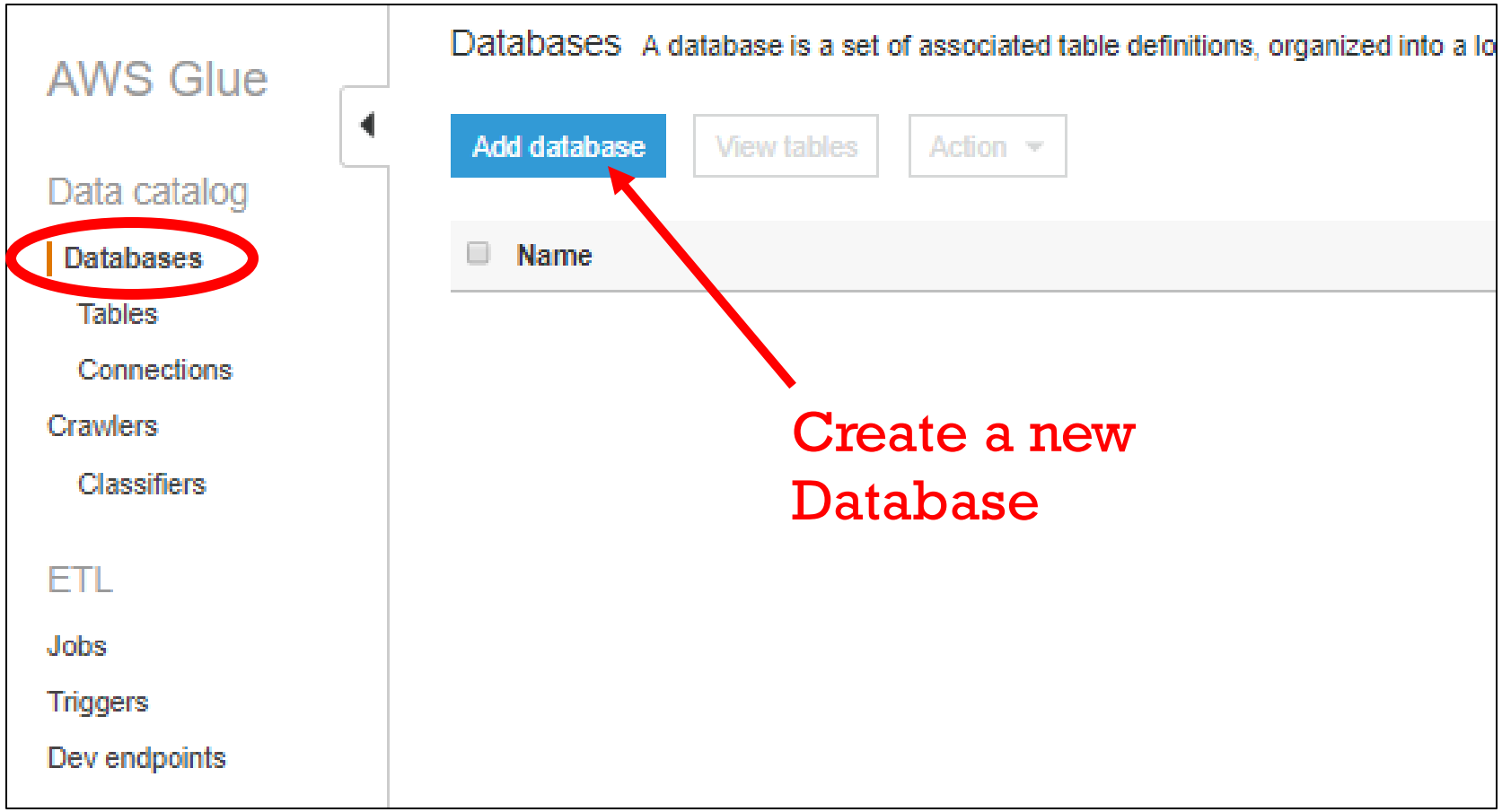
Glue Crawler



- Scans data to create metadata
- Determines column names and data types
  - Creates a Glue Table
  - Queryable with Athena

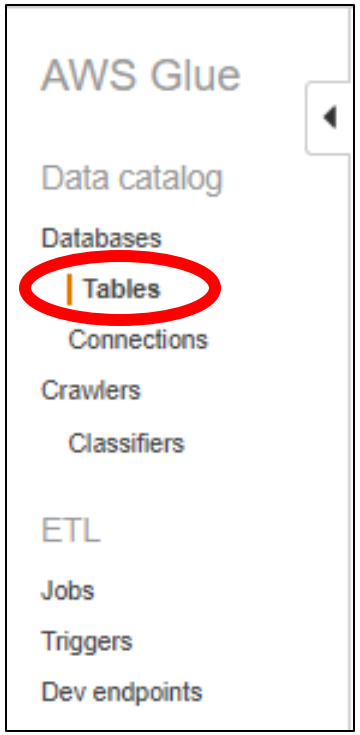
Glue

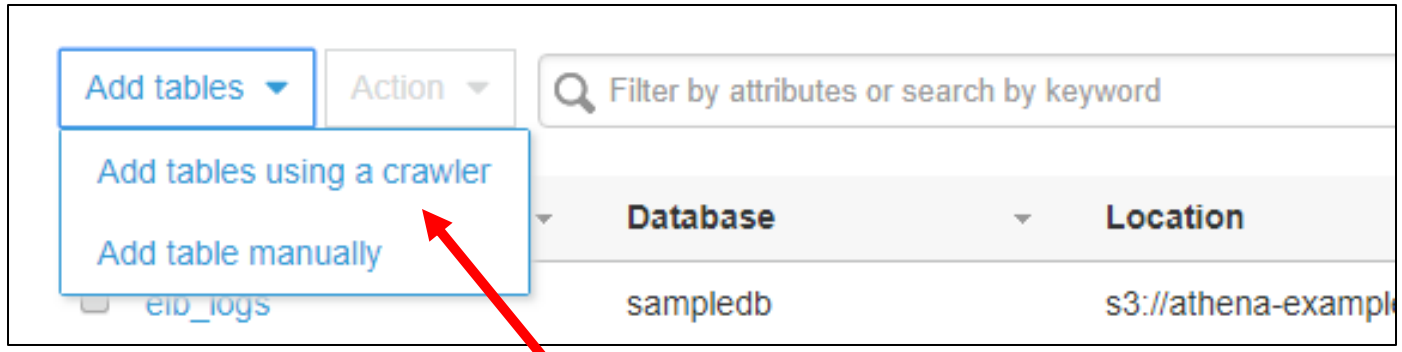
Create Glue Database



Glue Crawler

Create Table with Glue Crawler

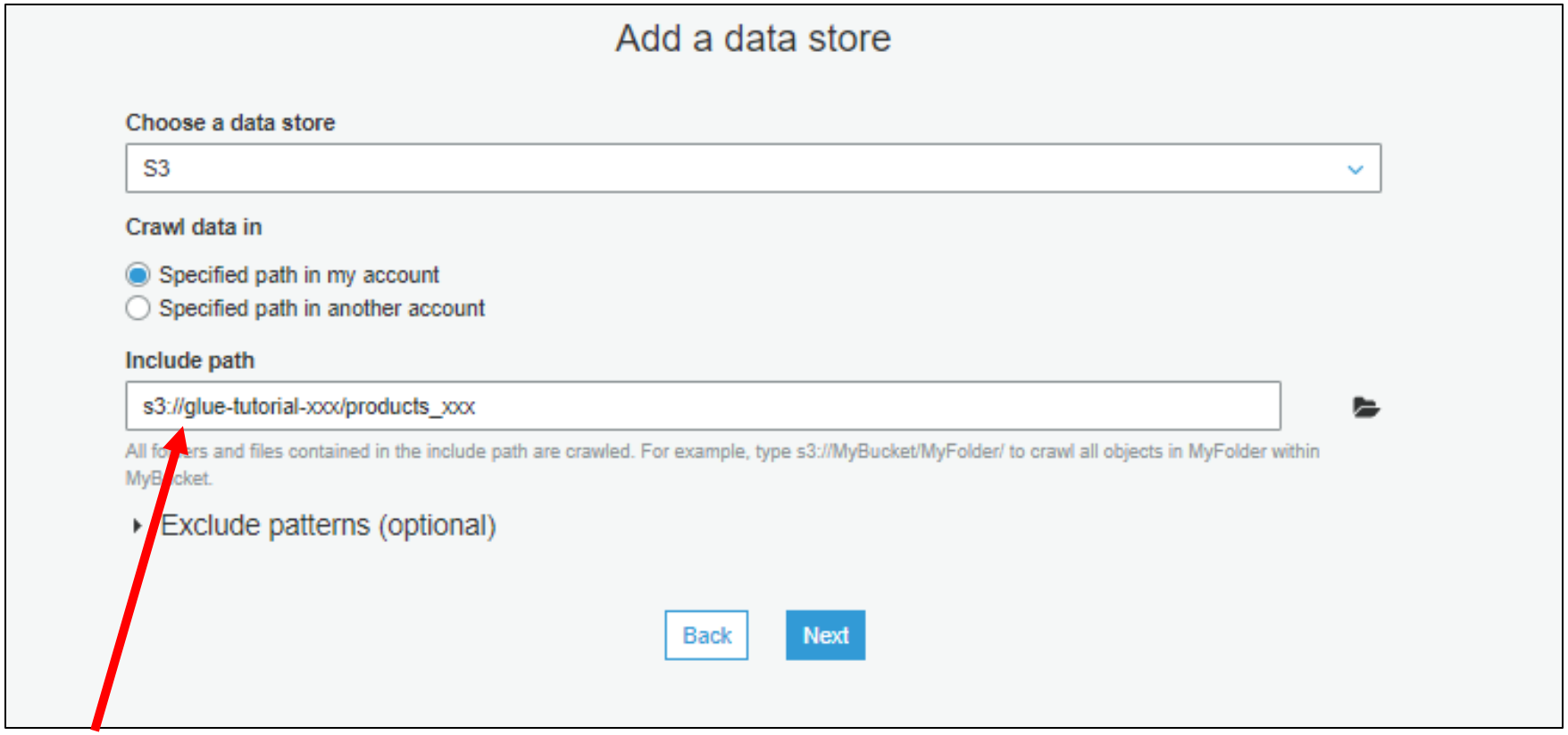




Create a table using a crawler

Glue Crawler

Create Table with Glue Crawler

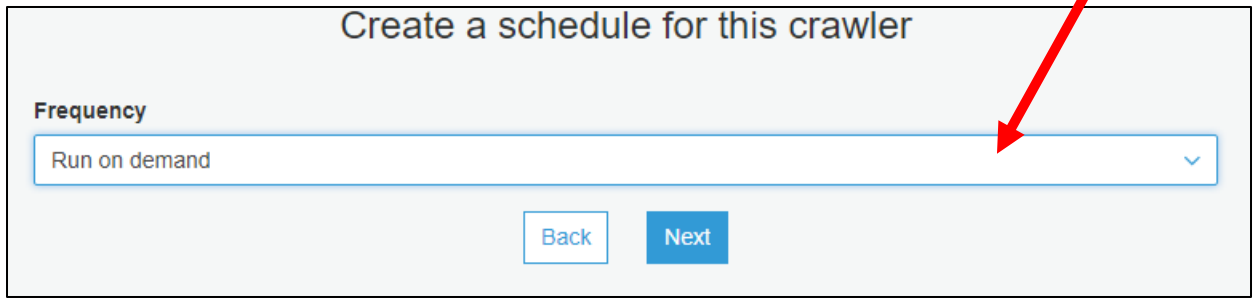


Specify the path for the table to search for in S3

Glue Crawler

Create Table with Glue Crawler

Your crawler can run on either a timed schedule or on demand



Glue Crawler

Create Table with Glue Crawler

Select your crawler

Add crawler | Run crawler | Action | Filter by attributes

Name	Schedule	Status	Logs	Last runtime	Median runtime	Tables updated
glue_tutorial_xxx		Ready		0 secs	0 secs	0

