Build Your First Big Data Application on AWS

Lex Crosett
Julia Soscia
Eric Ethridge
Natalie Rabinovich
Christian Yarros
Sayali Jojan

aws loft

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

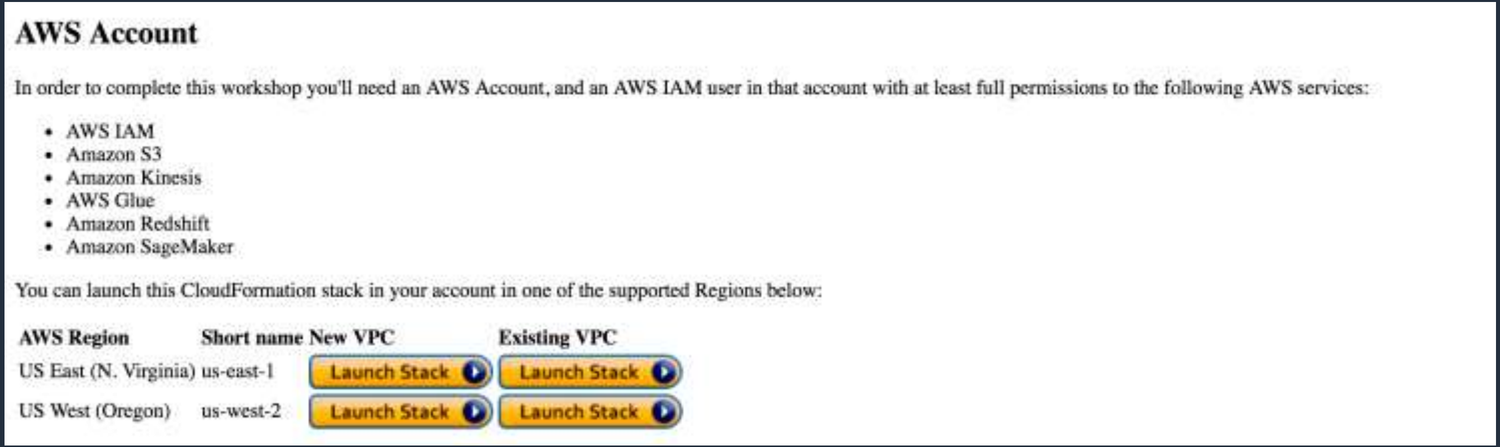
Workshop Agenda

- Download this presentation from Github
- Apply $25 AWS usage credit to your account (5 mins)
- Create AWS resources via CloudFormation (10 mins)
- Review application and AWS services used (45 mins)
- Build Your First Big Data Application on AWS! (90 mins)
- https://github.com/wrbaldwin/da-week

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Launch AWS CloudFormation template

- Browse to the link below and select New VPC
- Select N. Virginia or Oregon or Ireland or Frankfurt for the region to launch your stack.
- https://tinyurl.com/y6v2hjuz



AWS Account

In order to complete this workshop you'll need an AWS Account, and an AWS IAM user in that account with at least full permissions to the following AWS services:

- AWS IAM
- Amazon S3
- Amazon Kinesis
- AWS Glue
- Amazon Redshift
- Amazon SageMaker

You can launch this CloudFormation stack in your account in one of the supported Regions below:

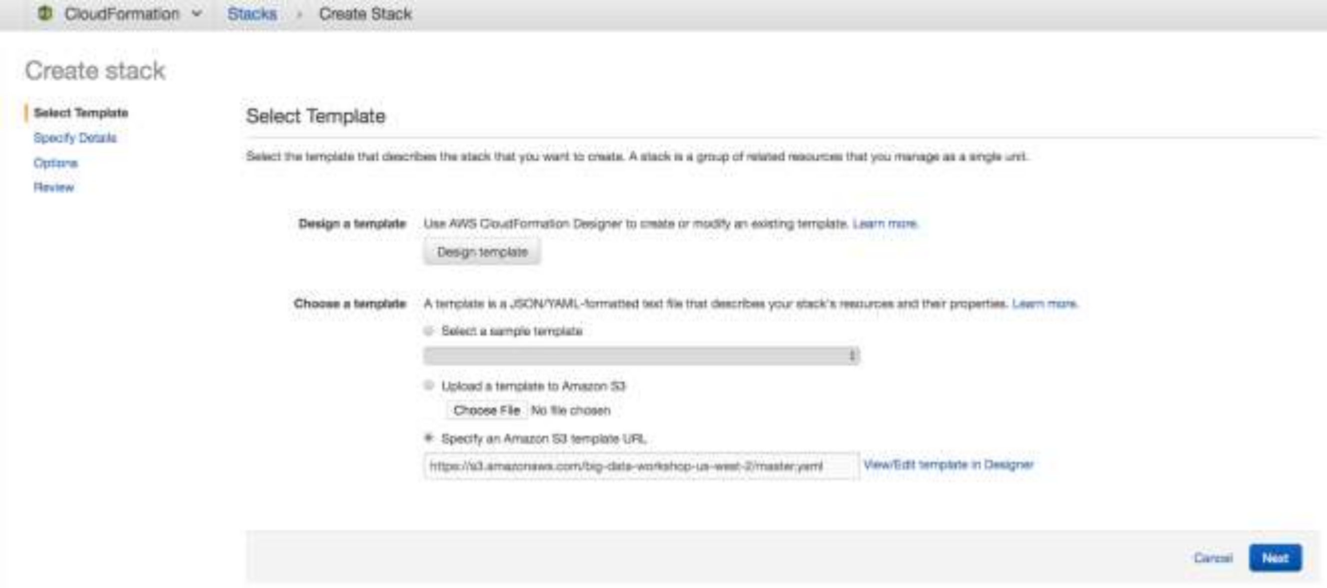
AWS Region	Short name	New VPC	Existing VPC
US East (N. Virginia)	us-east-1	Launch Stack	Launch Stack
US West (Oregon)	us-west-2	Launch Stack	Launch Stack

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Launch AWS CloudFormation template - Continued

- After selecting New or Existing VPC, you will be launched into the AWS CloudFormation service

- Verify the Amazon Simple Storage Service (Amazon S3) template URL, and click “Next”



Create stack

Select Template

Select the template that describes the stack that you want to create. A stack is a group of related resources that you manage as a single unit.

Design a template Use AWS CloudFormation Designer to create or modify an existing template. Learn more.

Choose a template A template is a JSON/YAML-formatted text file that describes your stack's resources and their properties. Learn more.

