# Amazon Athena Advanced Features

Michael Lin

Senior Solutions Architect
Amazon Web Services

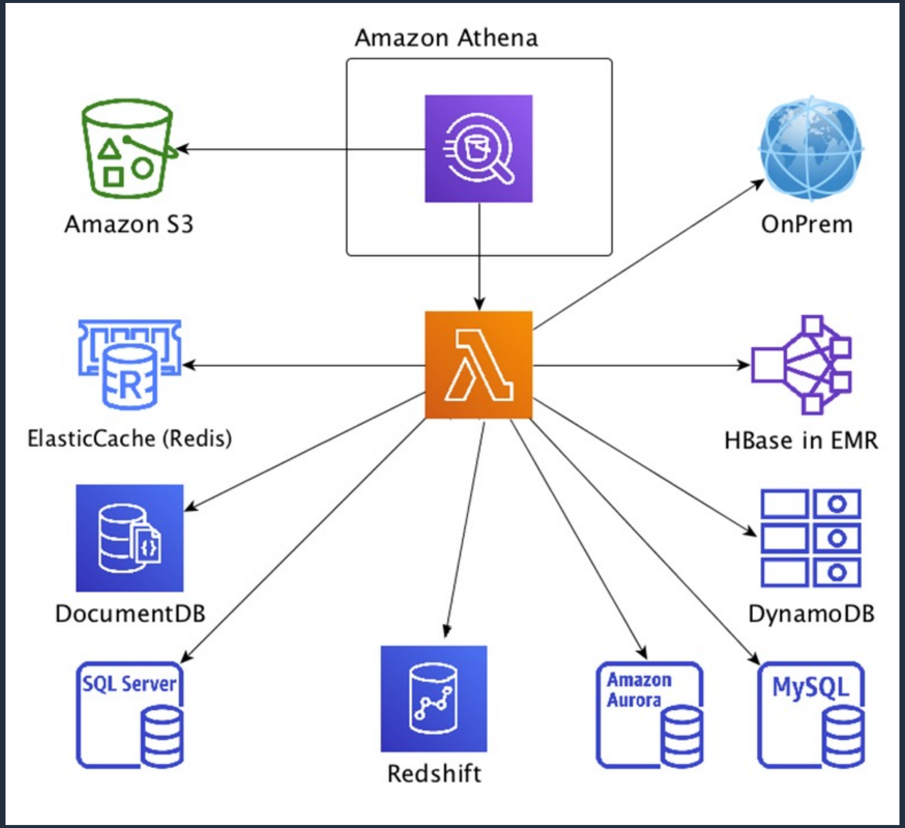
# Agenda

- Federated Queries
- User Defined Functions
- Machine Learning Capabilities

# Federated query in Athena

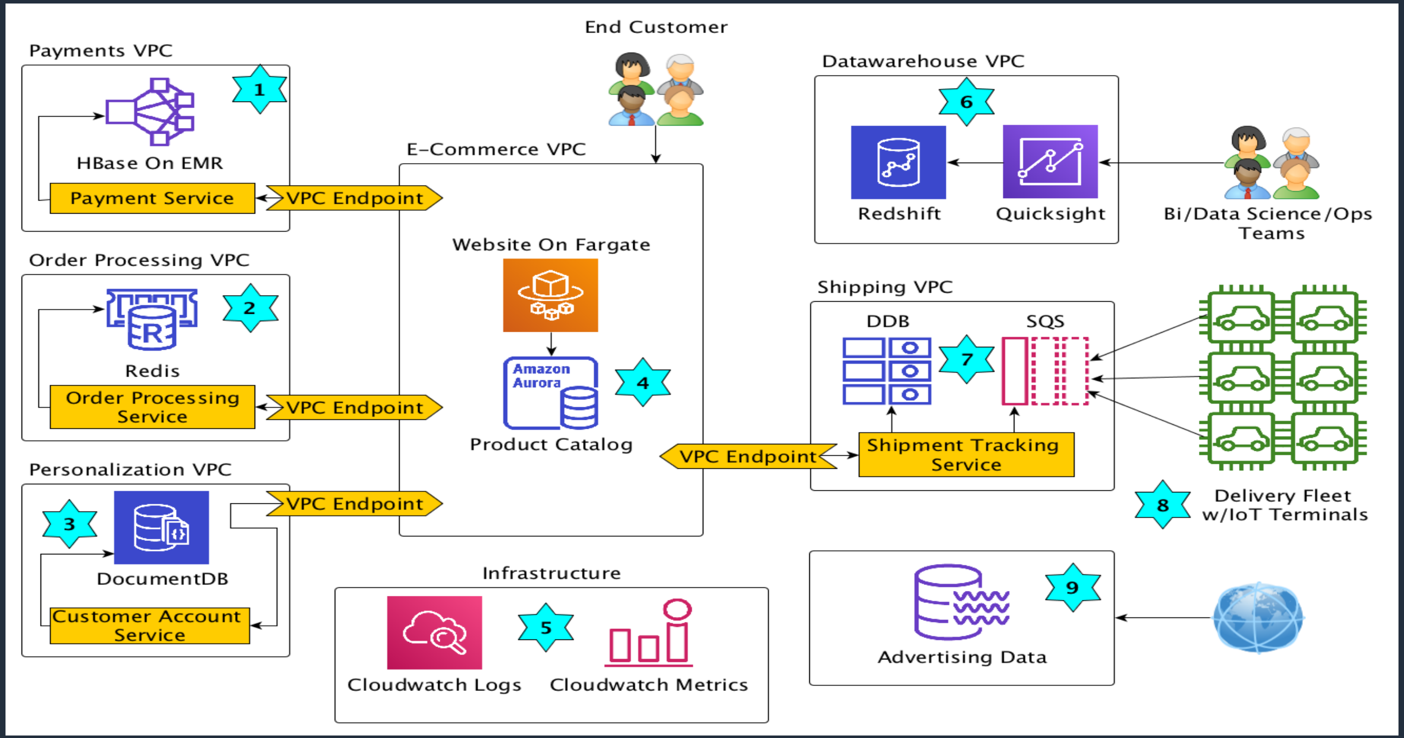
# What is federated query?

- Run query across relational, non-relational, object, or custom data sources
- Run query across On-Premises or cloud data sources
- Can be used for ad-hoc investigations, or complex pipelines, or applications