Run your crawler

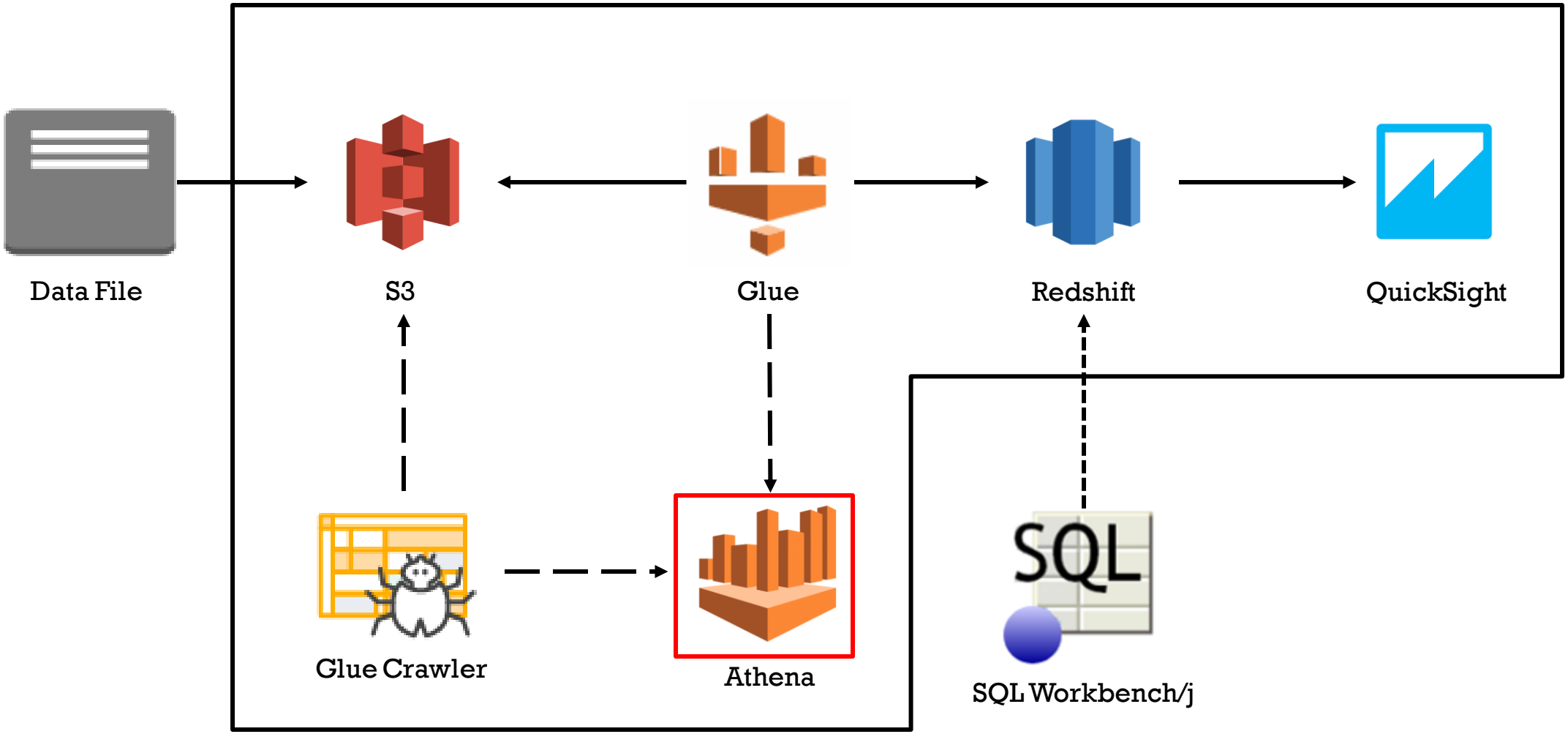
Tables A table is the metadata definition that represents your data, including its schema. A table can be used as a source or target in a job definition.

Add tables | Action | Filter by attributes or search by keyword

Name	Database	Location	Classification	La
products_xxx	glue_database_xxx	s3://glue-tutorial-xxx/products_xxx/	csv	22

Your table should be in the Tables tab

Athena

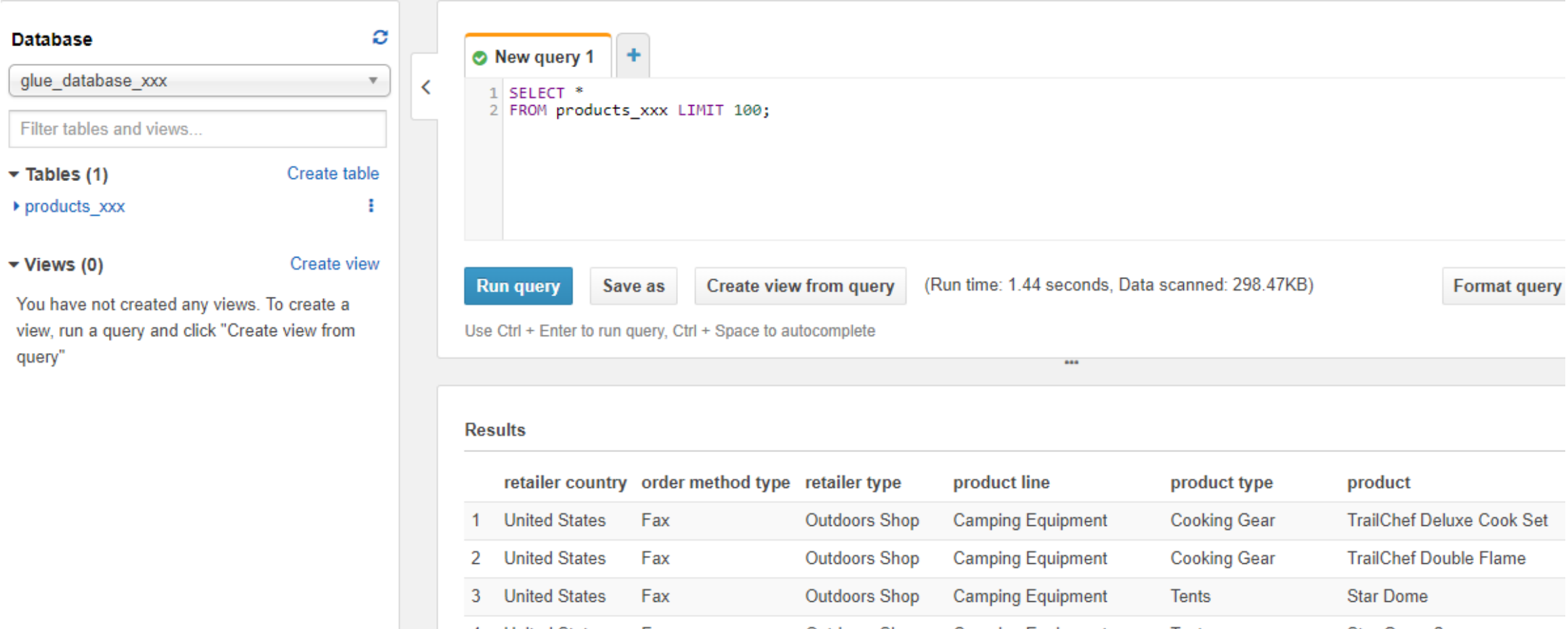

Data File
S3
Glue
Redshift
QuickSight
Glue Crawler
Athena
SQL
SQL Workbench/j

Athena

- Interactive query service used to analyze data
  - Data stored in S3
  - Run queries to verify your data is stored correctly

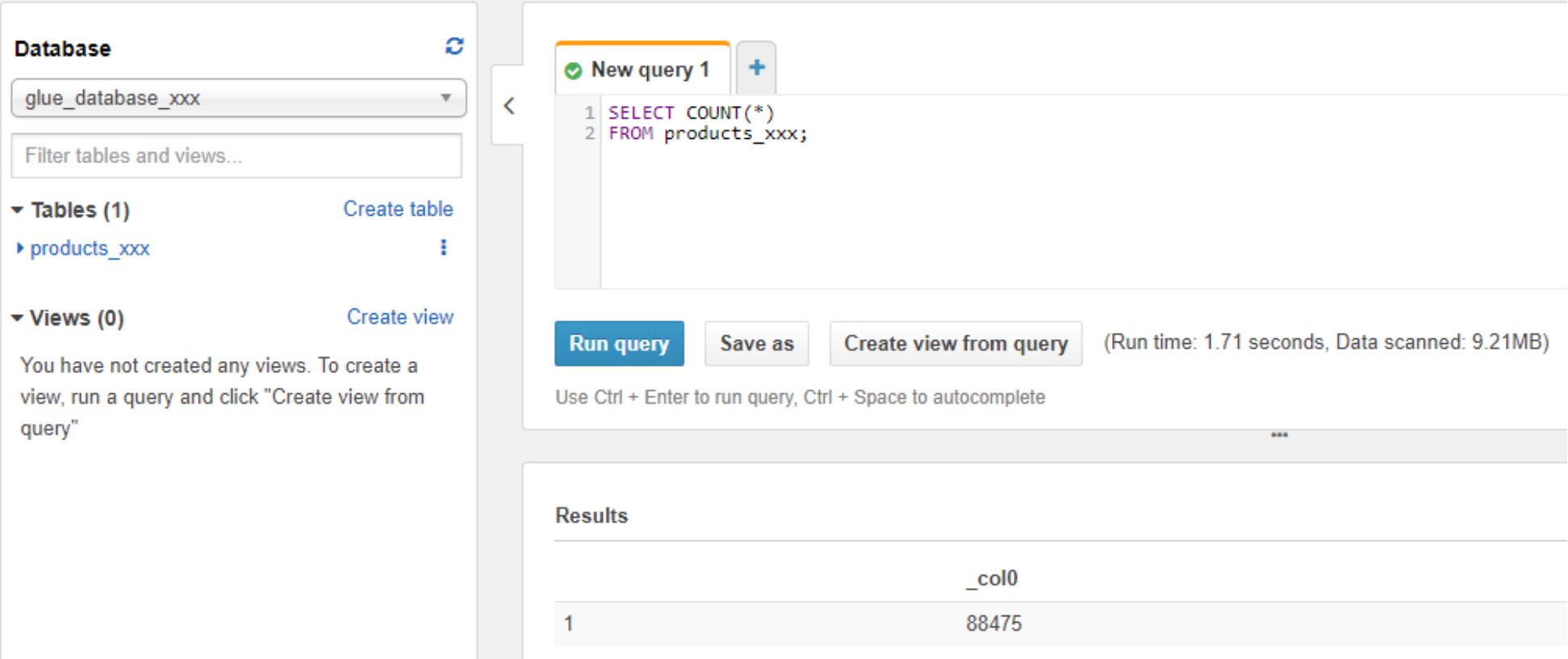
Athena

- Run an SQL select query to verify data populating correctly
- SELECT * FROM products_xxx LIMIT 100;