○ Select a sample template

○ Upload a template to Amazon S3

No file chosen

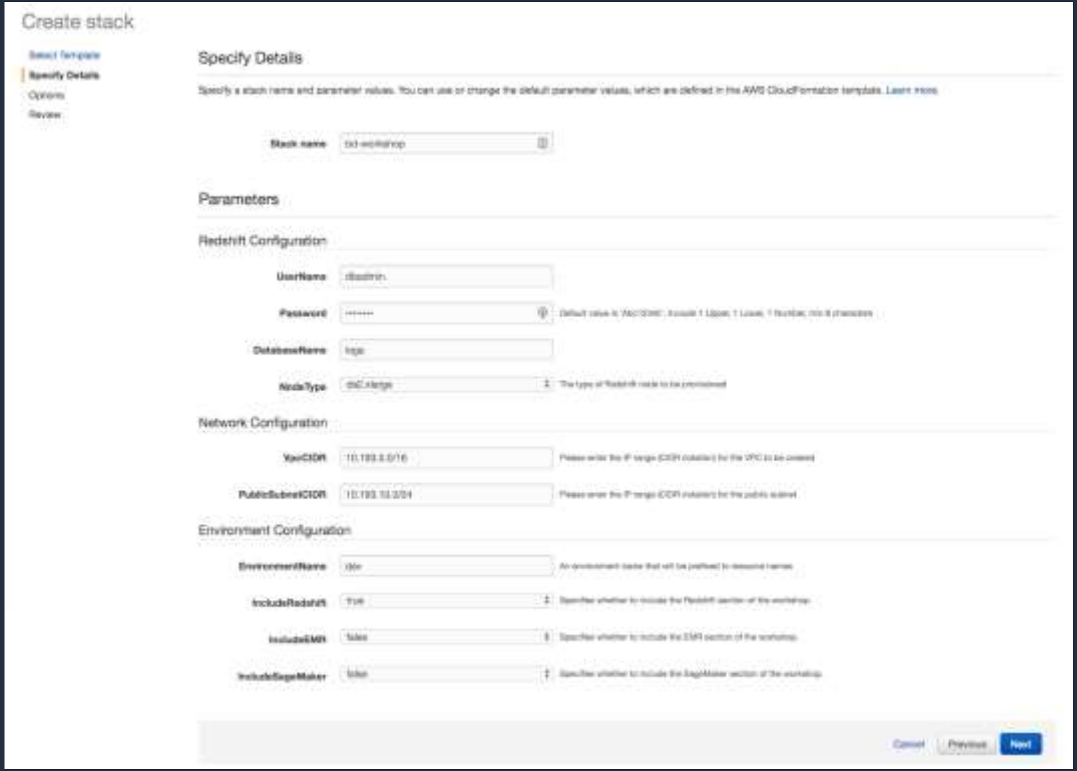
● Specify an Amazon S3 template URL

https://s3.amazonaws.com/big-data-workshop-us-west-2/master.yaml

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Launch AWS CloudFormation template - Continued

- All defaults can be left as-is for this lab. Depending on if you selected new or existing, the only change is picking a CIDR range for the VPC or selecting an existing VPC in your account.
- If there is a Amazon Redshift password in there, leave as is.
- If there is no password, enter Abc12345 as the password. Make a note of this for later.
- Click “Next”



Create stack

Specify Details

Specify a stack name and parameter values. You can use or change the default parameter values, which are defined in the AWS CloudFormation template. Learn more.

Stack name: bd-workshop

Parameters

Redshift Configuration

UserName: dbadmin

Password: •••••••• Default value is 'Abc12345'. Include 1 Upper, 1 Lower, 1 Number, min 8 characters

DatabaseName: [illegible]

NodeType: dc2.large The type of Redshift node to be provisioned

Network Configuration

VpcCIDR: 10.193.0.0/16 Please enter the IP range (CIDR notation) for the VPC to be created

PublicSubnetCIDR: 10.193.10.0/24 Please enter the IP range (CIDR notation) for the public subnet

Environment Configuration

EnvironmentName: dev An environment name that will be prefixed to resource names

IncludeRedshift: true Specifies whether to include the Redshift section of the workshop.

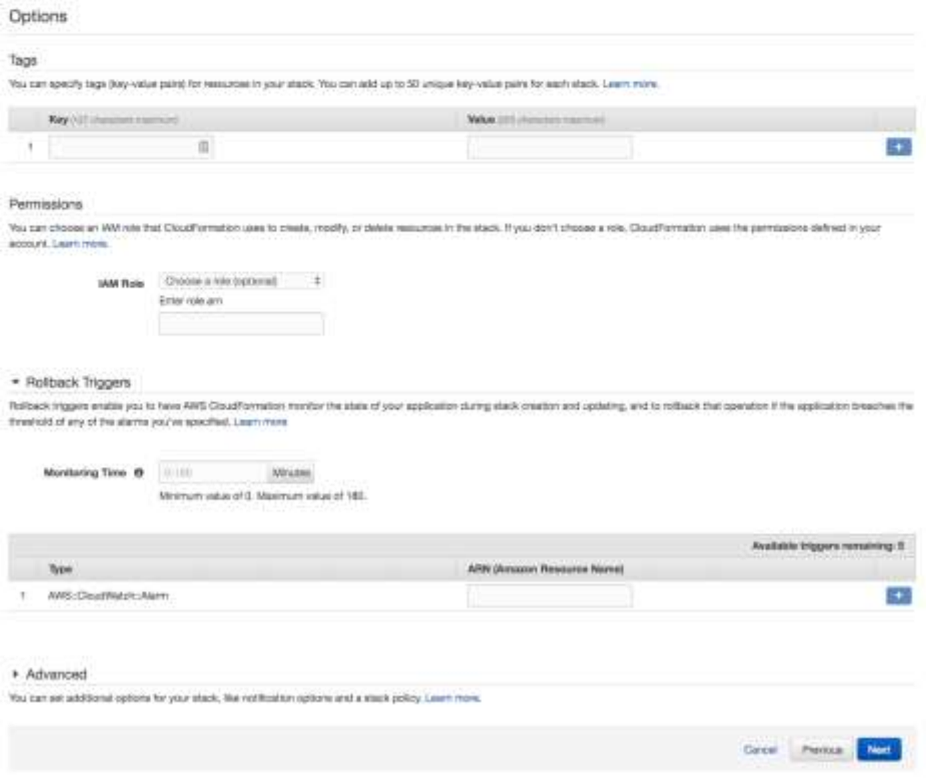
IncludeEMR: false Specifies whether to include the EMR section of the workshop.

IncludeSageMaker: false Specifies whether to include the SageMaker section of the workshop.

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Launch AWS CloudFormation template - Continued

- Leave the Options at their defaults and click “Next”.



Options

Tags

You can specify tags (key-value pairs) for resources in your stack. You can add up to 50 unique key-value pairs for each stack. Learn more.

	Key (127 characters maximum)	Value (255 characters maximum)	
1			+

Permissions

You can choose an IAM role that CloudFormation uses to create, modify, or delete resources in the stack. If you don't choose a role, CloudFormation uses the permissions defined in your account. Learn more.