# Why do you need federated query

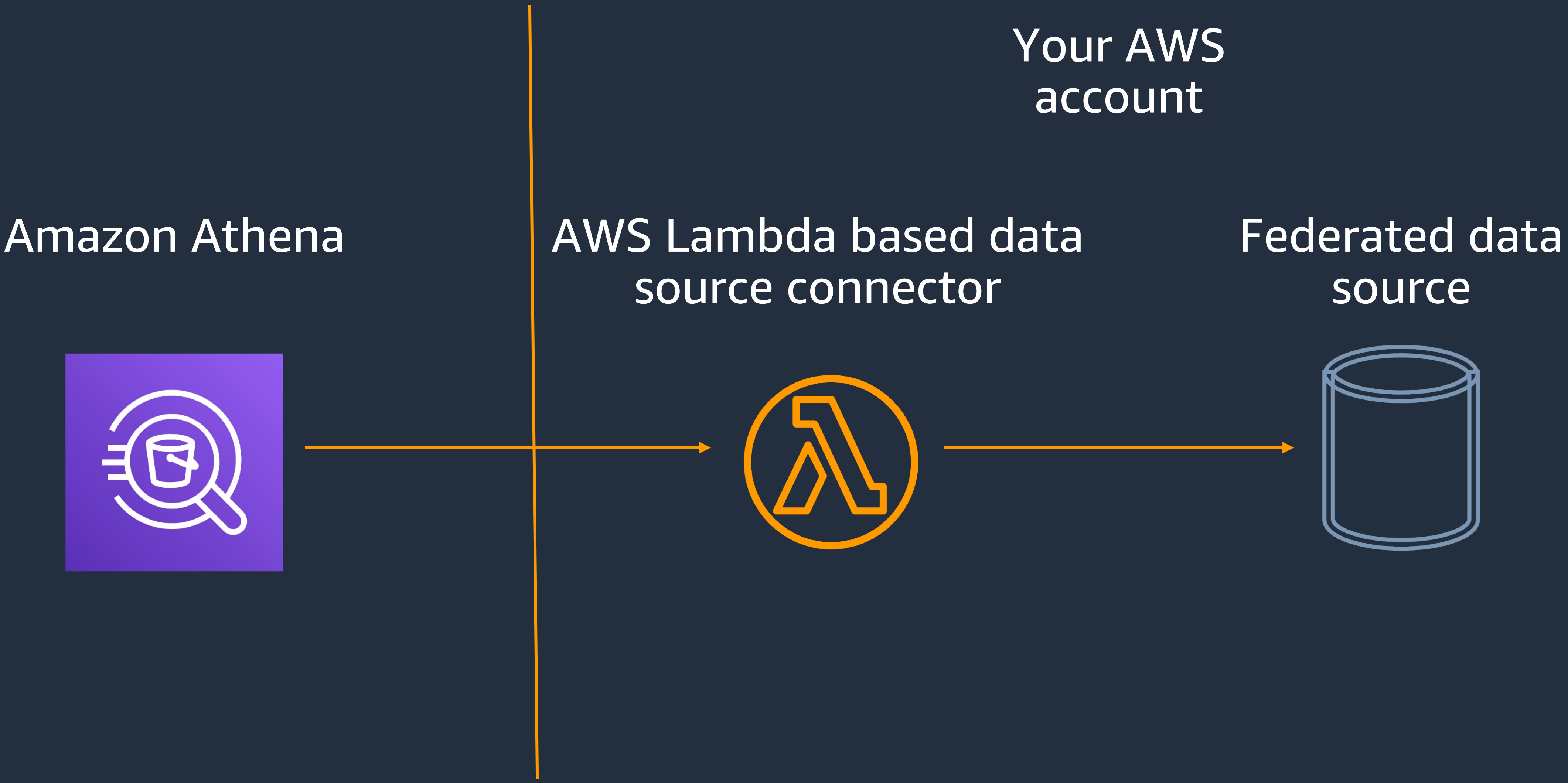
## Evolving architecture



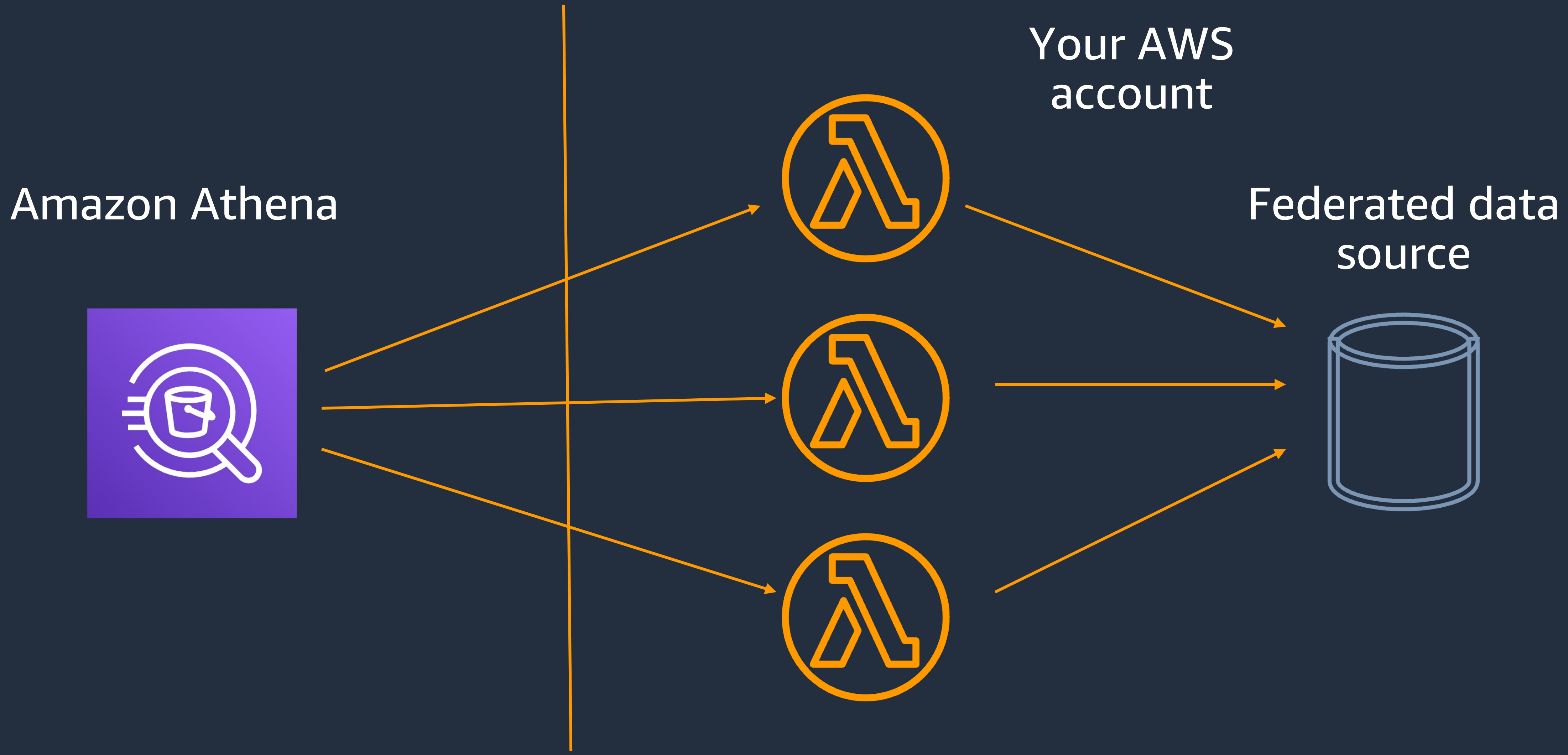
Engineering teams use fit for purpose databases