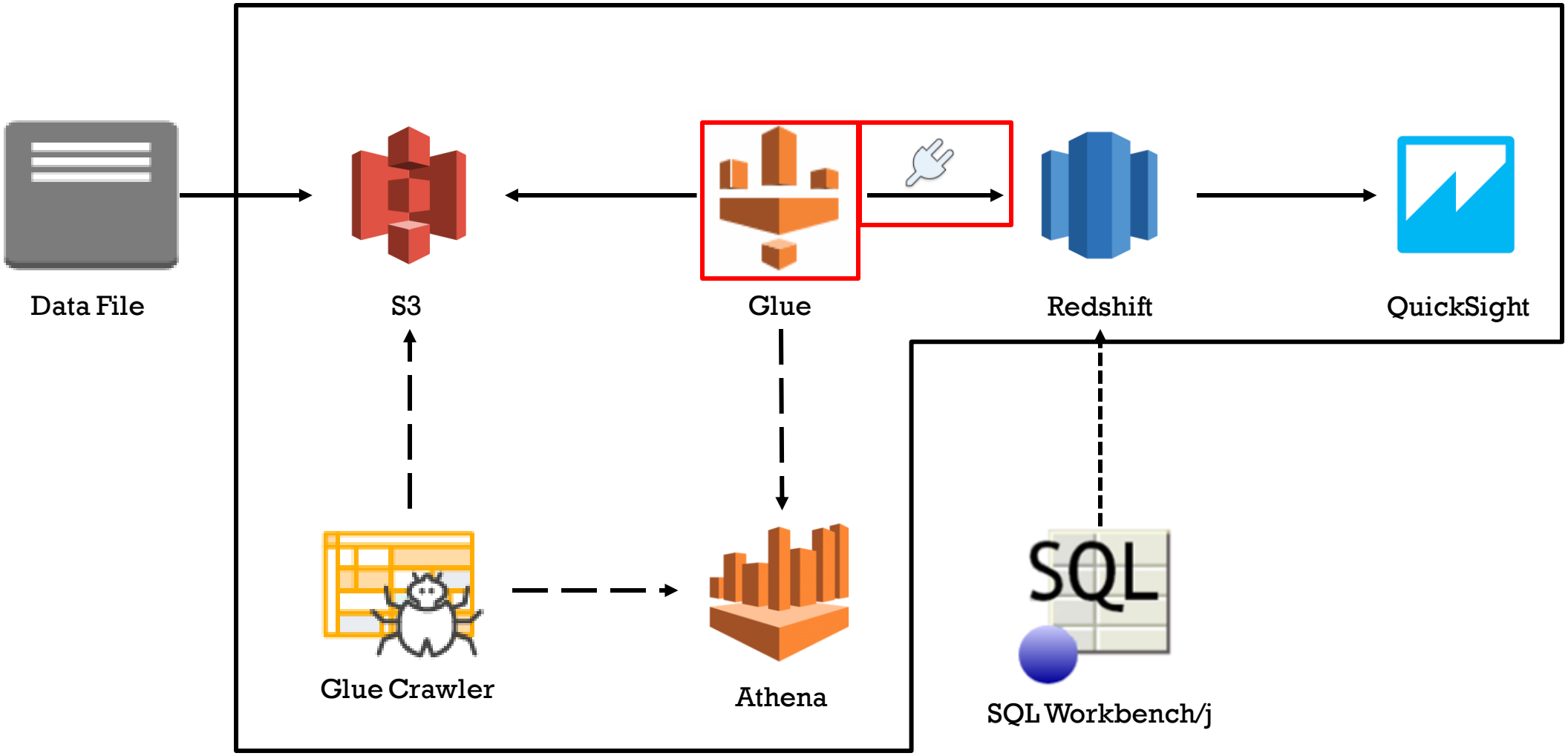


Athena

- Run an SQL count query to verify all data is there
- SELECT COUNT(*) FROM products_xxx;

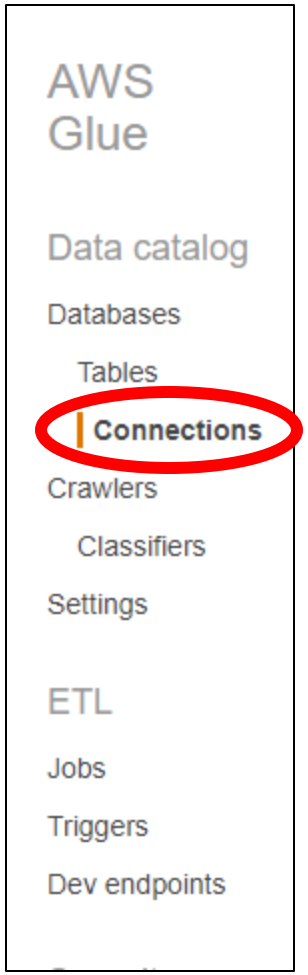


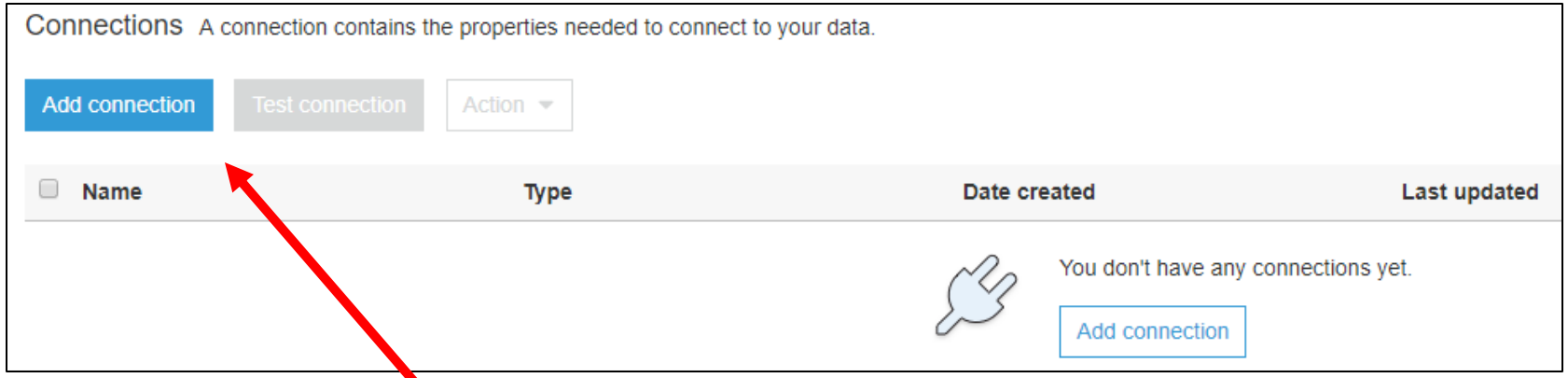
Glue


Data File
S3
Glue
Redshift
QuickSight
Glue Crawler
Athena
SQL
SQL Workbench/j

Glue

Create a connection to Redshift





Click on "Add Connection" to create a connection to the Redshift cluster

Glue

Create a connection to Redshift

Connection properties

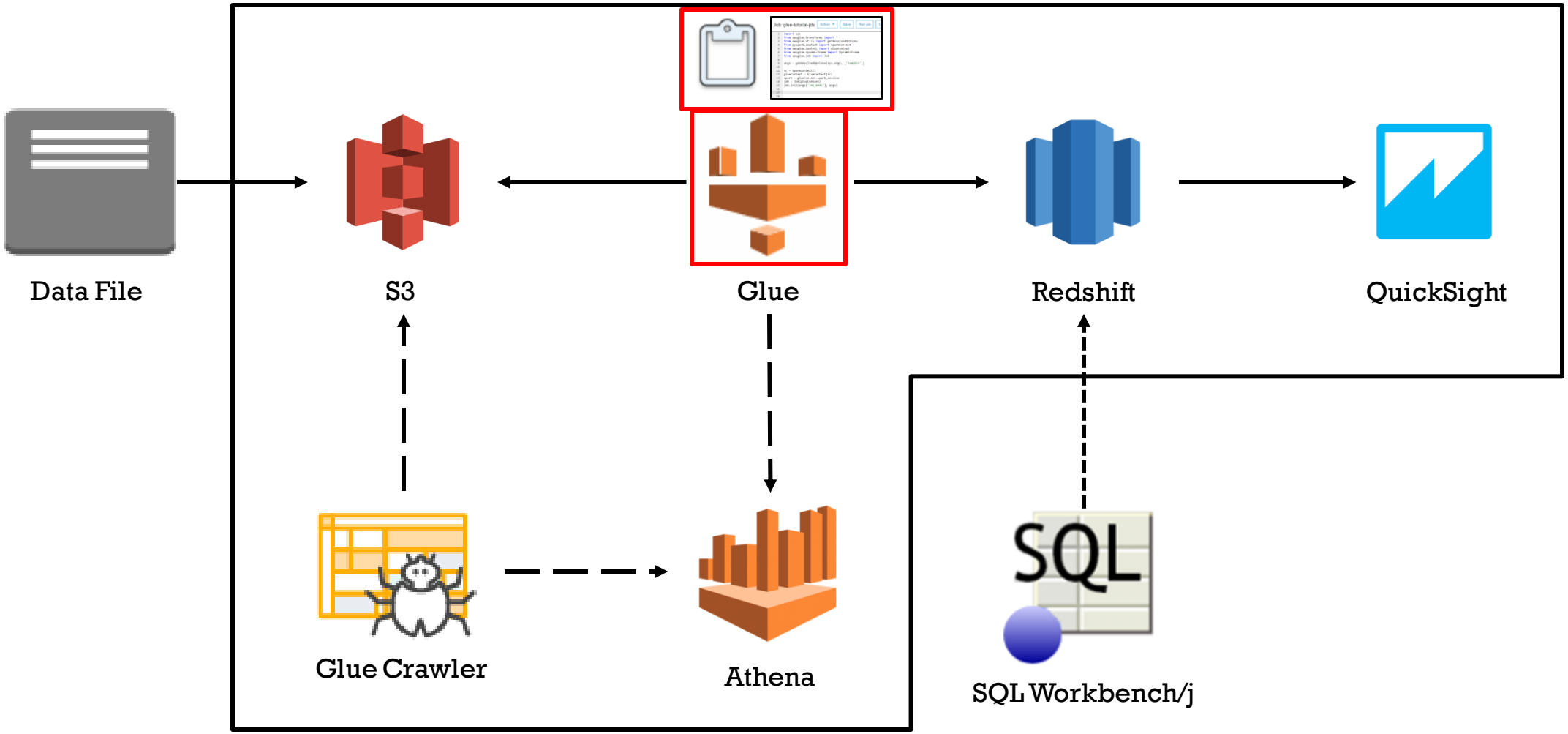
Name	glue_tutorial_xxx
Type	JDBC

Connection access

JDBC URL	jdbc:redshift://glue-tutorial-xxx.c5ytrmxef4xv.us-east-2.redshift.amazonaws.com:5439/glue_tutorial_database_xxx
Username	master
VPC Id	vpc-b2fb56da
Subnet	subnet-c72d85af
Security groups	sg-797ba212