IAM Role: Choose a role (optional)

Enter role arn

▾ Rollback Triggers

Rollback triggers enable you to have AWS CloudFormation monitor the state of your application during stack creation and updating, and to rollback that operation if the application breaches the threshold of any of the alarms you've specified. Learn more

Monitoring Time: 0-180 Minutes

Minimum value of 0. Maximum value of 180.

Available triggers remaining: 5

	Type	ARN (Amazon Resource Name)	
1	AWS::CloudWatch::Alarm		+

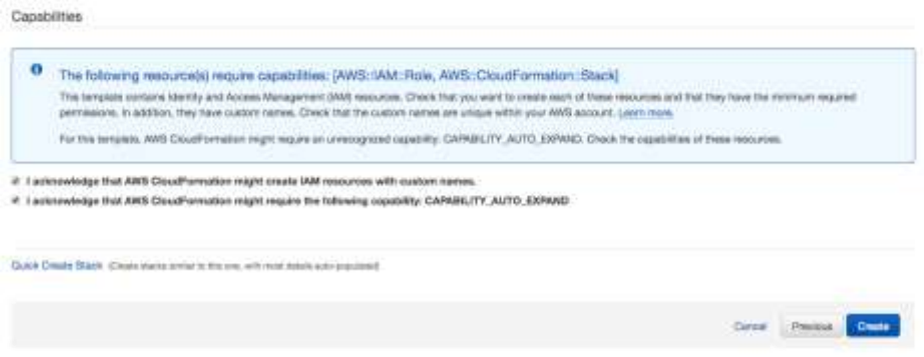
▸ Advanced

You can set additional options for your stack, like notification options and a stack policy. Learn more.

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Launch AWS CloudFormation template - Continued

- Acknowledge the AWS CloudFormation template will create AWS Identity and Access Management (IAM) resources in your account
- Acknowledge the CAPABILITY_AUTO_EXPAND as well.
- Click “Create”
- Verify the stacks are in “CREATE_IN_PROGRESS“ status.
- Wait for all CREATE_COMPLETE (approx. 15 minutes)



Capabilities

The following resource(s) require capabilities: [AWS::IAM::Role, AWS::CloudFormation::Stack]

This template contains Identity and Access Management (IAM) resources. Check that you want to create each of these resources and that they have the minimum required permissions. In addition, they have custom names. Check that the custom names are unique within your AWS account. Learn more.

For this template, AWS CloudFormation might require an unrecognized capability: CAPABILITY_AUTO_EXPAND. Check the capabilities of these resources.

☑ I acknowledge that AWS CloudFormation might create IAM resources with custom names.

☑ I acknowledge that AWS CloudFormation might require the following capability: CAPABILITY_AUTO_EXPAND

Quick Create Stack (Create stacks similar to this one, with most details auto-populated)

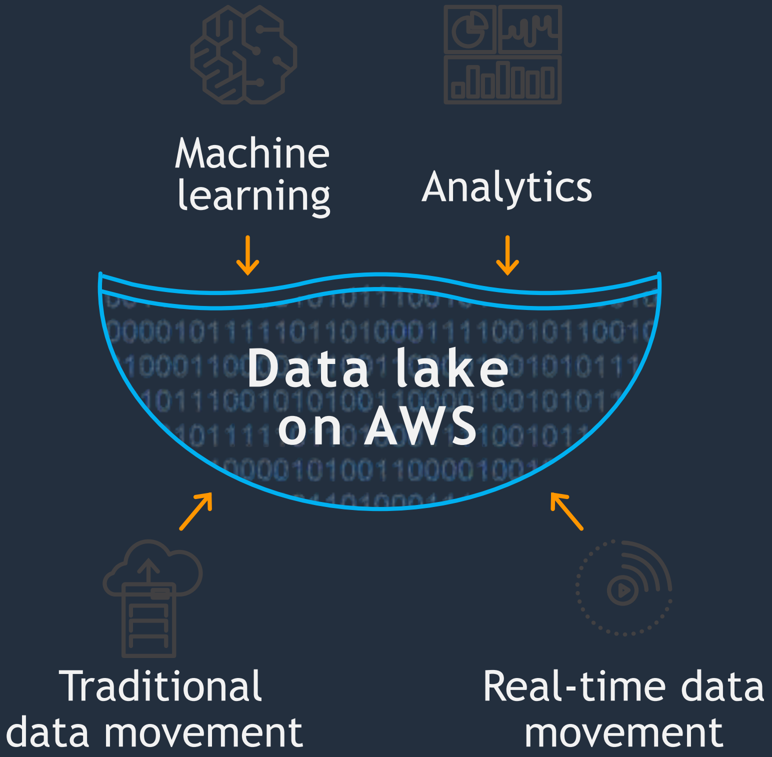
© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Workshop objectives

- Build an end-to-end analytics application on AWS
- Implement batch and real-time layers in a single application
- Explore AWS analytics portfolio for other use cases

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

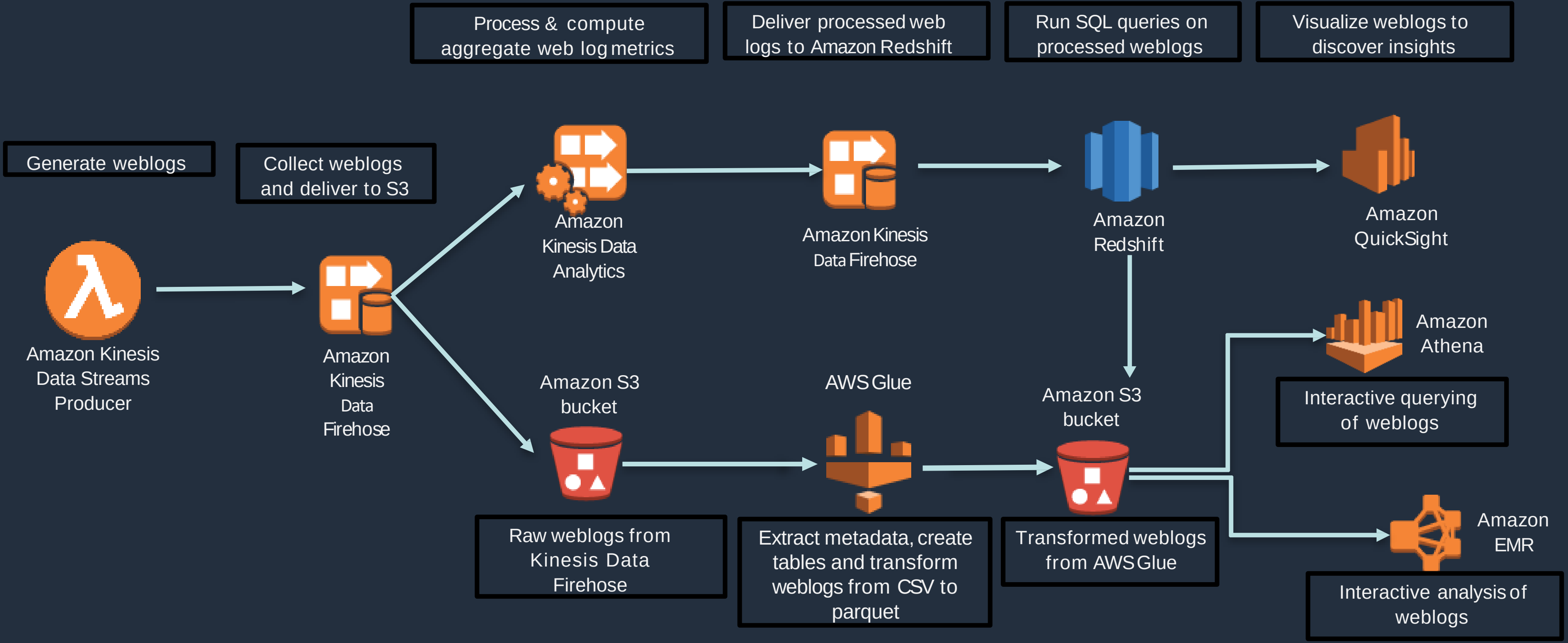
Why analytics on AWS?