Aggregating data for analytics is a challenge

# Anatomy of a federated query



# Running a federated query



# Federated query is simple to use

1

Deploy data source connector

2

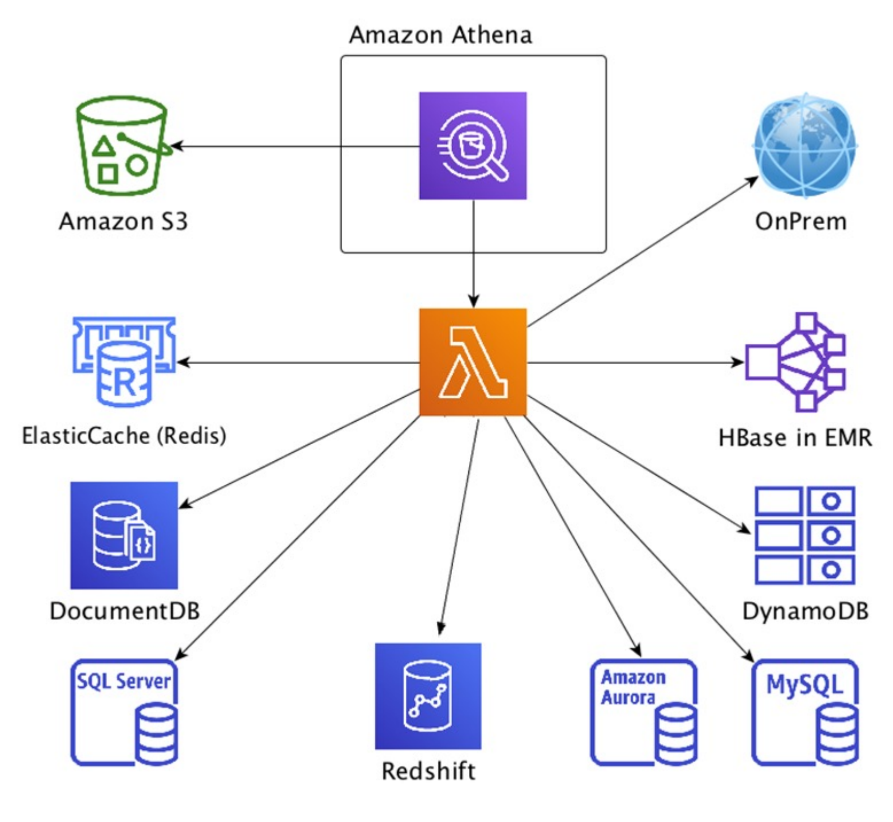
Register data source connector. Specify a catalog name

3

Write SQL Query <CatalogName>.Database.Table

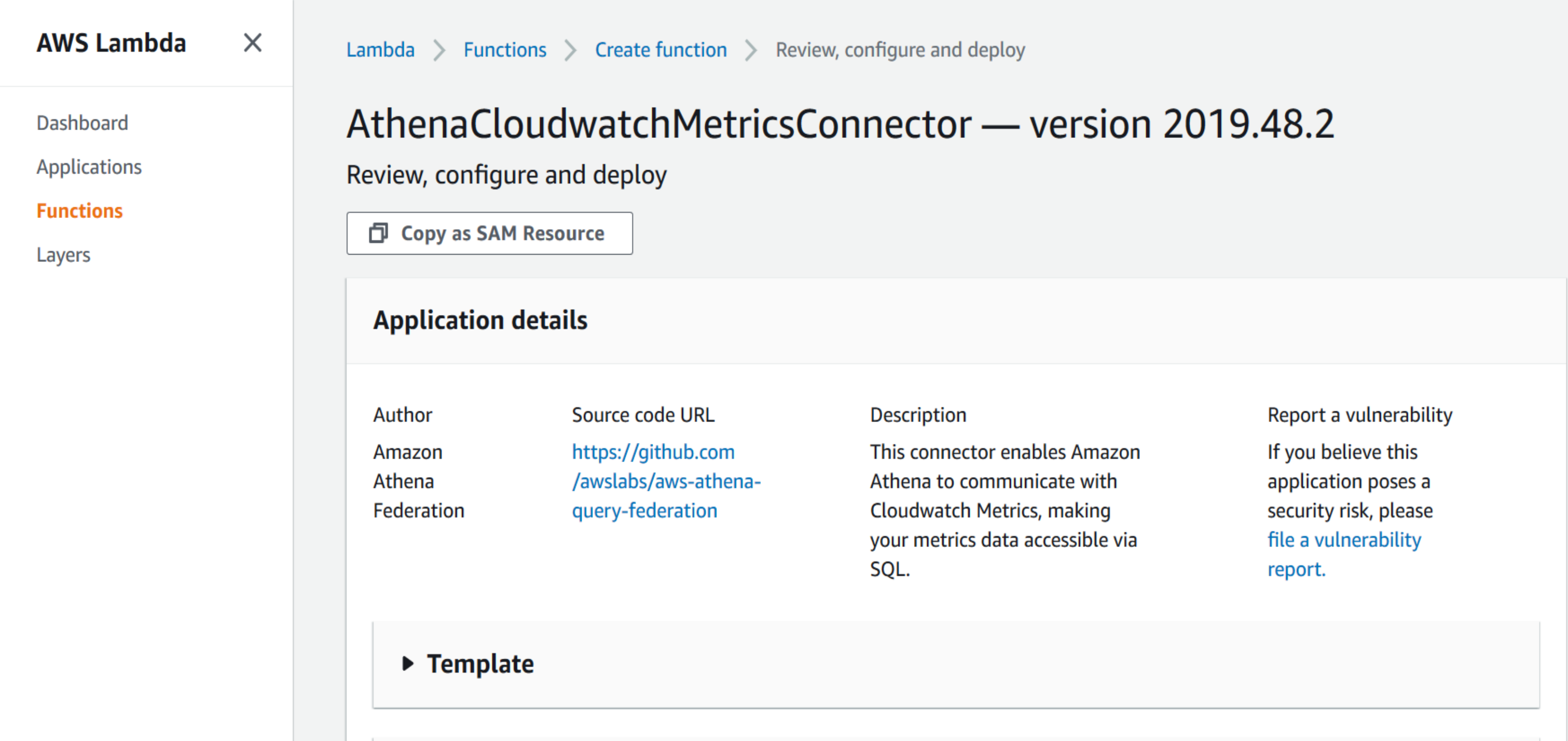
# How to deploy a data source connector

- Athena uses AWS Lambda based data source connectors
- Two ways to deploy connector
  - One-Click deploy using AWS Serverless Application Repository
  - Deploy connector code to Lambda



# One-click deploy using Serverless Application Repository

## Upload connector to AWS Serverless Application Repository



### AthenaCloudwatchMetricsConnector — version 2019.48.2

Review, configure and deploy

**Application details**

Author

Amazon Athena Federation

Source code URL

https://github.com/awslabs/aws-athena-query-federation

Description

This connector enables Amazon Athena to communicate with Cloudwatch Metrics, making your metrics data accessible via SQL.