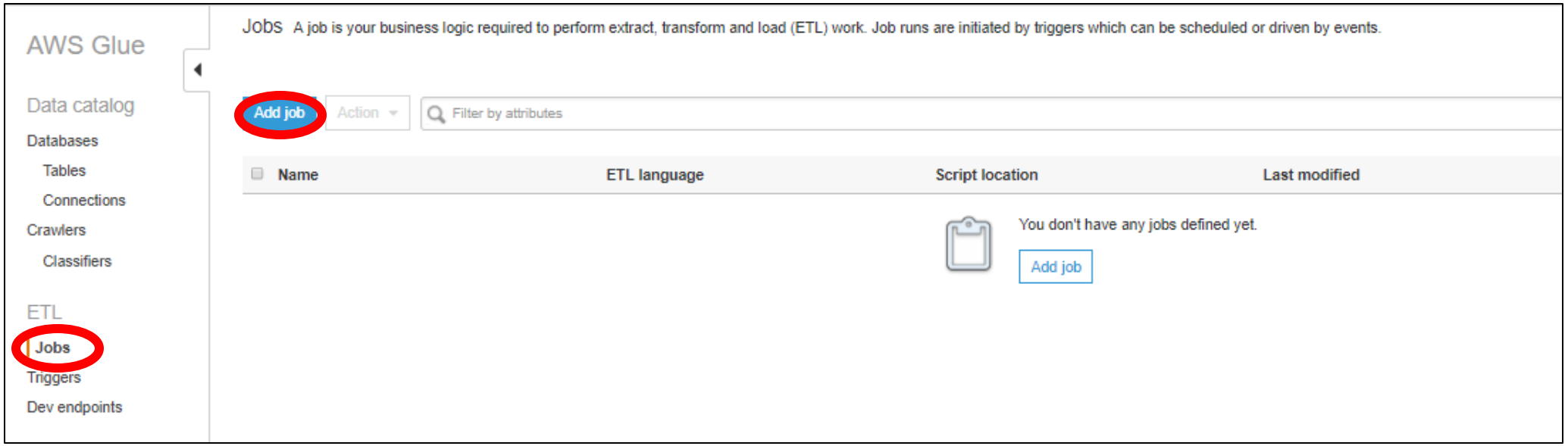
Back Finish

Glue


Data File
S3
Glue
Redshift
QuickSight
Glue Crawler
Athena
SQL
SQL Workbench/j

Glue

Create a Glue job



Glue

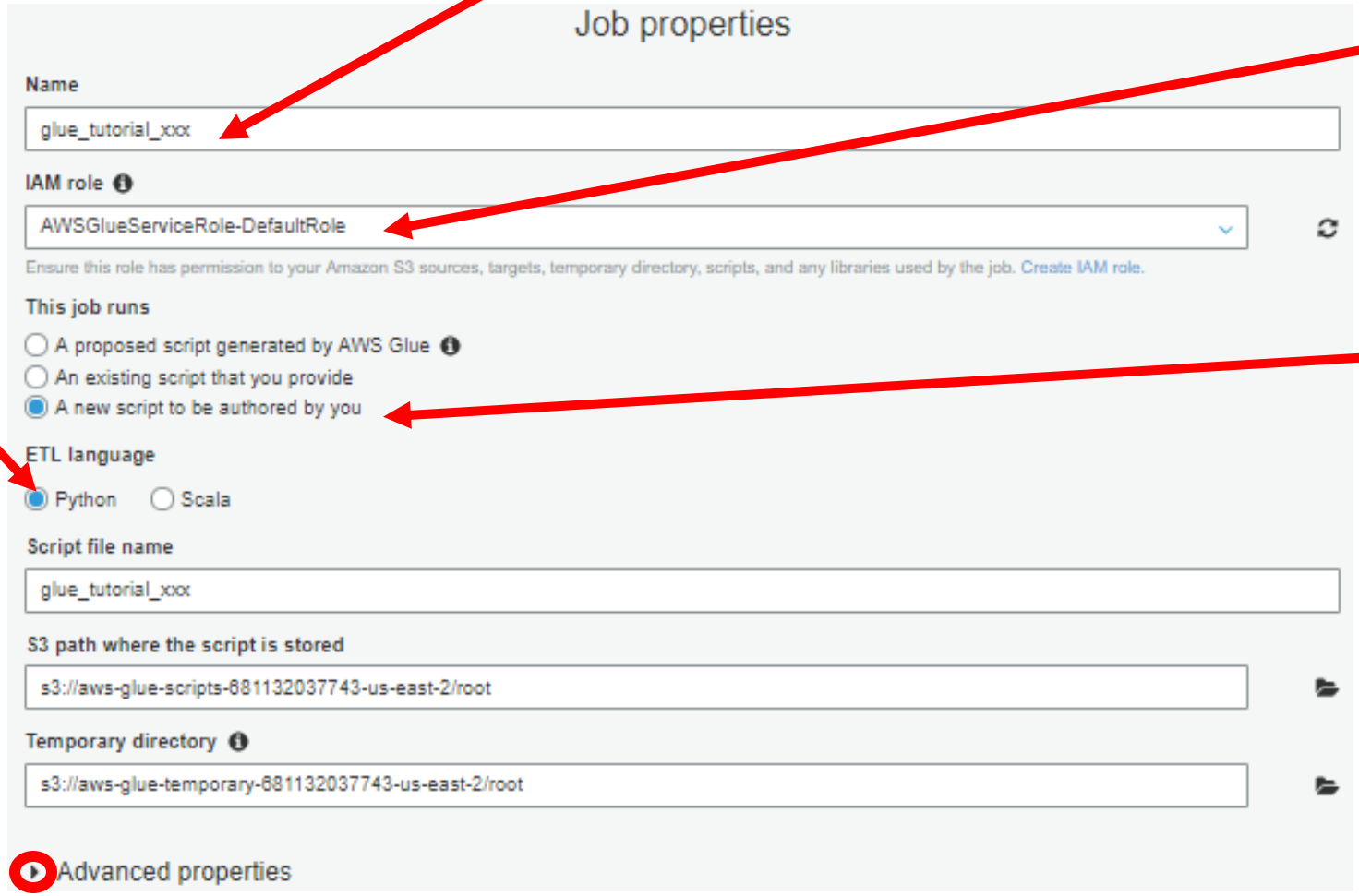
Create a Glue job

Give your job a name: glue-tutorial-XXX

Give your job a role to perform the actions necessary to run

The language used to write the script

Create a new blank script



Glue

Create a Glue job

DPU = Data Processing Unit. Glue jobs are charged per DPU hour. Change to 2

Job automatically stops after set time

Glue

Create a Glue job

Job parameters

Key	Value
--REDSHIFT_DB_NAME	glue_tutorial_database_xxx
--SCHEMA_NAME	sales_redshift_schema_xxx
--REDSHIFT_TABLE_NAME	products_redshift_table_xxx
--GLUE_DB_NAME	glue_database_xxx
--GLUE_TABLE_NAME	products_xxx
--CONNECTION_NAME	glue_tutorial_xxx
Type key...	Type value...

Next

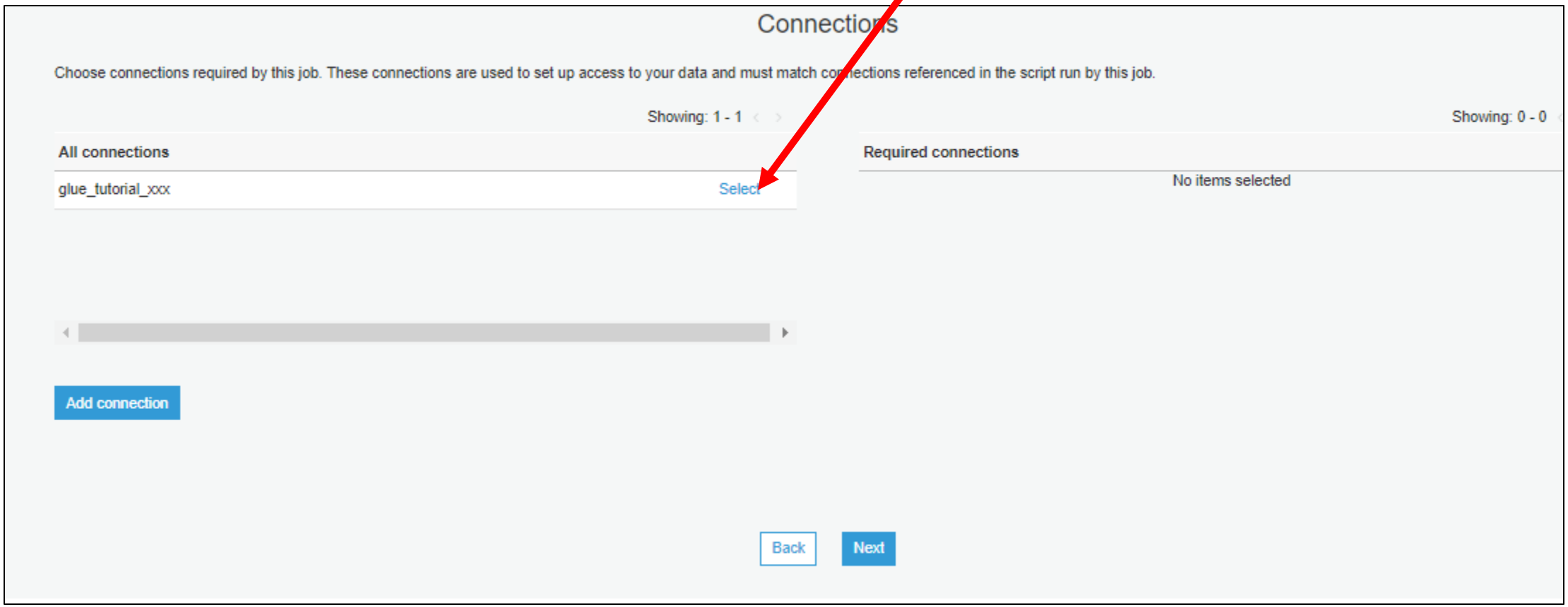
Parameters:
--REDSHIFT_DB_NAME
glue_tutorial_database_xxx
--REDSHIFT_TABLE_NAME
products_redshift_table_xxx
--SCHEMA_NAME
sales_redshift_schema_xxx
--GLUE_DB_NAME
glue_database_xxx
--GLUE_TABLE_NAME
products_xxx
--CONNECTION_NAME
glue_tutorial_xxx

Parameterize values to be used in the script

Glue

Create a Glue job

Select the Redshift connection that you want to use: glue-tutorial-XXX



Glue

Create a Glue job

Job properties

Name	glue_tutorial_xxx
IAM role	AWSGlueServiceRole-DefaultRole
ETL language	python
Connections	glue_tutorial_xxx
Path	s3://aws-glue-scripts-681132037743-us-east-2/root/glue_tutorial_xxx
Temporary directory	s3://aws-glue-temporary-681132037743-us-east-2/root

- Advanced properties
- Script libraries and job parameters (optional)

Back | Save job and edit script

Glue

Writing the Script

Job: glue_tutorial_xxx | Action | Save | Run job | Generate diagram | Insert template at cursor

```python
import sys
from awsglue.transforms import *
from awsglue.utils import getResolvedOptions
from pyspark.context import SparkContext
from awsglue.context import GlueContext
from awsglue.dynamicframe import DynamicFrame
from awsglue.job import Job

args = getResolvedOptions(sys.argv, ['TempDir'])

sc = SparkContext()
glueContext = GlueContext(sc)
spark = glueContext.spark_session
job = Job(glueContext)
job.init(args['JOB_NAME'], args)
```

PySpark is a service that allows the developer to perform data analysis on the data that is being used.

This is setting up the Spark and Glue environment to be able to interact with the data

Glue

Writing the Script

```
import sys
from awsglue.transforms import *
from awsglue.utils import getResolvedOptions
from pyspark.context import SparkContext
from awsglue.context import GlueContext
from awsglue.dynamicframe import DynamicFrame
from awsglue.job import Job
from pyspark.sql.functions import *
from pyspark.sql.types import *
from datetime import datetime

args = getResolvedOptions(sys.argv, ['TempDir', 'JOB_NAME', 'REDSHIFT_DB_NAME',
‘REDSHIFT_TABLE_NAME’, ‘GLUE_DB_NAME’, ‘GLUE_TABLE_NAME’, 'SCHEMA_NAME’,
'CONNECTION_NAME’])
sc = SparkContext()
glueContext = GlueContext(sc)
spark = glueContext.spark_session
job = Job(glueContext)
job.init(args['JOB_NAME'], args)
```

Include SQL functions, types, and datetime to use later

Add the parameters that were passed into the Glue job

Glue

Writing the Script

```
...
job.init(args['JOB_NAME'], args)

datasource =
glueContext.create_dynamic_frame.from_catalog(
    database = args['GLUE_DB_NAME'],
    table_name = args['GLUE_TABLE_NAME']
)
```

The data will be written to the datasource as a DynamicFrame

These are the database and the table that we created in Glue

Glue

Writing the Script

```
...
# Convert to PySpark Data Frame
sourcedata = datasource.toDF()

split_col = split(sourcedata["quarter"], " ")
sourcedata = sourcedata.withColumn("quarter new", split_col.getItem(0))
sourcedata = sourcedata.withColumn("profit", col("revenue")*col("gross margin"))
sourcedata = sourcedata.withColumn("current date", current_date())

# Convert back to Glue Dynamic Frame
datasource = DynamicFrame.fromDF(sourcedata, glueContext, "datasource")
```

sourcedata needs to be set to a Data Frame

This is where the transformations happen

Convert back to a Dynamic Frame

Glue

Writing the Script

```
...
applymapping = ApplyMapping.apply(
    frame = datasource,
    mappings = [
        ("retailer country", "string", "retailer_country", "varchar(20)"),
        ("order method type", "string", "order_method_type", "varchar(15)"),
        ("retailer type", "string", "retailer_type", "varchar(30)"),
        ("product line", "string", "product_line", "varchar(30)"),
        ("product type", "string", "product_type", "varchar(30)"),
        ("product", "string", "product", "varchar(50)"),
        ("year", "bigint", "year", "varchar(4)"),
        ("quarter new", "string", "quarter", "varchar(2)"),
        ("revenue", "double", "revenue", "numeric"),
        ("quantity", "bigint", "quantity", "integer"),
        ("gross margin", "double", "gross_margin", "decimal(15,10)"),
        ("profit", "double", "profit", "numeric"),
        ("current date", "date", "current_date", "date")
    ]
```