- Unmatched durability, and availability at EB scale
- Security, compliance and audit capabilities.
- Object-level controls for fine-grained access
- Fast performance by retrieving subsets of data
- Analyze with broad set of analytics & ML services
- Future proof your data lake architecture

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Your application architecture



Generate weblogs
Collect weblogs and deliver to S3
Process & compute aggregate web log metrics
Deliver processed web logs to Amazon Redshift
Run SQL queries on processed weblogs
Visualize weblogs to discover insights
Amazon Kinesis Data Streams Producer
Amazon Kinesis Data Firehose
Amazon Kinesis Data Analytics
Amazon Kinesis Data Firehose
Amazon Redshift
Amazon QuickSight
Amazon S3 bucket
AWS Glue
Amazon S3 bucket
Amazon Athena
Interactive querying of weblogs
Amazon EMR
Interactive analysis of weblogs
Raw weblogs from Kinesis Data Firehose
Extract metadata, create tables and transform weblogs from CSV to parquet
Transformed weblogs from AWS Glue

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Streaming ingest with Amazon Kinesis

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Streaming with Amazon Kinesis

Easily collect, process, and analyze video and data streams in real time

Kinesis Video Streams

Capture, process, and store video streams

Kinesis Data Streams

Capture, process, and store data streams

Kinesis Data Firehose

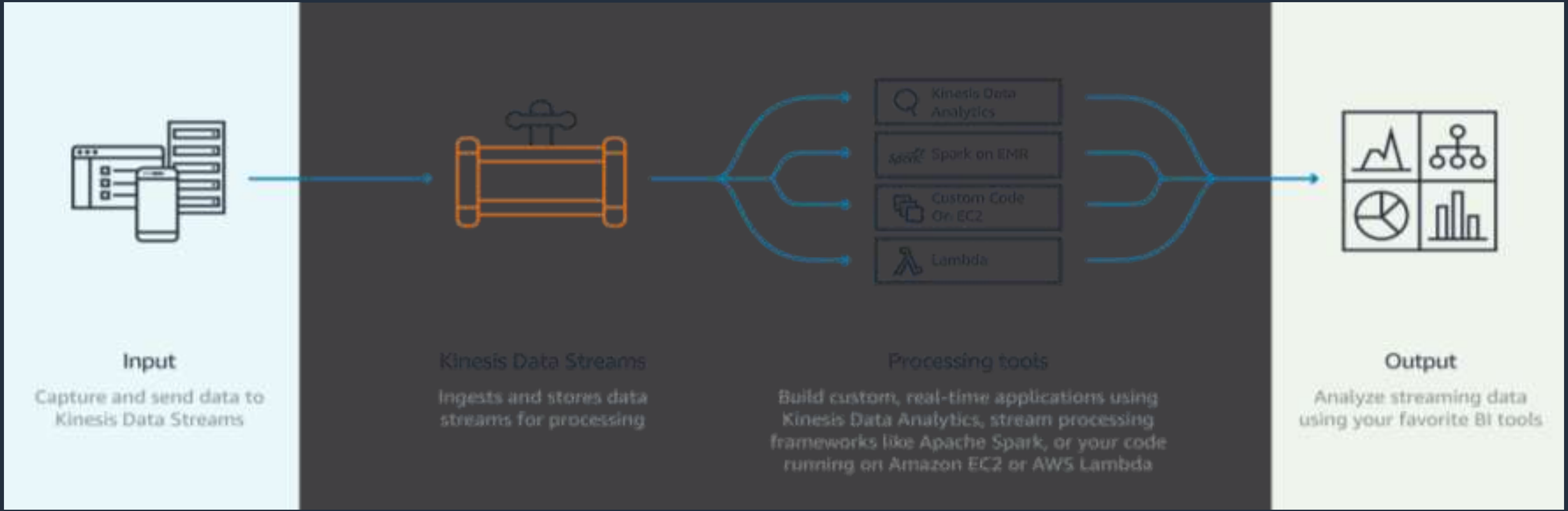
Load data streams into data stores

Kinesis Data Analytics

Analyze data streams with SQL

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Amazon Kinesis Data Streams



Input

Capture and send data to Kinesis Data Streams

Kinesis Data Streams

Ingests and stores data streams for processing

Processing tools

Build custom, real-time applications using Kinesis Data Analytics, stream processing frameworks like Apache Spark, or your code running on Amazon EC2 or AWS Lambda