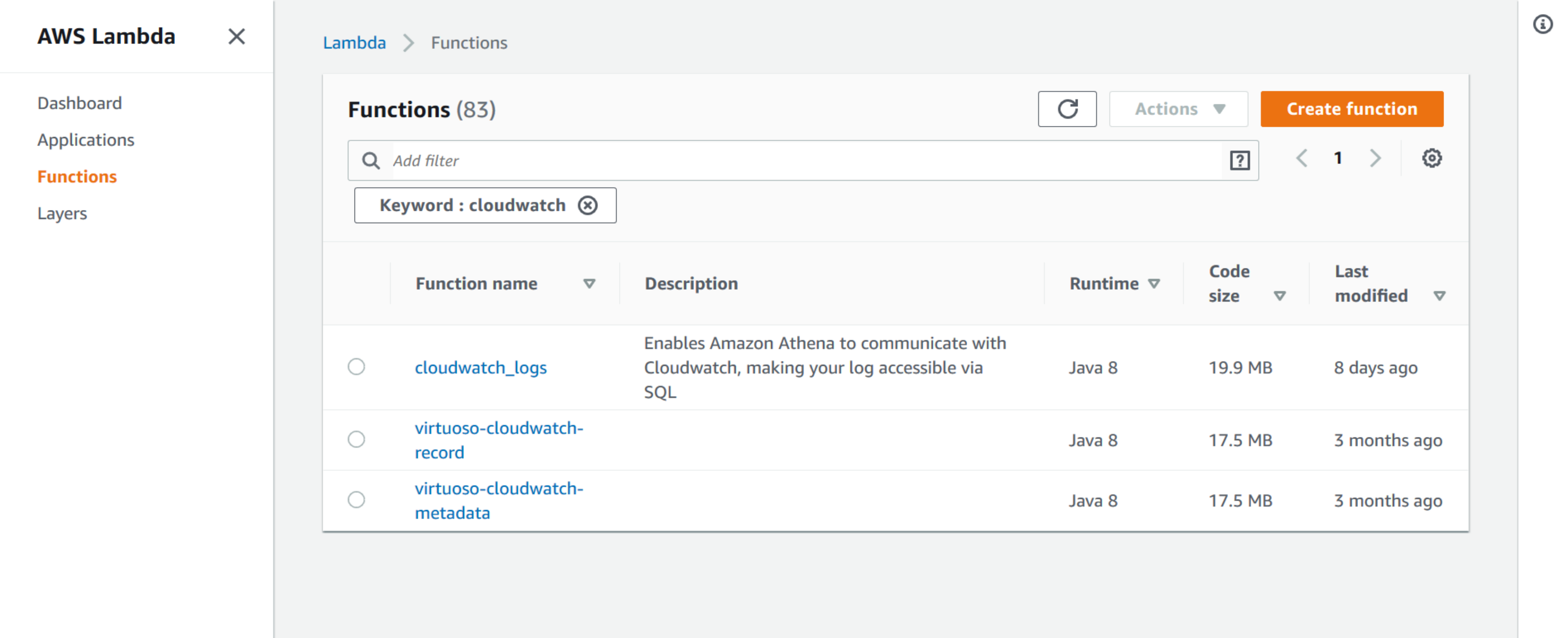
Report a vulnerability

If you believe this application poses a security risk, please file a vulnerability report.

Deploy | Register | Use

# Deploy connector to AWS Lambda

## Upload connector to AWS Lambda using Lambda API, UI



Functions (83)

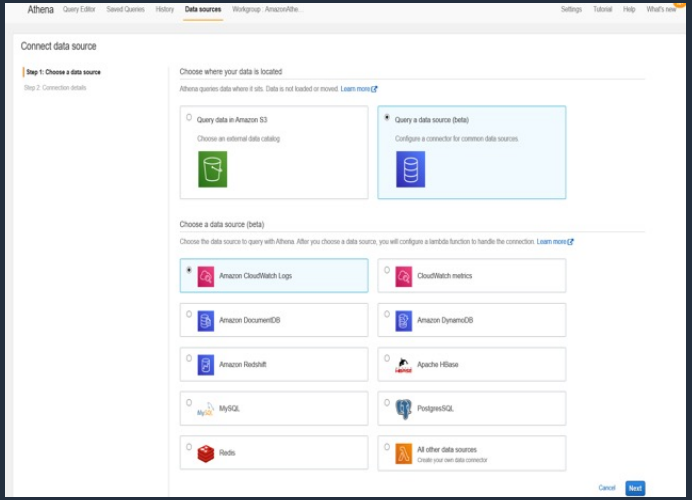
Keyword : cloudwatch

| | Function name | Description | Runtime | Code size | Last modified |
|---|---|---|---|---|---|
| ○ | cloudwatch_logs | Enables Amazon Athena to communicate with Cloudwatch, making your log accessible via SQL | Java 8 | 19.9 MB | 8 days ago |
| ○ | virtuoso-cloudwatch-record | | Java 8 | 17.5 MB | 3 months ago |
| ○ | virtuoso-cloudwatch-metadata | | Java 8 | 17.5 MB | 3 months ago |

Deploy | Register | Use

# Use Athena Console to register connector

## To use an existing data source connector



Connect data source

Step 1: Choose a data source

Step 2: Connection details

Choose where your data is located

Athena queries data where it sits. Data is not loaded or moved. Learn more

○ Query data in Amazon S3
Choose an external data catalog

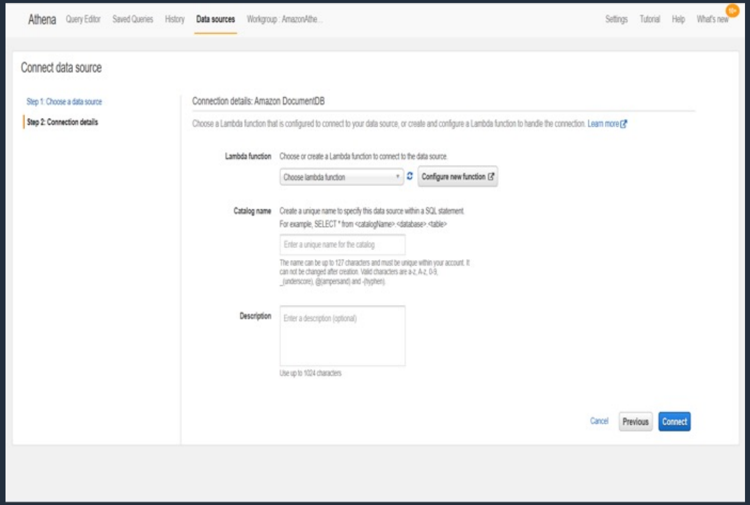
● Query a data source (beta)
Configure a connector for common data sources.

Choose a data source (beta)

Choose the data source to query with Athena. After you choose a data source, you will configure a lambda function to handle the connection. Learn more

● Amazon CloudWatch Logs
○ CloudWatch metrics
○ Amazon DocumentDB
○ Amazon DynamoDB
○ Amazon Redshift
○ Apache HBase
○ MySQL
○ PostgresSQL
○ Redis
○ All other data sources
Create your own data connector

Discover



Connect data source

Step 1: Choose a data source

Step 2: Connection details

Connection details: Amazon DocumentDB

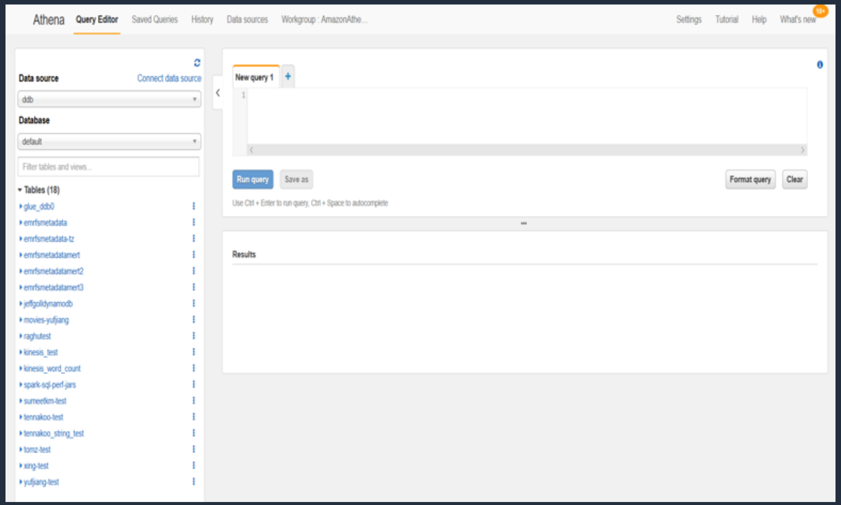
Choose a Lambda function that is configured to connect to your data source, or create and configure a Lambda function to handle the connection. Learn more