This is how the data in the DynamicFrame will be mapped to the columns in Redshift

Glue

Writing the Script

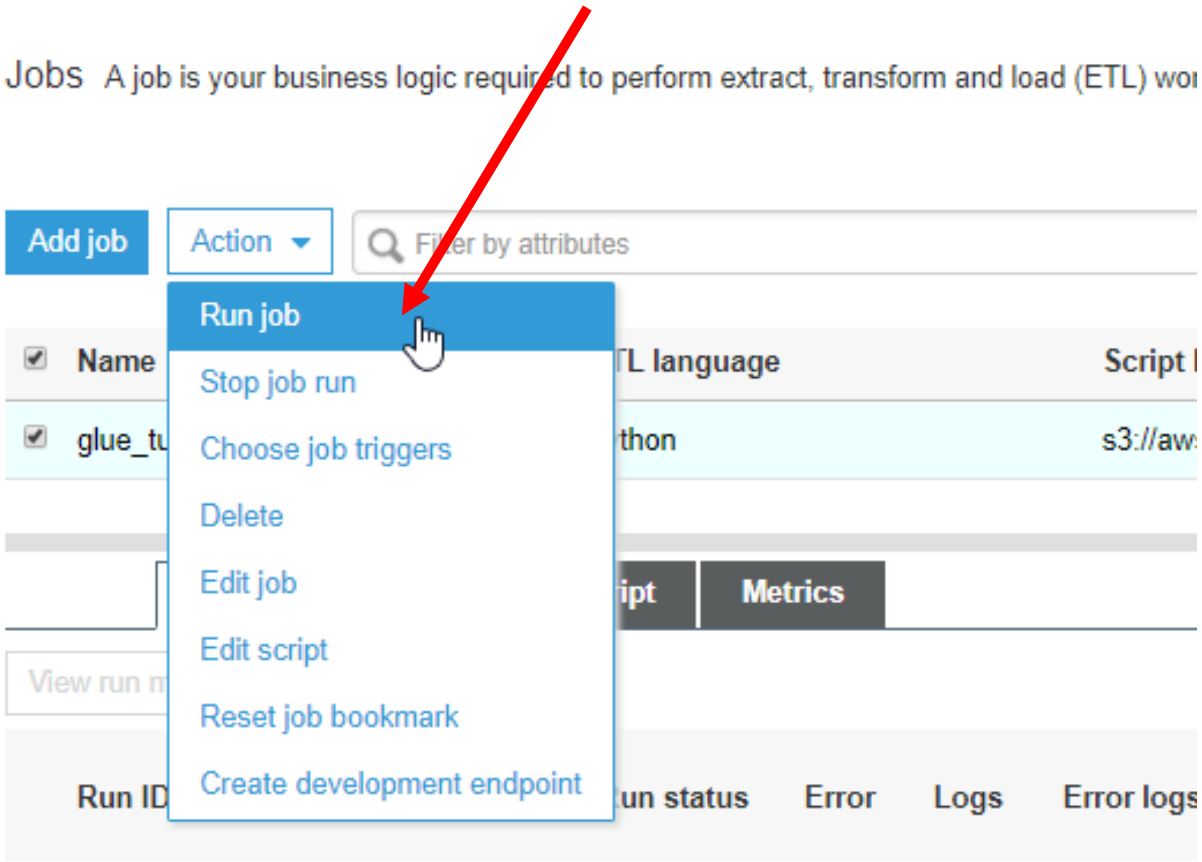
```
...
# datasink (loading) using spark
datasink = glueContext.write_dynamic_frame.from_jdbc_conf(
    frame = applymapping,
    catalog_connection = args['CONNECTION_NAME'],
    connection_options = {
        "dbtable": "{}.{}".format(args['SCHEMA_NAME'], args['REDSHIFT_TABLE_NAME']),
        "database": args['REDSHIFT_DB_NAME']
    },
    redshift_tmp_dir = args["TempDir"]
)
```

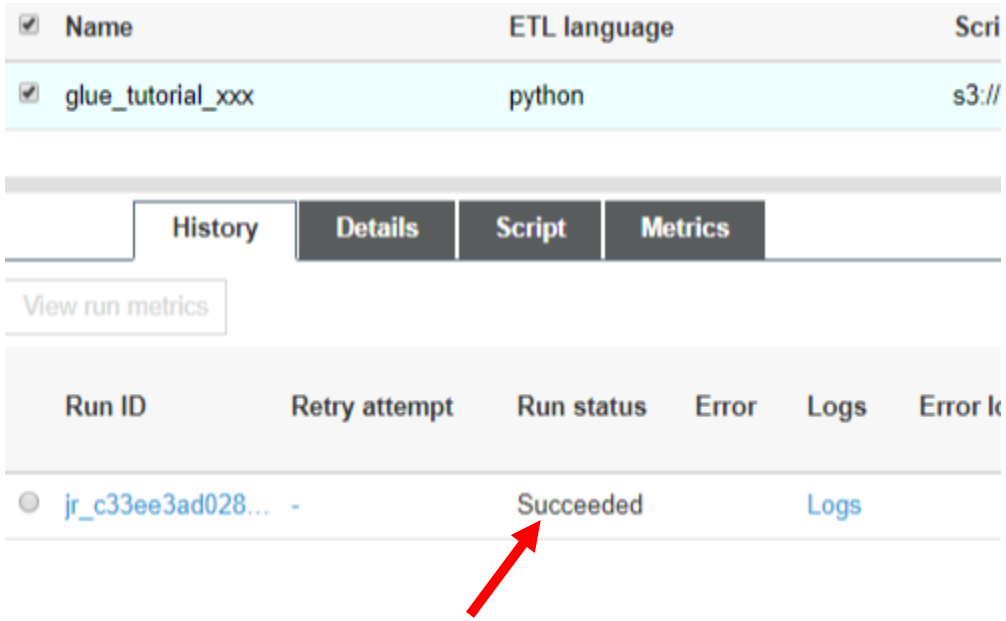
The datasink will connect to Redshift using the parameters given and load the data to Redshift

Glue

Run the Glue job

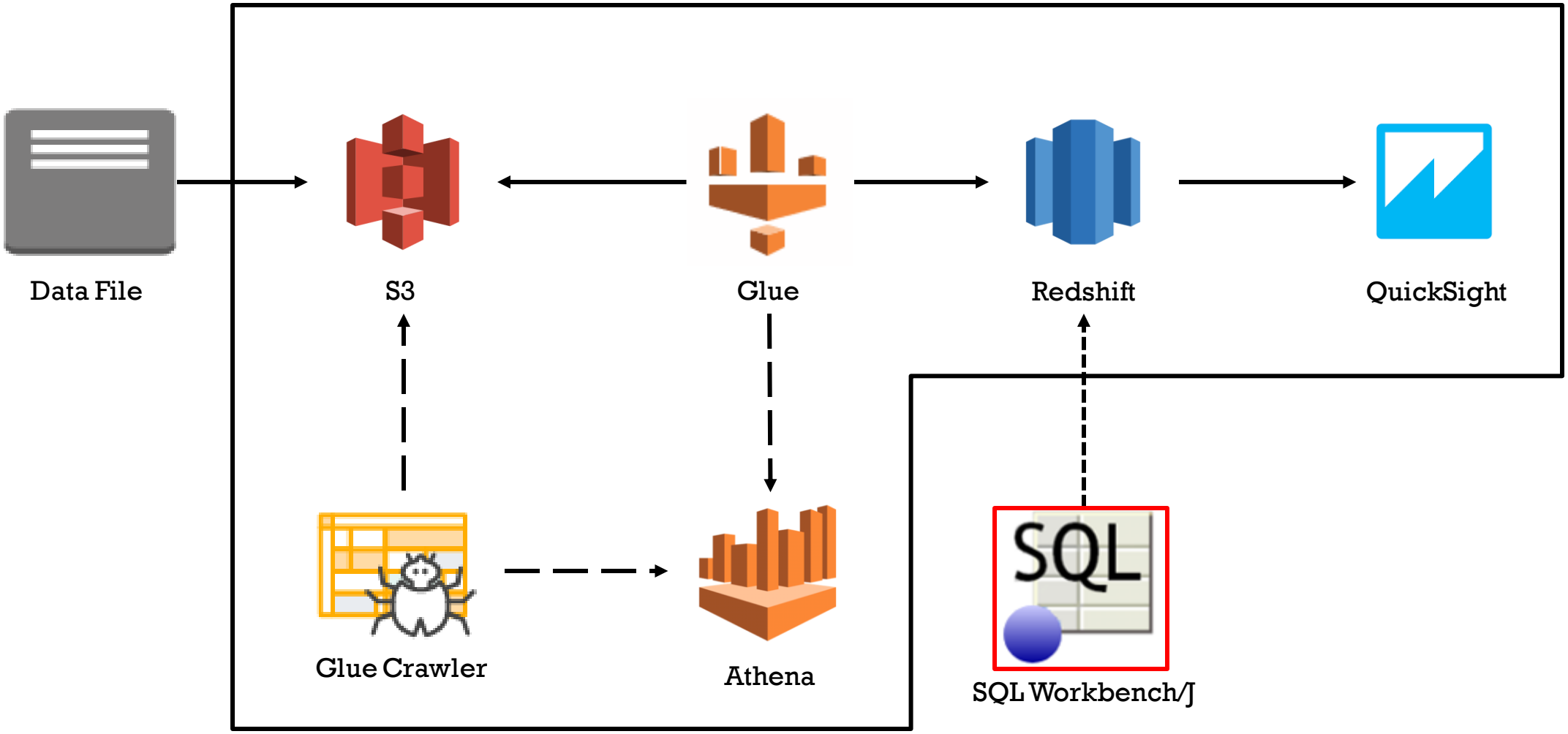
Run your Glue job





When the job succeeds, check the Redshift table

SQL Workbench



Redshift

Verify data in the table

```
1 SELECT *
2 FROM sales_redshift_schema_xxx.products_redshift_table_xxx LIMIT 100;
3
4
```