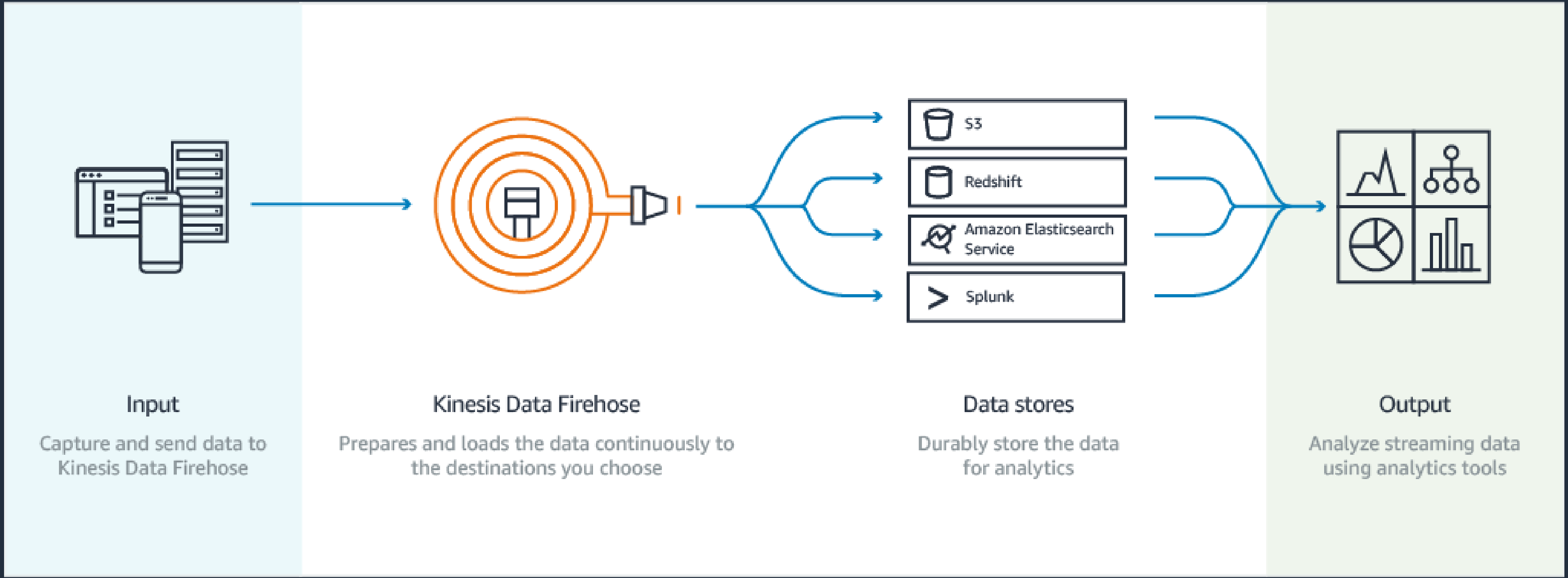
Output

Analyze streaming data using your favorite BI tools

- Easy administration and low cost
- Build real-time applications with framework of choice
- Secure, durable storage

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Amazon Kinesis Data Firehose



Input

Capture and send data to Kinesis Data Firehose

Kinesis Data Firehose

Prepares and loads the data continuously to the destinations you choose

Data stores

Durably store the data for analytics

Output

Analyze streaming data using analytics tools

- Zero administration and seamless elasticity
- Direct-to-data store integration
- Serverless, continuous data transformations

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Amazon Kinesis - Firehose vs. Streams

Kinesis Data Streams

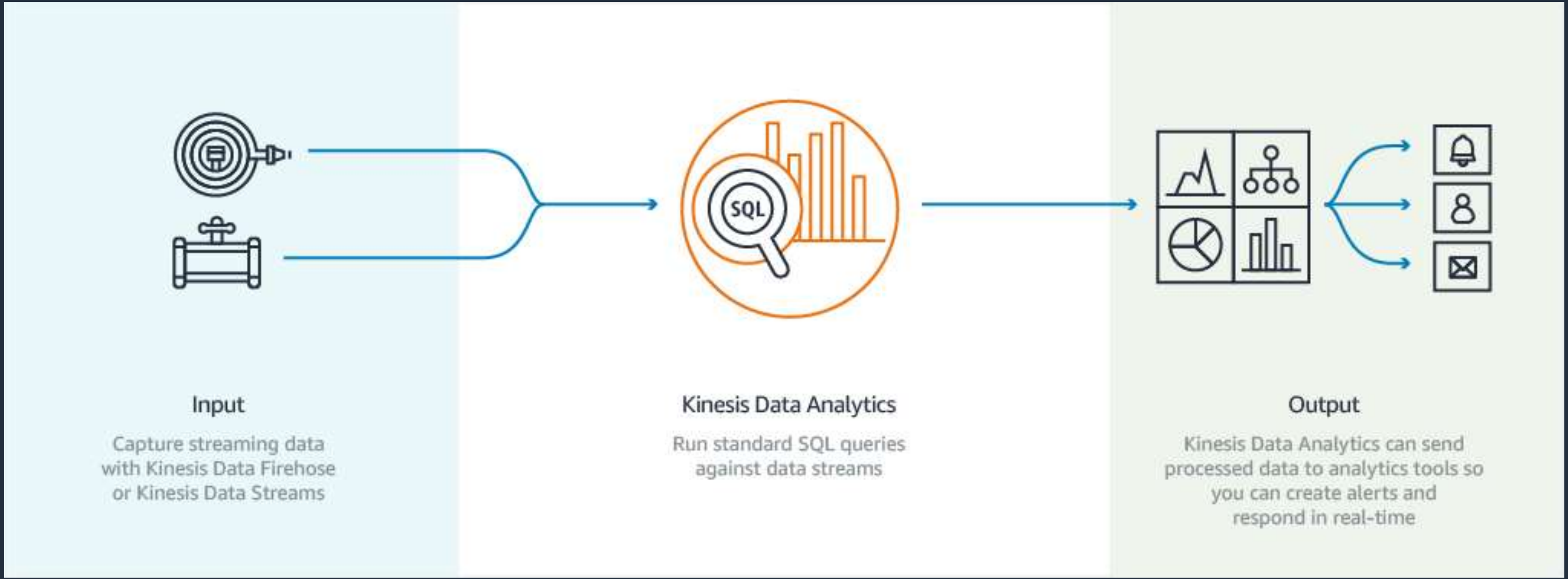
Amazon Kinesis Data Streams is for use cases that require custom processing, per incoming record, with sub-1 second processing latency, and a choice of stream processing frameworks

Kinesis Data Firehose

Amazon Kinesis Data Firehose is for use cases that require zero administration, ability to use existing analytics tools based on Amazon S3, Amazon Redshift, and Amazon ES, and a data latency of 60 seconds or higher

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

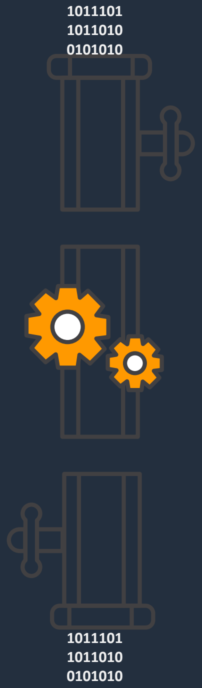
Amazon Kinesis Data Analytics


SQL
Input
Capture streaming data with Kinesis Data Firehose or Kinesis Data Streams
Kinesis Data Analytics
Run standard SQL queries against data streams
Output
Kinesis Data Analytics can send processed data to analytics tools so you can create alerts and respond in real-time

- Powerful real-time applications
- Easy to use, fully managed
- Automatic elasticity
- Windowed aggregations

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Kinesis Data Analytics applications



Connect to streaming source

Easily write SQL code to process streaming data

Continuously deliver SQL results

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Kinesis Data Analytics application metadata

- Note that Amazon Kinesis adds metadata to each record being sent, which was shown in the formatted record sample:
  - The **ROWTIME** represents the time when the Kinesis application inserts a row in the first in-application stream. It’s a special column used for time series analytics. This is also known as the *processing time*.
  - The **APPROXIMATE_ARRIVAL_TIME** is the time the record was added to the streaming source. This is also known as *ingest time* or *server-side time*.
  - The **Event Time** is the timestamp when the event occurred. It’s a also called *client side time*. Its useful because it’s the time when an event occurred at the client.