Lambda function: Choose or create a Lambda function to connect to the data source.
Choose lambda function

Catalog name: Create a unique name to specify this data source within a SQL statement.
For example, SELECT * from <catalogName>.<database>.<table>
Enter a unique name for the catalog
The name can be up to 127 characters and must be unique within your account. It can not be changed after creation. Valid characters are a-z, A-z, 0-9, _(underscore), @(ampersand) and -(hyphen).

Description: Enter a description (optional)
Use up to 1024 characters

Select



Data source: ddb

Database: default

Filter tables and views...

Tables (18)
- glue_ddb0
- emrfsmetadata
- emrfsmetadata-tz
- emrfsmetadatamert
- emrfsmetadatamert2
- emrfsmetadatamert3
- jeffgolldynamodb
- movies-yufjiang
- raghutest
- kinesis_test
- kinesis_word_count
- spark-sql-perf-jars
- sumeetkm-test
- tennakoo-test
- tennakoo_string_test
- tomz-test
- xing-test
- yufjiang-test

Use Ctrl + Enter to run query, Ctrl + Space to autocomplete

Query

# Registration-less federated query

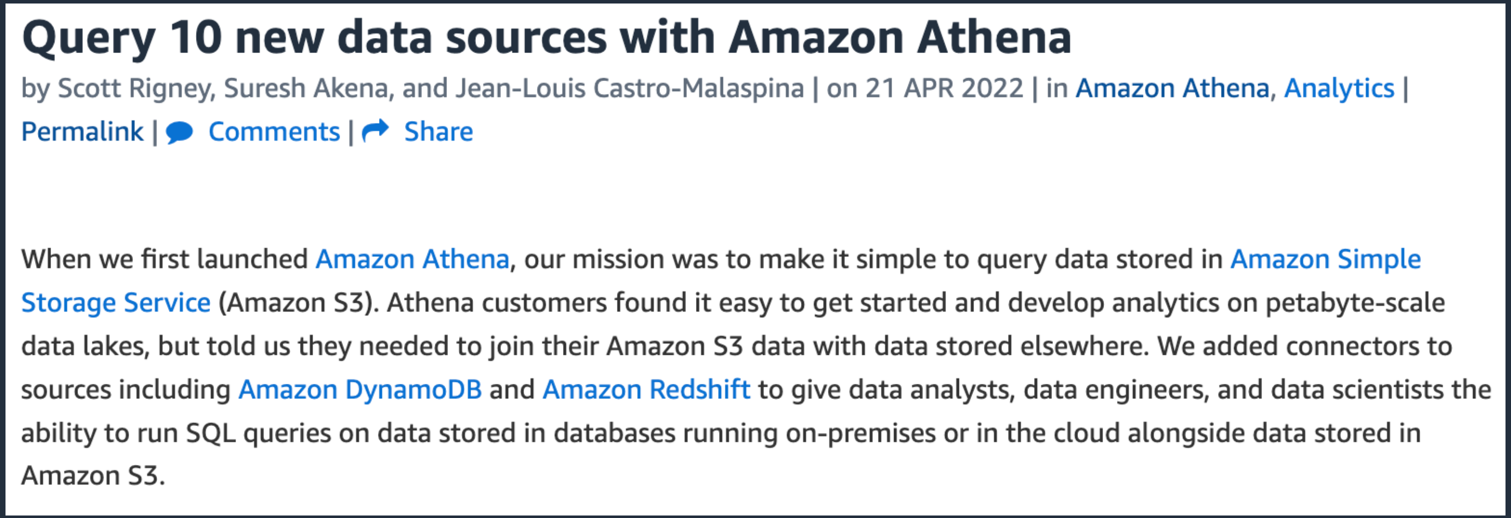
- Useful for quick prototyping
- Add the prefix “lambda:<function_name>”. as the catalog name
- Example: “`SELECT * from “lambda:cmdb”.e2.ec2_instances`” would run a federated query to query our ec2 instance list

# Data source connecters available today

- Hbase
  - Parallelizes by region server and supports predicate pushdown.
- DocumentDB
  - On-the-fly schema inference or configure explicit schema using the Glue Data Catalog.
  - Supports predicate pushdown.
- DynamoDB
  - On-the-fly schema inference or configure explicit schema using the Glue Data Catalog.
  - Supports parallel scan and predicate pushdown.
- JDBC
  - Works with Aurora, MySQL, Postgres, and Redshift and supports parallel scans and predicate pushdown.

# Data source connectors available today (cont’d)

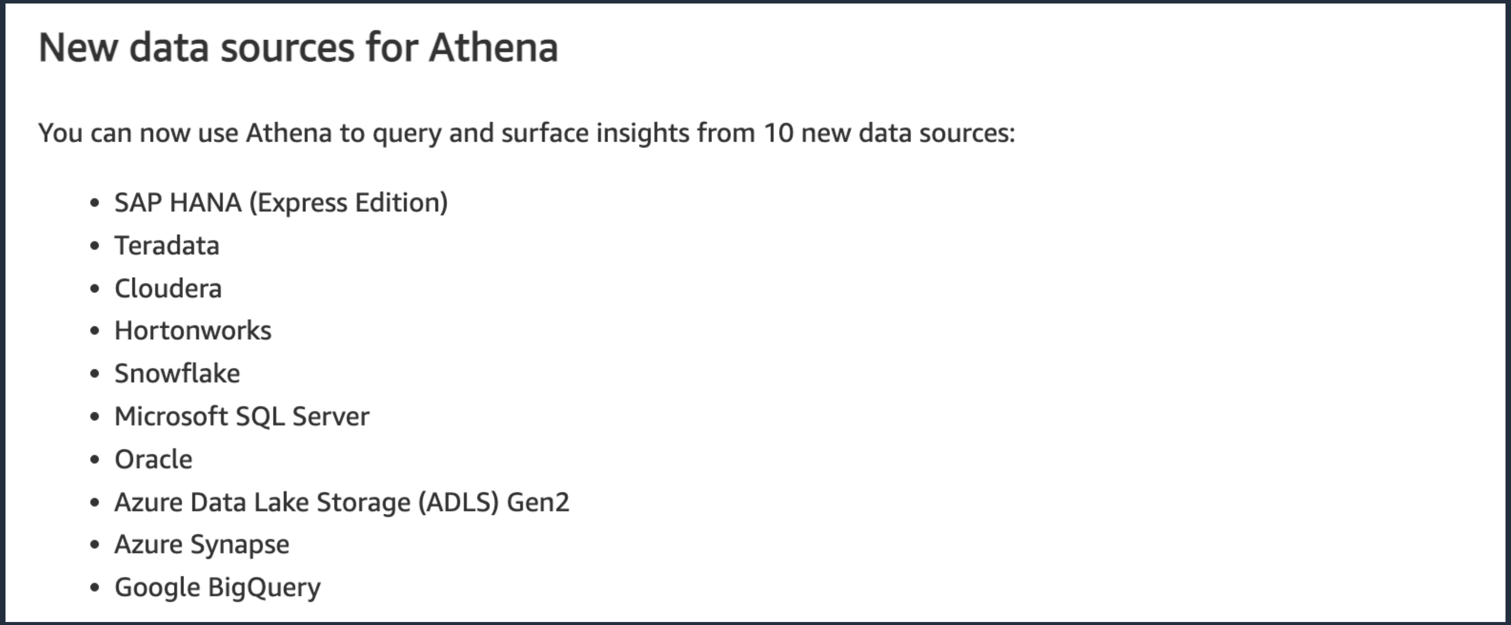
- Redis
  - Use your Redis z-sets, hmaps, or key prefixes to define tables in the Glue Data Catalog and then query them from Athena
- CloudWatch Logs
  - Support parallel scan of log streams, predicate pushdown support, and rich regular expressions
- CloudWatch Metrics
  - Support parallel scan of metric namespaces and dimension as well a predicate pushdown
- TPDS Data Generator
  - Supports parallel scans and predicate pushdown as a reference implementation for building your own connector