Result 1 | Messages

retailer_country	order_method_type	retailer_type	product_line	product_type	product	year	revenue	quantity	gross_margin	profit	timestamp	quarter	current_date
United States	Fax	Outdoors Shop	Camping Equipment	Cooking Gear	TrailChef Deluxe Cook Set	2012	59628.66	489	0.35	20723.82		Q1	2018-08-29
United States	Fax	Outdoors Shop	Camping Equipment	Tents	Star Dome	2012	89940.48	147	0.35	31728.48		Q1	2018-08-29
United States	Fax	Outdoors Shop	Camping Equipment	Sleeping Bags	Hibernator Lite	2012	119822.20	1415	0.29	34922.20		Q1	2018-08-29
United States	Fax	Outdoors Shop	Camping Equipment	Sleeping Bags	Hibernator Camp Cot	2012	41837.46	426	0.34	14040.96		Q1	2018-08-29
United States	Fax	Outdoors Shop	Camping Equipment	Lanterns	Firefly Extreme	2012	9393.30	189	0.43	4078.62		Q1	2018-08-29
United States	Fax	Outdoors Shop	Camping Equipment	Lanterns	EverGlow Butane	2012	6940.03	109	0.36	2511.36		Q1	2018-08-29
United States	Fax	Outdoors Shop	Mountaineering Equipment	Rope	Husky Rope 60	2012	14109.40	79	0.29	4115.11		Q1	2018-08-29
United States	Fax	Outdoors Shop	Mountaineering Equipment	Rope	Husky Rope 200	2012	77288.64	143	0.31	24328.59		Q1	2018-08-29
United States	Fax	Outdoors Shop	Mountaineering Equipment	Safety	Husky Harness	2012	34154.90	559	0.28	9687.47		Q1	2018-08-29
United States	Fax	Outdoors Shop	Mountaineering Equipment	Safety	Granite Signal Mirror	2012	4074.84	126	0.51	2095.38		Q1	2018-08-29
United States	Fax	Outdoors Shop	Mountaineering Equipment	Climbing Accessories	Granite Belay	2012	19476.80	296	0.48	9273.68		Q1	2018-08-29
United States	Fax	Outdoors Shop	Mountaineering Equipment	Climbing Accessories	Firefly Climbing Lamp	2012	17998.56	464	0.43	7697.76		Q1	2018-08-29
United States	Fax	Outdoors Shop	Mountaineering Equipment	Climbing Accessories	Firefly Rechargeable Battery	2012	11673.60	1520	0.59	6885.60		Q1	2018-08-29
United States	Fax	Outdoors Shop	Mountaineering Equipment	Tools	Granite Ice	2012	25041.60	333	0.48	12064.59		Q1	2018-08-29
United States	Fax	Outdoors Shop	Mountaineering Equipment	Tools	Granite Shovel	2012	9543.16	164	0.34	3216.04		Q1	2018-08-29
United States	Fax	Outdoors Shop	Mountaineering Equipment	Tools	Granite Axe	2012	32870.40	856	0.49	16161.28		Q1	2018-08-29
United States	Fax	Outdoors Shop	Personal Accessories	Watches	Mountain Man Extreme	2012	6499.80	23	0.59	3827.43		Q1	2018-08-29
United States	Fax	Outdoors Shop	Personal Accessories	Eyewear	Polar Ice	2012	3825.80	37	0.52	1987.27		Q1	2018-08-29
United States	Fax	Outdoors Shop	Personal Accessories	Knives	Bear Survival Edge	2012	8414.75	97	0.48	4049.75		Q1	2018-08-29
United States	Fax	Outdoors Shop	Outdoor Protection	Insect Repellents	BugShield Extreme	2012	25010.58	3801	0.63	15812.16		Q1	2018-08-29
United States	Fax	Outdoors Shop	Outdoor Protection	First Aid	Compact Relief Kit	2012	4057.20	180	0.60	2437.20		Q1	2018-08-29

Enhancements

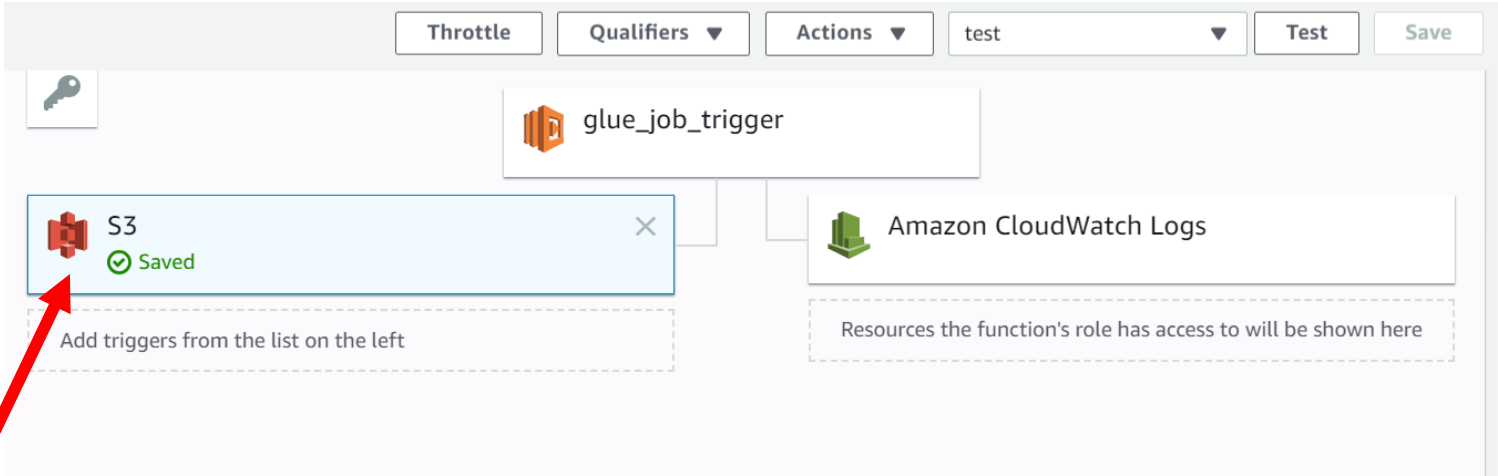
└─**Improve the versatility of your Glue job**

- Create a Glue Trigger
  - Automatically run the Glue job
  - Run multiple different Glue jobs
- Control how resources can interact with other services
- Easily create, modify, and delete as well as move Glue jobs with a template
- Create reports for business analytics with the data that was loaded with the Glue job.

Glue Trigger

Automatically run Glue job using Lambda – a serverless function

- Instead of running the Glue job manually, have it run automatically when a file is added to S3
- Use a Lambda
- You can set a Lambda to run when a file lands in an S3 bucket
- Then make the Lambda run the Glue job



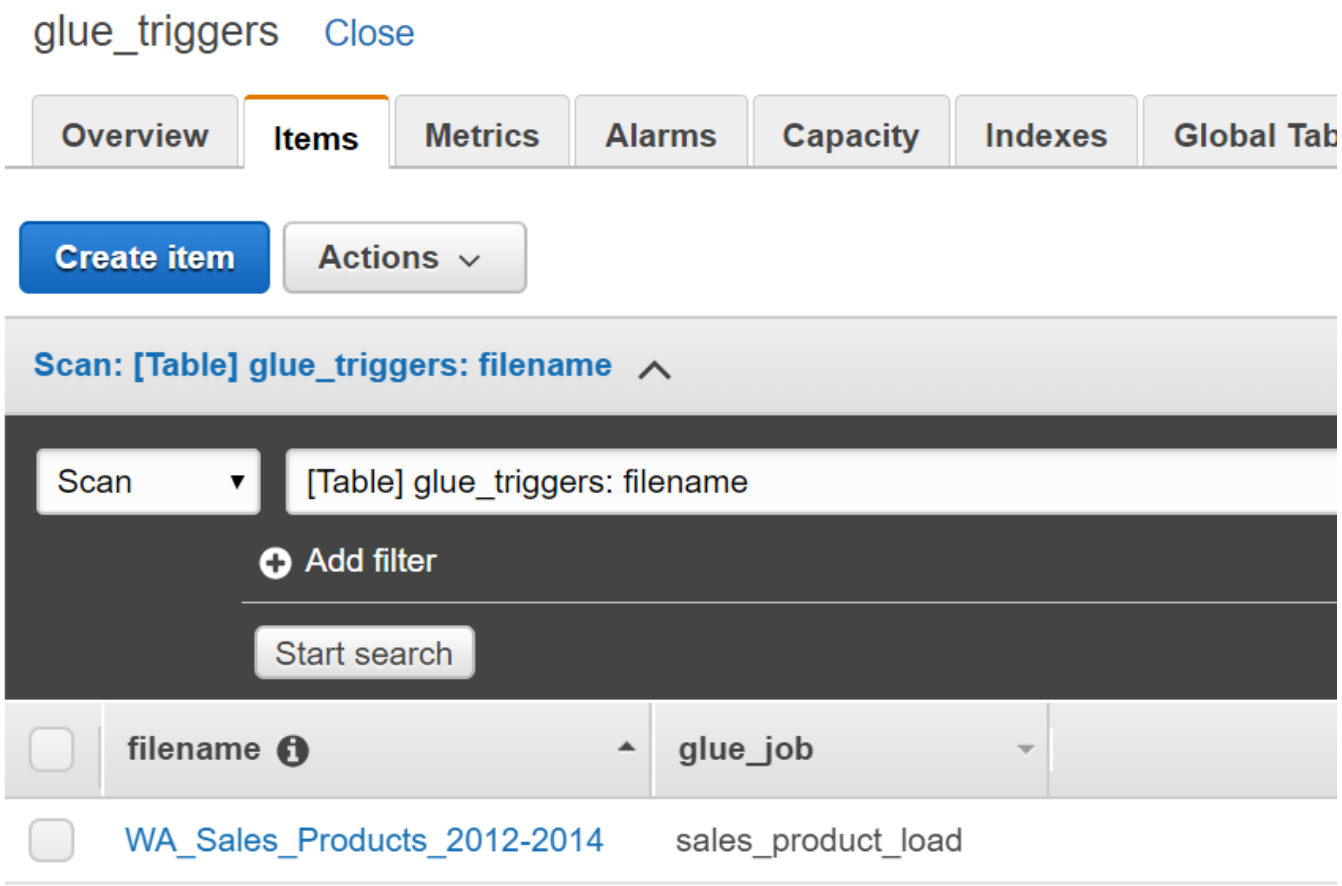
lambda_function

```
import boto3

def lambda_handler(event, context):

    glue = boto3.client('glue')
    glueResponse = glue.start_job_run(JobName = "sales_product_load")
```

Glue Trigger

└─ **Run multiple different Glue jobs with DynamoDB – a non-relational database**

- The Lambda currently can only run one Glue job
- It would be better if it could run different Glue jobs based on the file.
- We could store that information in a DynamoDB table



Glue Trigger

└─**Automatically run Glue job using Lambda**

- The Lambda can look up the filename in the DynamoDB table to find which Glue job to run

Lambda receives an event from S3, which includes the 'key'

lambda_function

```python
import boto3

def lambda_handler(event, context):

    sourceKeyName = event['Records'][0]['s3']['object']['key']
    filename = sourceKeyName.rsplit('/',1)[1].split('.',1)[0]

    dynamodb = boto3.resource('dynamodb')
    table = dynamodb.Table('glue_triggers')

    dynamoDBResponse = table.get_item(Key = { "filename" : filename })
    glue_job = dynamoDBResponse['Item']['glue_job']

    glue = boto3.client('glue')
    glueResponse = glue.start_job_run(JobName = glue_job)
```

This returns the Glue job associated with that file

We get the filename from the key, then search the DynamoDB table with it

Glue Trigger

└─**IAM Roles determine how a resource can interact with other services**

Log output

The area below shows the logging calls in your code. These correspond to a single row within the CloudWatch log group corresponding to this Lambda function. Click here to view the CloudWatch log group.

```
START RequestId: 2df6f8a8-95cb-11e8-aedb-510d0136df8b Version: $LATEST
An error occurred (AccessDeniedException) when calling the GetItem operation: User: arn:aws:sts::952552944372:assumed-
role/lambda_basic_execution/glue_job_trigger is not authorized to perform: dynamodb:GetItem on resource: arn:aws:dynamodb:us-east-
1:952552944372:table/glue_triggers: ClientError
Traceback (most recent call last):
```

- If you made the lambda from the previous slides, you would get an AccessDeniedException
- We need to add permission to the Lambda's IAM Role to access DynamoDB and Glue

Glue Trigger

└─**IAM Roles determine how a resource can interact with other services**

Permissions | Trust relationships | Access Advisor | Revoke sessions

Permissions policies (1 policy applied)

Attach policies

Add inline policy

Policy name	Policy type
AWSLambdaBasicExecutionRole-5812fc8e-f707-4957-96eb-47ec0eacd883	Managed policy

Add permissions to lambda_basic_execution

Attach Permissions

Create policy

Filter policies

dynamo

Policy name	Type	Used as
AmazonDynamoDBRe…	AWS managed	None

AmazonDynamoDBReadOnlyAccess

Provides read only access to Amazon DynamoDB via the AWS Management Console.