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Calculate an aggregate metric

Tumbling

- Fixed size and non-overlapping
- Use FLOOR() or STEP() function in a GROUP BY statement

Sliding

- Fixed size and overlapping; row boundaries are determined when new rows enter window
- Use standard OVER and WINDOW clause

Custom

- Not fixed size and overlapping; row boundaries by conditions
- Implementations vary, but typically require two steps (Step 1—identify boundaries, Step 2—perform computation)

Stagger

- Not fixed size and non-overlapping; windows open when the first event matching the partition key arrives
- Use WINDOWED BY STAGGER and PARTITION BY statements

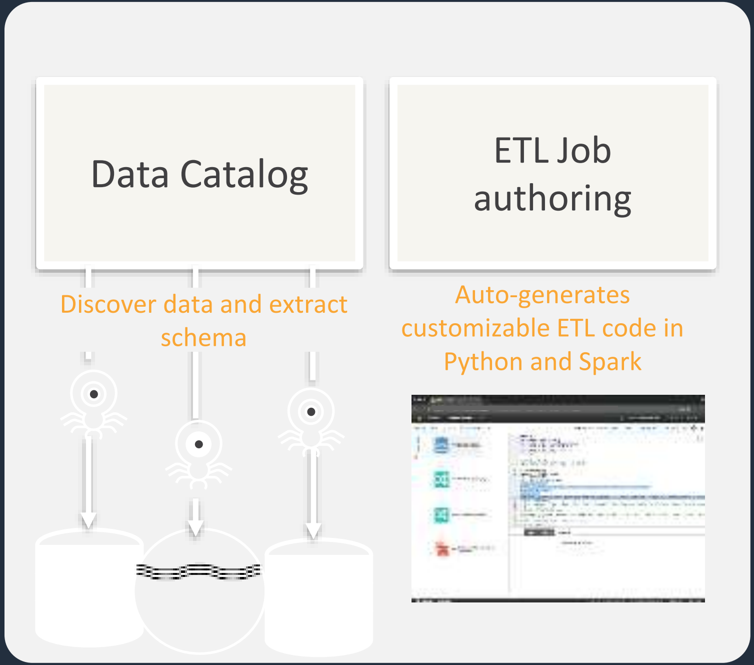
© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Extract, Transform, and Load (ETL) with AWS Glue

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

AWS Glue - Components

Make data discoverable



Data Catalog

Discover data and extract schema

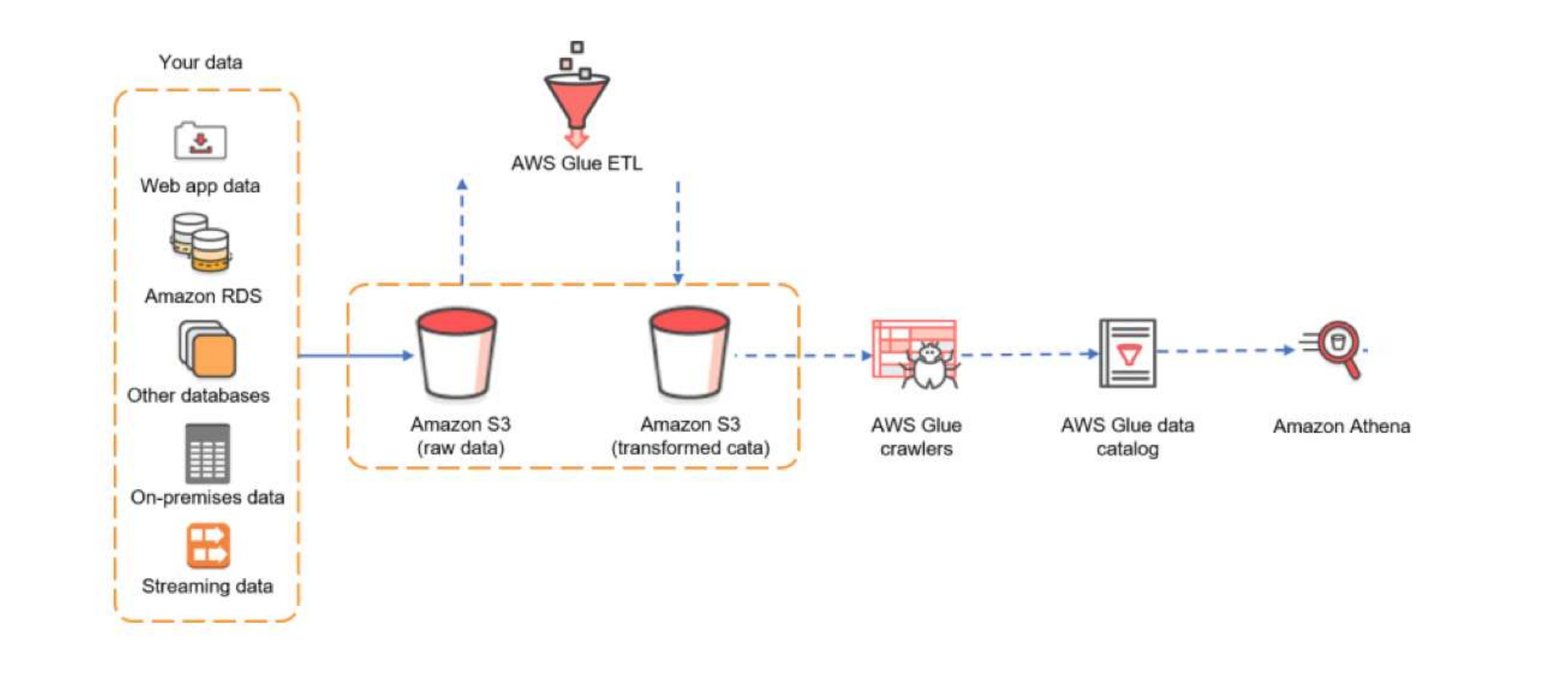
ETL Job authoring

Auto-generates customizable ETL code in Python and Spark

- Automatically discovers data and stores schema
- Catalog makes data searchable and available for ETL
- Catalog contains table and job definitions
- Computes statistics to make queries efficient

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

AWS Glue - How it works



© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Querying data in Amazon S3 with Amazon Athena and Amazon Redshift Spectrum

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Interactive query service

Amazon Athena

- Query directly from Amazon S3
- Use ANSI SQL
- Serverless
- Multiple data formats
- Cost-effective

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Familiar technologies under the covers