# Query 10 new data sources with Amazon Athena

by Scott Rigney, Suresh Akena, and Jean-Louis Castro-Malaspina | on 21 APR 2022 | in Amazon Athena, Analytics | Permalink | Comments | Share

When we first launched Amazon Athena, our mission was to make it simple to query data stored in Amazon Simple Storage Service (Amazon S3). Athena customers found it easy to get started and develop analytics on petabyte-scale data lakes, but told us they needed to join their Amazon S3 data with data stored elsewhere. We added connectors to sources including Amazon DynamoDB and Amazon Redshift to give data analysts, data engineers, and data scientists the ability to run SQL queries on data stored in databases running on-premises or in the cloud alongside data stored in Amazon S3.



## New data sources for Athena

You can now use Athena to query and surface insights from 10 new data sources:

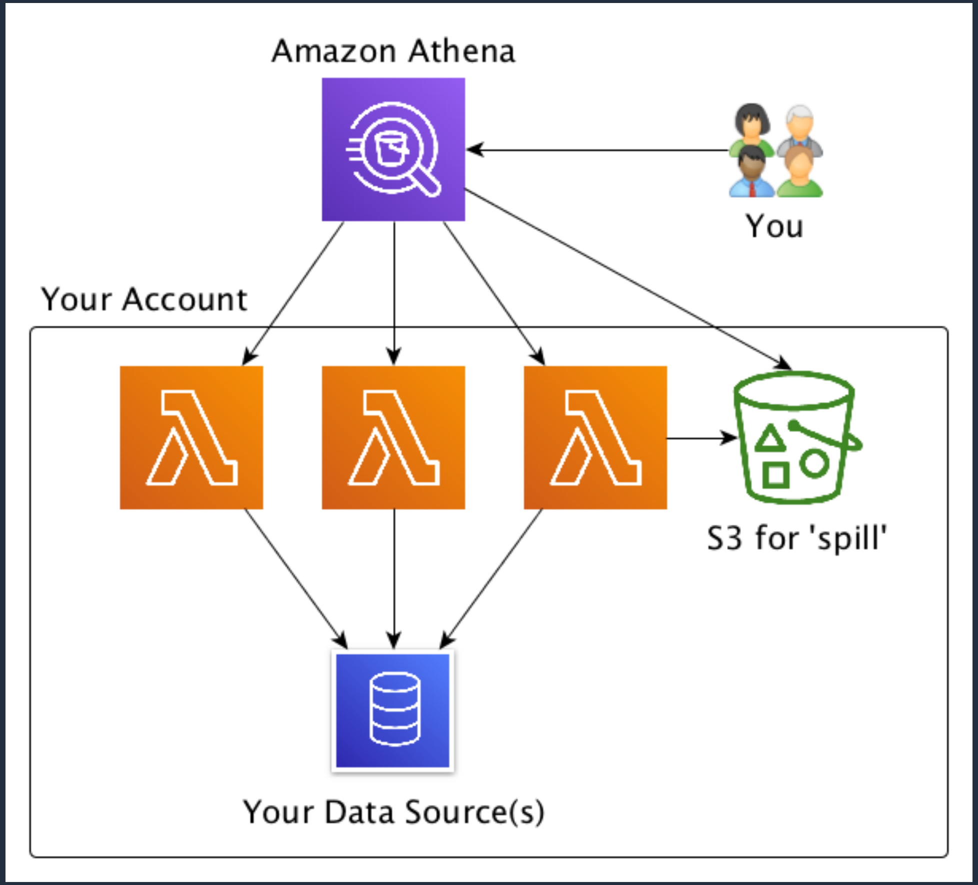
- SAP HANA (Express Edition)
- Teradata
- Cloudera
- Hortonworks
- Snowflake
- Microsoft SQL Server
- Oracle
- Azure Data Lake Storage (ADLS) Gen2
- Azure Synapse
- Google BigQuery

https://aws.amazon.com/blogs/big-data/query-10-new-data-sources-with-amazon-athena/

# Also, build your own data source connector

Use Athena Query Federation SDK and create connector to your own data source

- Features:
  - S3 spill
  - Partition pruning
  - Parallel scans
  - Portable columnar memory-format (Apache Arrow)
  - Authorization
  - Congestion control/avoidance



https://github.com/awslabs/aws-athena-query-federation

# Self-service ETL jobs using federated query

1

One SQL query reading data from multiple sources

Output in S3

2

CTAS and INSERT INTO to create tables and convert to optimized format

3

Schedule using Lambda or build applications

https://aws.amazon.com/blogs/big-data/simplify-etl-data-pipelines-using-amazon-athenas-federated-queries-and-user-defined-functions/

# User Defined Functions (UDFs) in Athena

# What are the challenges without UDFs

- Difficult to pre- or post-process data without UDFs
- Duplication of raw data for access controls to columns (e.g. masking PII)
- Learn and use multiple applications for invoking custom code and using SQL queries for analysis

# Invoke your own functions in Athena queries

- UDFs powered by AWS Lambda
- Network calls supported
- Invoke UDF in SELECT and/or FILTER phase
- Athena optimizes performance, you focus only on processing logic