Visual editor | JSON

Expand all | Collapse all

Glue (1 action)

Service Glue

Actions Specify the actions allowed in Glue

close

start

- StartCrawler
- StartCrawlerSchedule
- StartJobRun
- StartTrigger

CLOUDFORMATION

└─**Templates**

- Template used build the infrastructure for AWS resources
- Use Case:
  - Build Glue job through Cloud Formation vs Glue console
  - Advantages
    - Easy to modify
    - Easy to create multiple Glue jobs with similar patterns
    - Easy to delete multiple related resources at once
    - Easy to deploy to a different account

CLOUDFORMATION

└─ **Templates**

```
Resources:
  MyJob:
    Type: AWS::Glue::Job
    Properties:
      Command:
        Name: glueetl
      ScriptLocation: !Ref ScriptLocation
      AllocatedCapacity: 2
      DefaultArguments:
        "--REDSHIFT_DB_NAME": !Ref RedshiftDBName
          "--SCHEMA_NAME": !Ref SchemaName
          "--REDSHIFT_TABLE_NAME": !Ref RedshiftTableName
          "--GLUE_TABLE_NAME": !Ref GlueTableName
          "--CONNECTION_NAME": !Ref GlueConnectionName
          "--GLUE_DB_NAME": !Ref GlueDatabaseName
      ExecutionProperty:
          MaxConcurrentRuns: 2
      Connections: !Ref GlueConnectionName
      MaxRetries: 0
      Name: !Ref GlueJobName
```

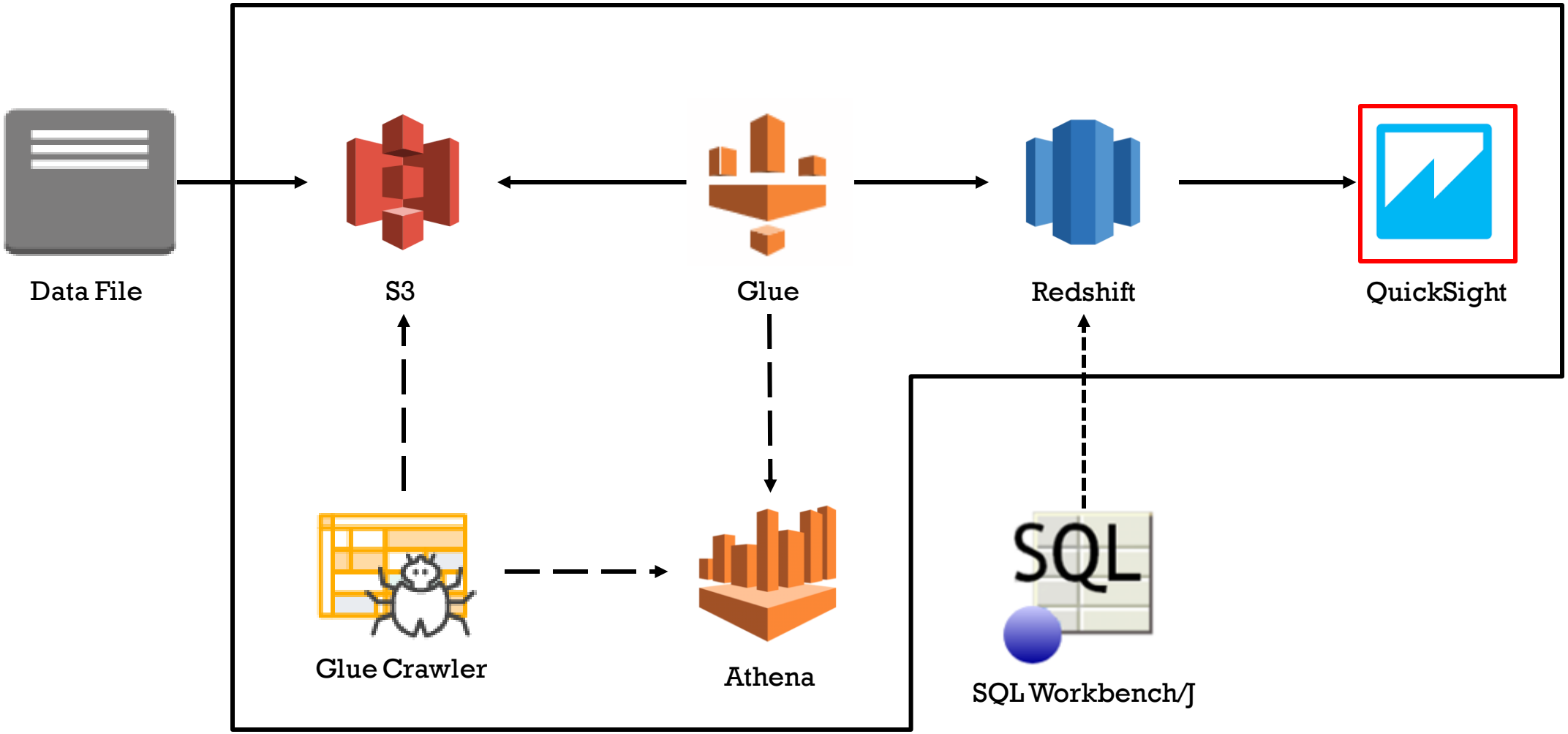
CLOUDFORMATION

└─**Templates**

```
AWSTemplateFormatVersion: "2010-09-09"

Parameters:
  GlueDatabaseName:
    Type: String
    Default: glue_database_XXX
  GlueConnectionName:
    Type: String
    Default: glue_tutorial_XXX
  RedshiftDBName:
    Type: String
    Default: glue_tutorial_database_XXX
  SchemaName:
    Type: String
    Default: sales_redshift_schema_XXX
  RedshiftTableName:
    Type: String
    Default: products_redshift_table_XXX
  GlueTableName:
    Type: String
    Default: products_glue_table_XXX
  GlueJobName:
    Type: String
    Default: glue_tutorial
  ScriptLocation:
    Type: String
    Default: "s3://glue-tutorial-XXX/products_XXX"
```

QUICKSIGHT

AWS Business Intelligence Tool



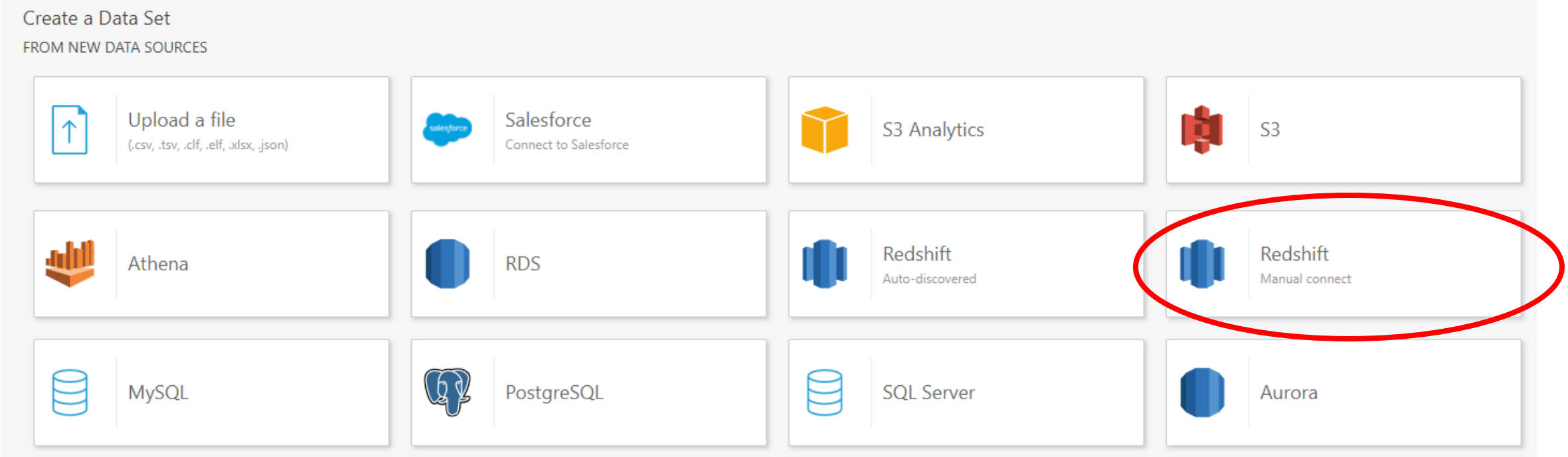
QUICKSIGHT

AWS Business Intelligence Tool

- Cloud based Business Intelligence reporting tool
- Build Reports from
  - Files in S3
  - Redshift
  - Athena