Used for SQL queries

In-memory distributed query engine

ANSI-SQL compatible with extensions

Used for DDL functionality

Complex data types

Multitude of formats

Supports data partitioning

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Comparing performance and cost savings for compression and columnar format

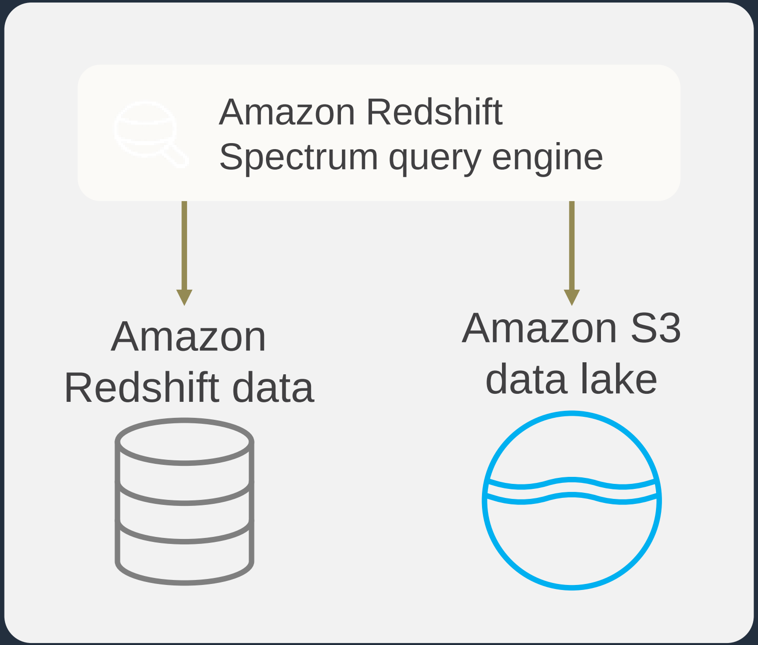
Dataset	Size on Amazon S3	Query run time	Data scanned	Cost
Data stored as text files	1 TB	236 seconds	1.15 TB	$5.75
Data stored in Apache parquet format*	130 GB	6.78 seconds	2.51 GB	$0.013
Savings / Speedup	87% less with parquet	34x faster	99% less data scanned	99.7% savings

(*compressed using Snappy compression)
https://aws.amazon.com/blogs/big-data/analyzing-data-in-s3-using-amazon-athena/

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Amazon Redshift Spectrum

Extend the data warehouse to your Amazon S3 data lake



- Scale compute and storage separately
- Join data across Amazon Redshift and Amazon S3
- Amazon Redshift SQL queries against exabytes in Amazon S3
- Stable query performance and unlimited concurrency
- Parquet, ORC, Grok, Avro, & CSV data formats
- Pay only for the amount of data scanned

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Defining external schema and creating tables

- Define an external schema in Amazon Redshift using the Athena data catalog or your own Apache Hive Metastore

```
CREATE EXTERNAL SCHEMA <schema_name>
```

- Query external tables using <schema_name>.<table_name>

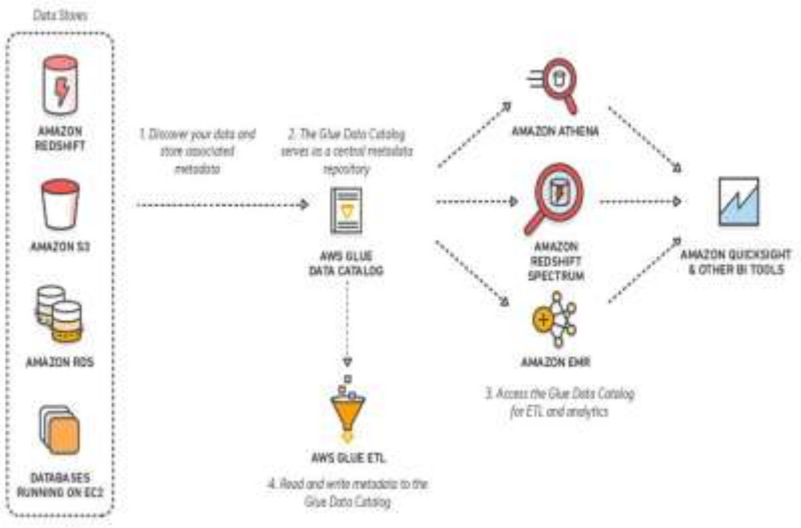
- Register external tables using AWS Glue Data Catalog, your Hive Metastore client, or from Amazon Redshift CREATE EXTERNAL TABLE syntax

```
CREATE EXTERNAL TABLE <table_name>
[PARTITIONED BY <column_name, data_type, …>]
STORED AS file_format
LOCATION s3_location
[TABLE PROPERTIES property_name=property_value, …];
```

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Recap: AWS Glue, Amazon Redshift Spectrum, and Athena

- Used the AWS Glue ETL job to convert and place the raw log data to parquet in S3
- Used the AWS Glue crawler to extract schema for parquet data in Amazon S3 and place in AWS Glue Data Catalog (weblogs_dev.parquet)
- Used the Redshift Spectrum external table to query parquet data in Amazon S3 (spectrum.parquet)
- Used Athena to run queries on the same parquet data in Amazon S3 (weblogs_dev.parquet)
- Both Athena and Redshift Spectrum use the same AWS Glue Data Catalog


Data Stores
AMAZON REDSHIFT
AMAZON S3
AMAZON RDS
DATABASES RUNNING ON EC2
1. Discover your data and store associated metadata
2. The Glue Data Catalog serves as a central metadata repository
AWS GLUE DATA CATALOG
AWS GLUE ETL
4. Read and write metadata to the Glue Data Catalog
AMAZON ATHENA
AMAZON REDSHIFT SPECTRUM
AMAZON EMR
3. Access the Glue Data Catalog for ETL and analytics
AMAZON QUICKSIGHT & OTHER BI TOOLS

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Workshop activities schedule