# UDFs in Athena

Write once

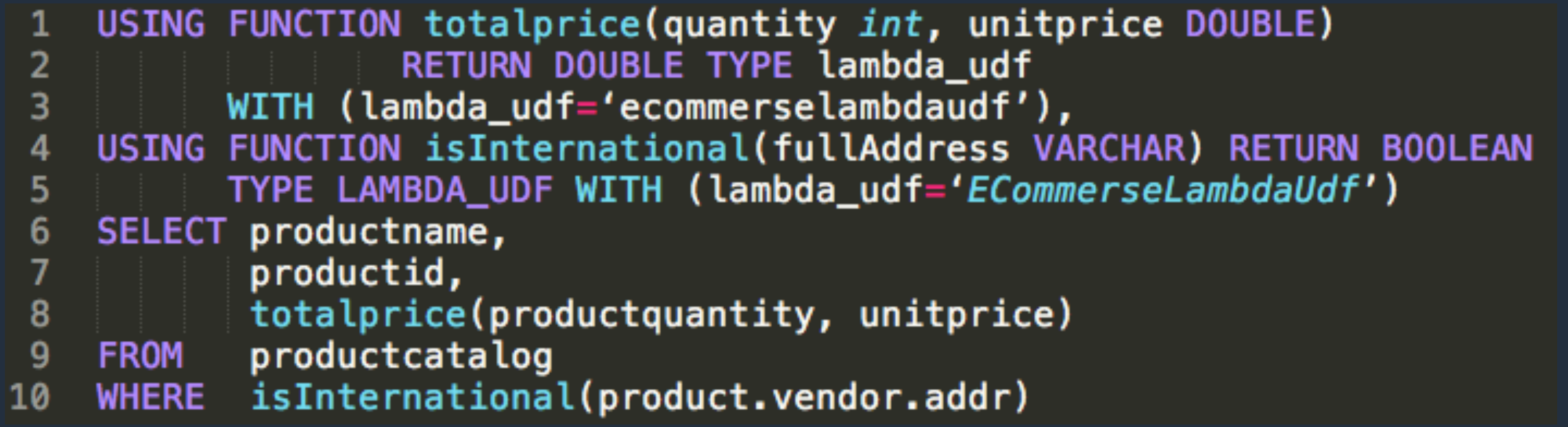
Deploy once

Invoke as many times as needed in a query

# Athena UDFs code sample

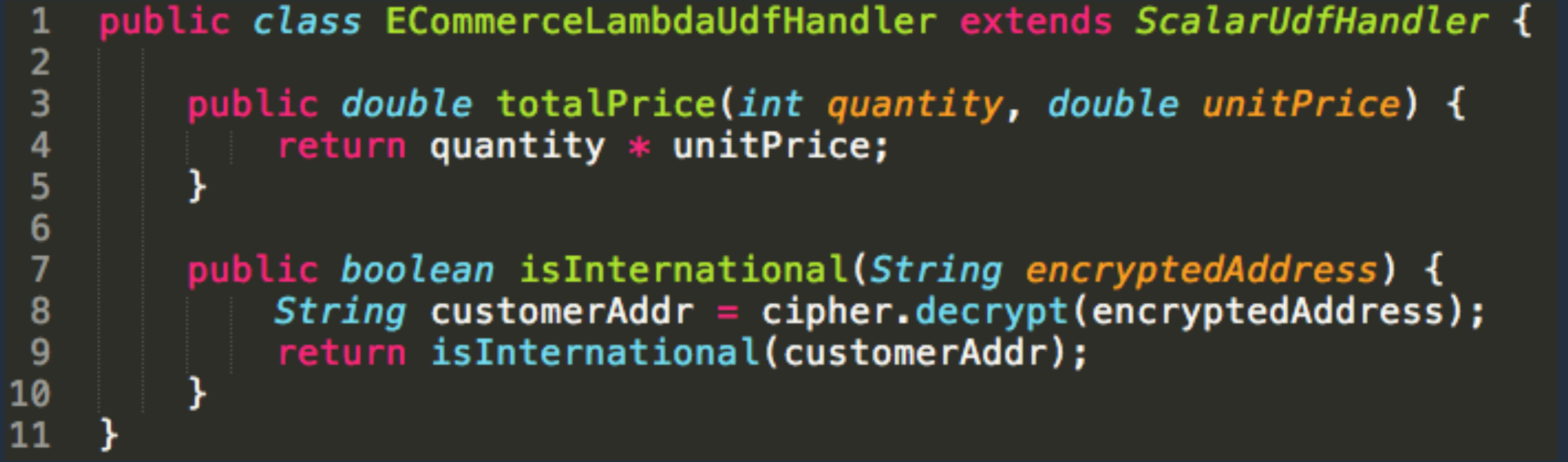
- Simple to write, deploy, and invoke
- Scalar functions
- Powered by AWS Lambda

Athena Query



```
USING FUNCTION totalprice(quantity int, unitprice DOUBLE)
              RETURN DOUBLE TYPE lambda_udf
      WITH (lambda_udf=‘ecommerselambdaudf’),
USING FUNCTION isInternational(fullAddress VARCHAR) RETURN BOOLEAN
      TYPE LAMBDA_UDF WITH (lambda_udf=‘ECommerseLambdaUdf’)
SELECT productname,
       productid,
       totalprice(productquantity, unitprice)
FROM   productcatalog
WHERE  isInternational(product.vendor.addr)
```

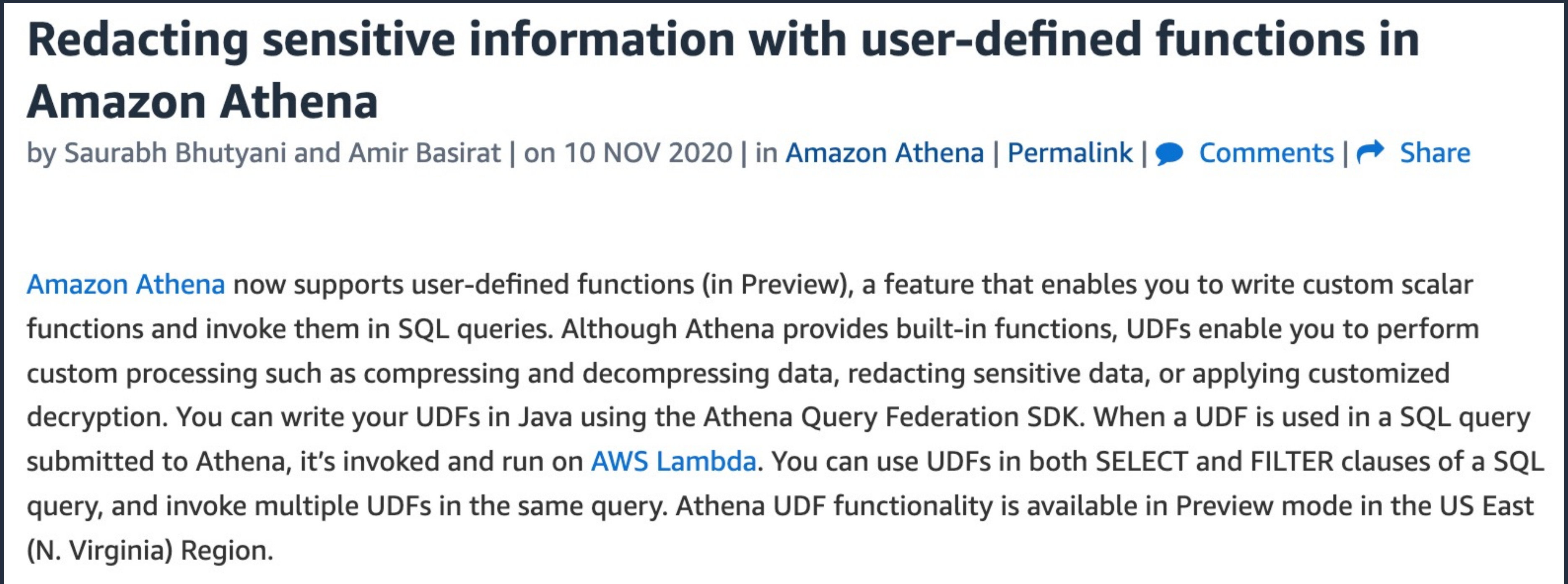
UDF Lambda Code



```
public class ECommerceLambdaUdfHandler extends ScalarUdfHandler {

    public double totalPrice(int quantity, double unitPrice) {
        return quantity * unitPrice;
    }

    public boolean isInternational(String encryptedAddress) {
        String customerAddr = cipher.decrypt(encryptedAddress);
        return isInternational(customerAddr);
    }
}
```