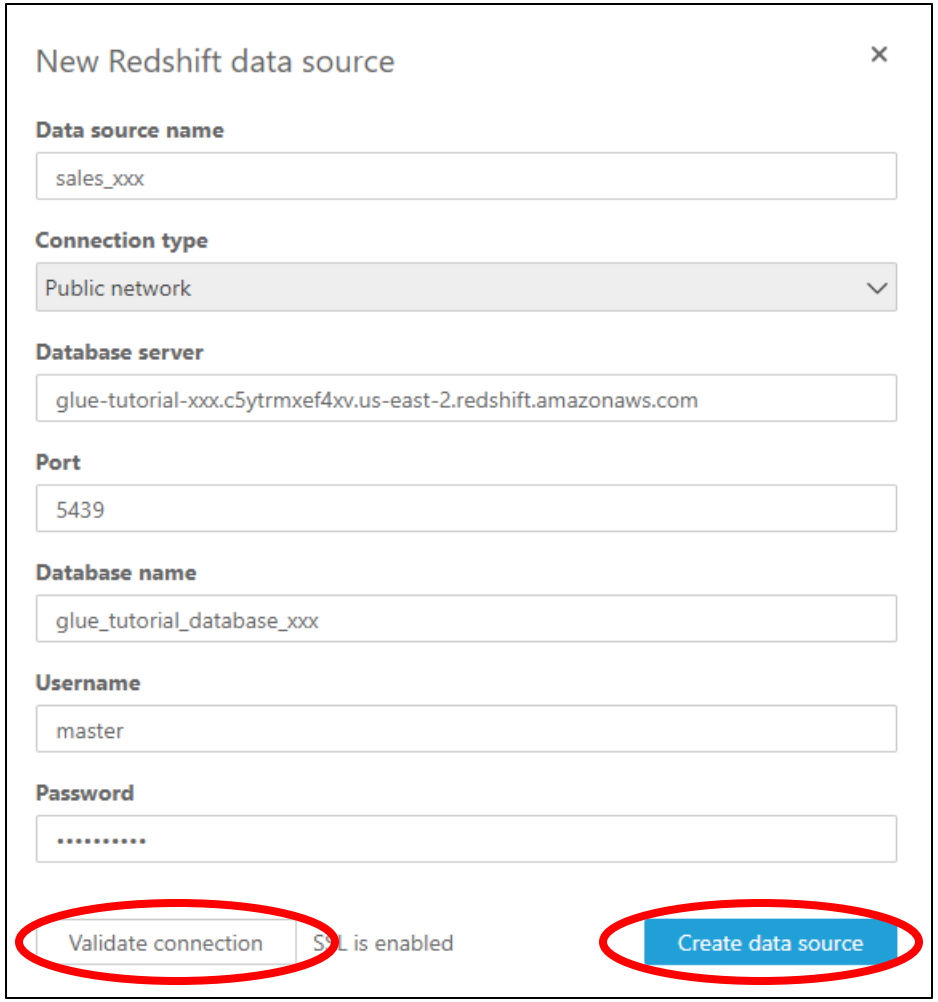
QUICKSIGHT

AWS Business Intelligence Tool



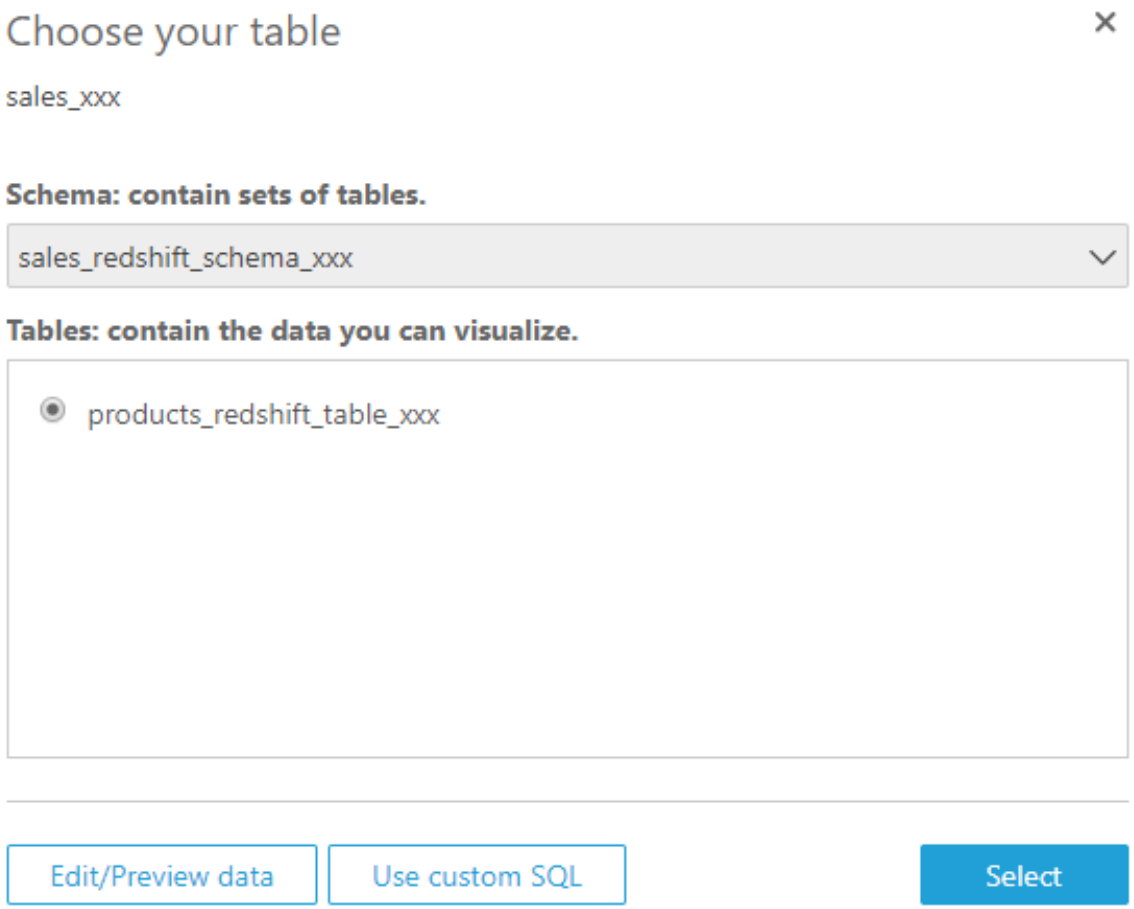
QUICKSIGHT

AWS Business Intelligence Tool



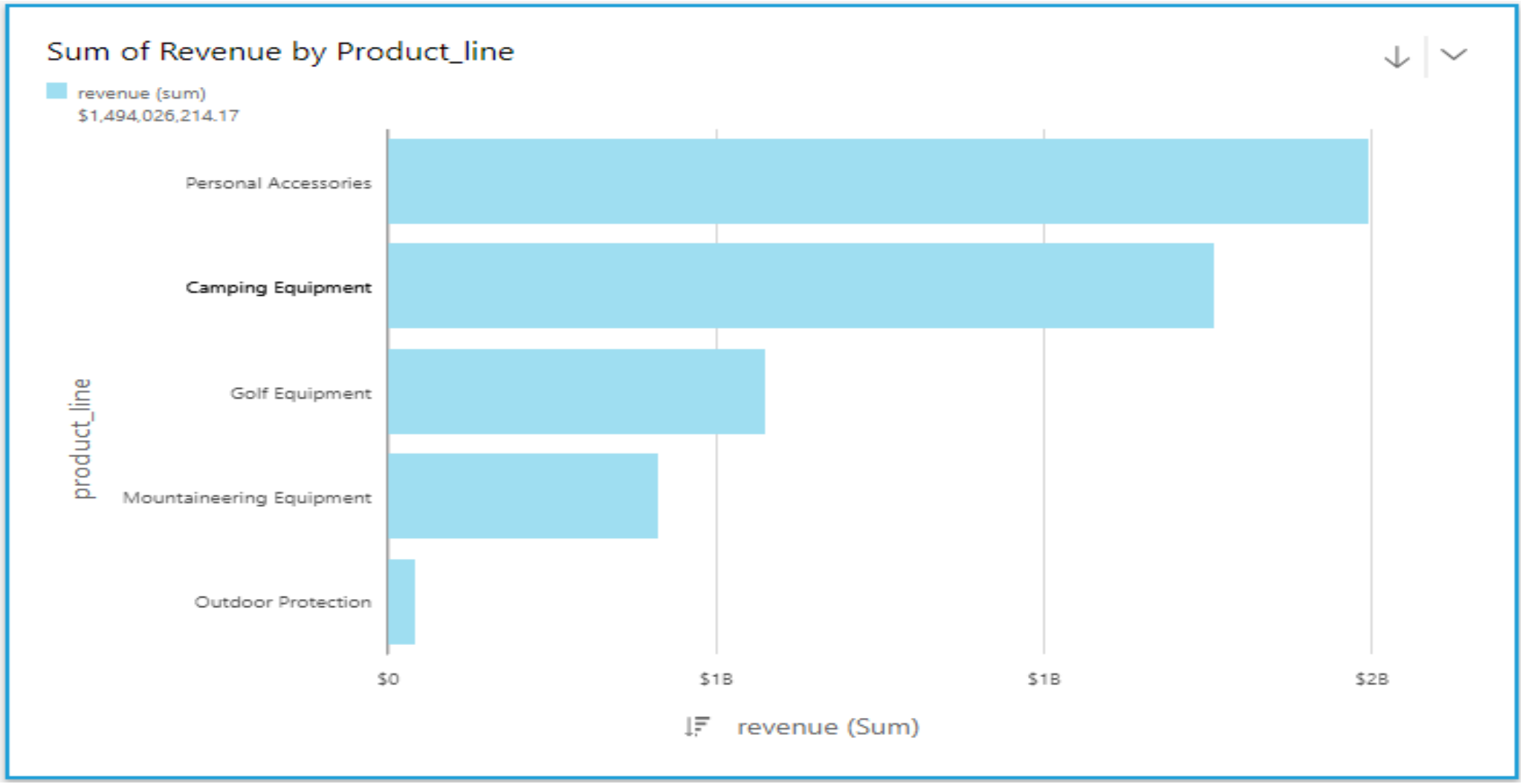
QUICKSIGHT

AWS Business Intelligence Tool



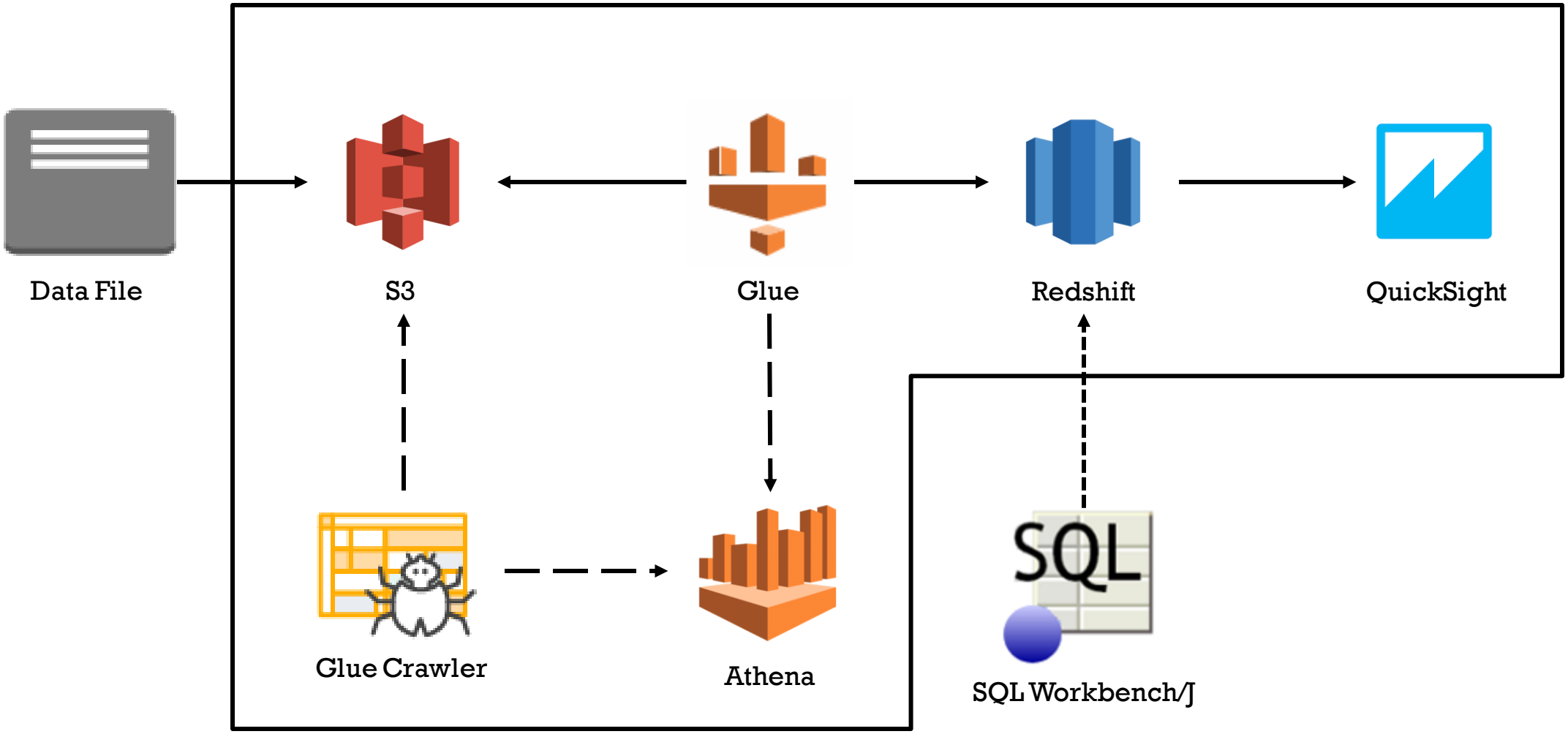
QUICKSIGHT

AWS Business Intelligence Tool



SUMMARY

AWS Data Workflow



CONCLUSION

Glue - AWS ETL Tool

Simple –

Use AWS for your entire ETL workflow
Less Setup

Flexible –

Good for developers as well as non-developers
Customizable

Cost Effective –

Cheaper than other ETL tools
Pay only when you use Glue

RESOURCES

AWS Glue Documentation
https://aws.amazon.com/glue/

Pricing
Informatica
https://aws.amazon.com/marketplace/pp/B0752DY9DV?qid=1534179668153&sr=0-1&ref_=srh_res_product_title
Glue
https://aws.amazon.com/glue/pricing/
Matillion
https://aws.amazon.com/marketplace/pp/B010ED5YF8

AWS Services Documentation
https://aws.amazon.com/documentation/

Hadoop vs AWS
https://www.trustradius.com/compare-products/amazon-web-services-vs-hadoop
https://databricks.com/blog/2017/05/31/top-5-reasons-for-choosing-s3-over-hdfs.html
https://data-flair.training/blogs/13-limitations-of-hadoop/

Links

- AWS Glue Tutorial Presentation: https://github.com/jackdsilverman/aws-glue-tutorial/blob/master/glue-tutorial.pptx
- AWS Glue Workshop: https://github.com/jackdsilverman/aws-glue-tutorial

James Zhang jzhang@manifestcorp.com

Lydia White lwhite@manifestcorp.com

Thanks to Jack Silverman and Jerry Ralph