# Redacting sensitive information with user-defined functions in Amazon Athena

by Saurabh Bhutyani and Amir Basirat | on 10 NOV 2020 | in Amazon Athena | Permalink | Comments | Share

Amazon Athena now supports user-defined functions (in Preview), a feature that enables you to write custom scalar functions and invoke them in SQL queries. Although Athena provides built-in functions, UDFs enable you to perform custom processing such as compressing and decompressing data, redacting sensitive data, or applying customized decryption. You can write your UDFs in Java using the Athena Query Federation SDK. When a UDF is used in a SQL query submitted to Athena, it's invoked and run on AWS Lambda. You can use UDFs in both SELECT and FILTER clauses of a SQL query, and invoke multiple UDFs in the same query. Athena UDF functionality is available in Preview mode in the US East (N. Virginia) Region.

https://aws.amazon.com/blogs/big-data/redacting-sensitive-information-with-user-defined-functions-in-amazon-athena/

# ML capabilities in Athena

# Why do you need ML capabilities in Athena

Number of employees:

SQL proficiency > ML proficiency

SQL proficiency > Python proficiency

SQL proficiency > JAVA proficiency

...

…

Running inference in SQL queries is an advantage

# Invoke machine learning models for inference in SQL Queries

- Deploy ML model once on Amazon SageMaker, use n times
- Run inference on data anywhere
- No need to build applications to enable inference
- No additional setup required

# Sample ML use-cases

- Find IP addresses associated with suspicious activity in application logs
- Find products with revenue anomalies (+/-)
- Find suspected fraud in transaction records
- Predict whether a proposed new video game would be a hit

https://aws.amazon.com/blogs/big-data/prepare-data-for-model-training-and-invoke-machine-learning-models-with-amazon-athena/

# Use Athena to train ML model

Federated Athena query to select data from any data source

Transform data using UDFs in Athena

Train and deploy model on Amazon SageMaker

# Use Athena to run inference using ML model

Deploy ML model on SageMaker

Write UDF to pre or post process data

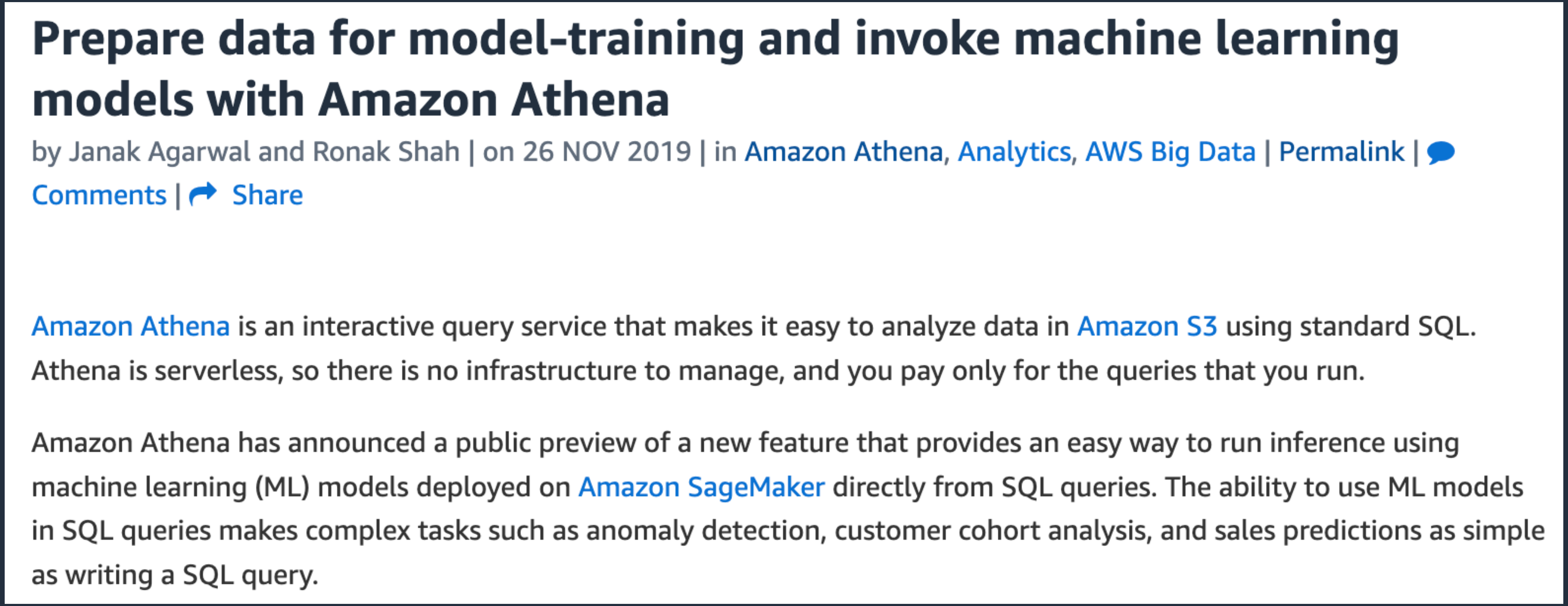
Anyone in organization can run inference on data from any data source

# Sample query to invoke inference

```
USING EXTERNAL FUNCTION predict(platform int, genre int, critic_score int, user_score
int, rating int) returns double TYPE SAGEMAKER_INVOKE_ENDPOINT
WITH (sagemaker_endpoint='xgboost-2019-11-22-00-52-22-742'),

USING EXTERNAL FUNCTION normalize_genre(value VARCHAR) RETURNS int TYPE LAMBDA_INVOKE
WITH (lambda_name= 'VideoNormalization'),

SELECT predict (platform, genre, critic_score, user_score, rating), name
FROM
    (SELECT name,
         normalize_genre(genre) AS genre,
         critic_score,
         user_score,
FROM video_game_data.video_games);
```



# Prepare data for model-training and invoke machine learning models with Amazon Athena

by Janak Agarwal and Ronak Shah | on 26 NOV 2019 | in Amazon Athena, Analytics, AWS Big Data | Permalink | 💬 Comments | ➦ Share

Amazon Athena is an interactive query service that makes it easy to analyze data in Amazon S3 using standard SQL. Athena is serverless, so there is no infrastructure to manage, and you pay only for the queries that you run.

Amazon Athena has announced a public preview of a new feature that provides an easy way to run inference using machine learning (ML) models deployed on Amazon SageMaker directly from SQL queries. The ability to use ML models in SQL queries makes complex tasks such as anomaly detection, customer cohort analysis, and sales predictions as simple as writing a SQL query.

https://aws.amazon.com/blogs/big-data/prepare-data-for-model-training-and-invoke-machine-learning-models-with-amazon-athena/

# Thank you!

Michael Lin
linmicht@amazon.com