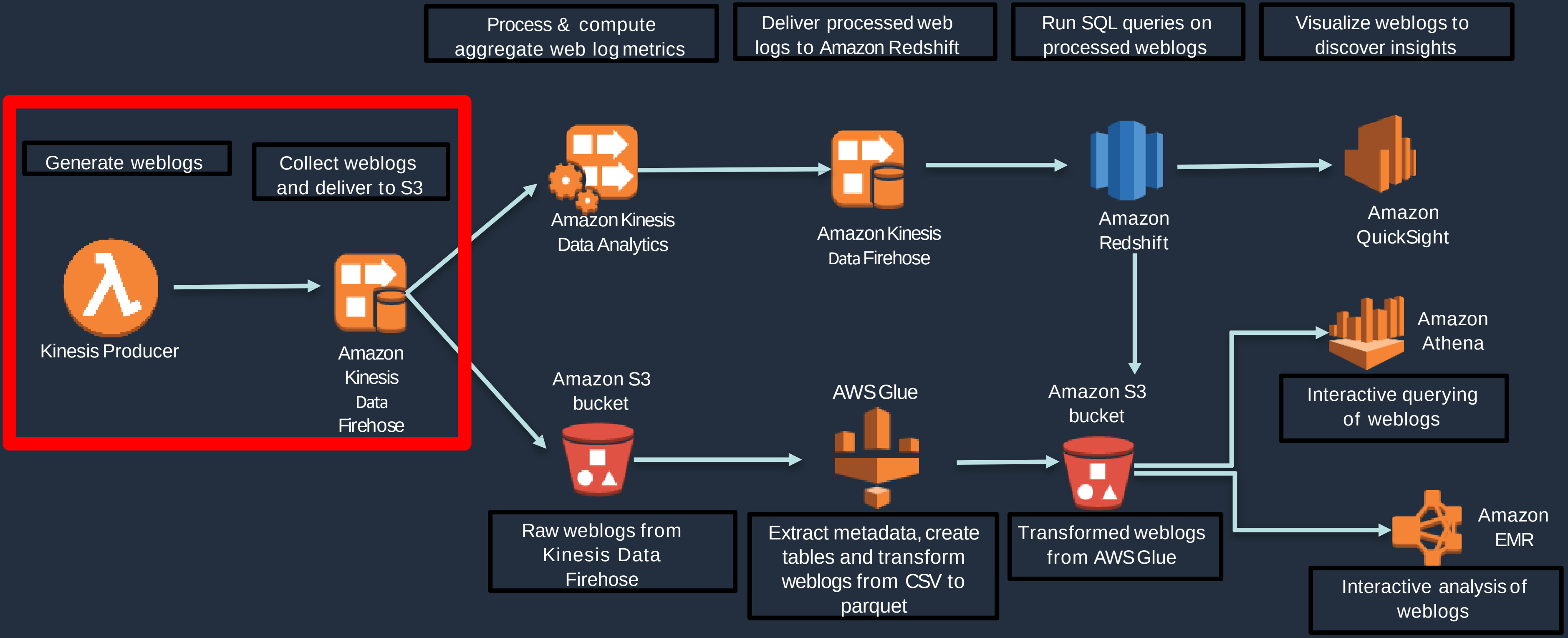
- First big data app workshop activities:
  - 1: Collect logs with Kinesis Data Firehose delivery stream—5 mins
  - 2: Real-time streaming queries using Kinesis Data Analytics—20 mins
  - 3: Deliver streaming results to Amazon Redshift—5 mins
  - 4: Transform weblogs using AWS Glue—30 mins
  - 5: Query parquet data with Amazon Redshift Spectrum and Athena—30 mins
- Total hands-on lab time: 90 mins
- Activities have to be completed in sequence
- Pace yourself
- Helpers are available!

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Activity 1
Collect logs using a Kinesis Data Firehose delivery stream

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Your application architecture


Process & compute aggregate web log metrics
Deliver processed web logs to Amazon Redshift
Run SQL queries on processed weblogs
Visualize weblogs to discover insights
Generate weblogs
Collect weblogs and deliver to S3
Kinesis Producer
Amazon Kinesis Data Firehose
Amazon Kinesis Data Analytics
Amazon Kinesis Data Firehose
Amazon Redshift
Amazon QuickSight
Amazon Athena
Interactive querying of weblogs
Amazon S3 bucket
AWS Glue
Amazon S3 bucket
Amazon EMR
Raw weblogs from Kinesis Data Firehose
Extract metadata, create tables and transform weblogs from CSV to parquet
Transformed weblogs from AWS Glue
Interactive analysis of weblogs

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Collect logs w/a Kinesis Data Firehose delivery stream

Time: 5 minutes

We are going to:

- Write to a Data Firehose delivery stream—Simulate writing transformed Apache weblogs to a Data Firehose delivery stream that is configured to deliver data into an S3 bucket.

- There are **many** different tools and libraries that can be used to write data to a Data Firehose delivery stream. One popular option is called the Amazon Kinesis Agent.

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Collect logs w/a Kinesis Data Firehose delivery stream

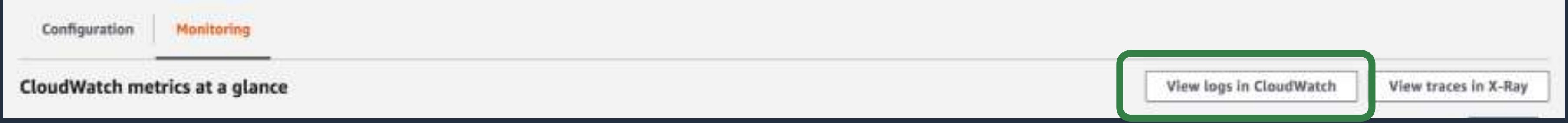
- So that we don’t have to install or set up software on your machine, we are going to use a Lambda function to simulate using the Amazon Kinesis Agent. The Lambda function can populate a Data Firehose delivery stream using a template and is simple to setup.

- **Let’s get started!**

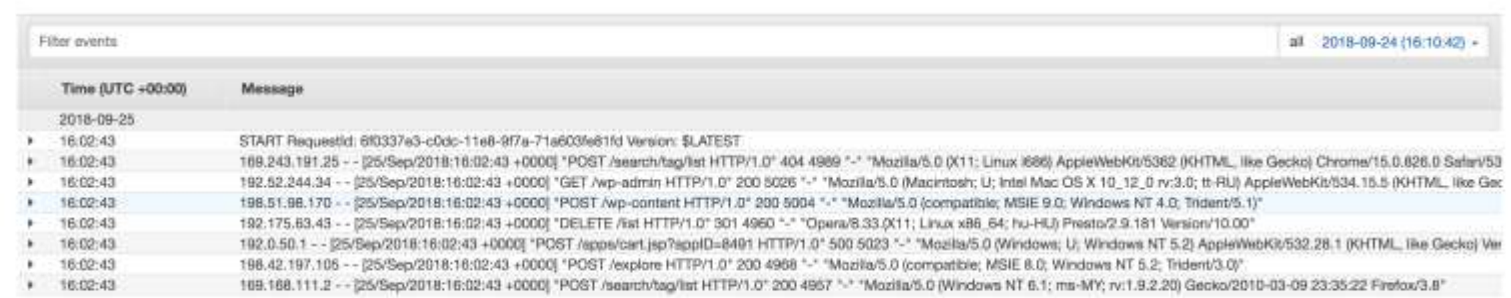
© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Activity 1A: Verify Lambda function delivering logs

- Go to the Lambda console and find a function named like:
  - <StackName>-KinesisStack-xxx-GenerateLogsLambdaFunc-xxxxxxx
- Go to the Monitoring tab and click the “View logs in CloudWatch” button



- View the top log stream and you should see Apache log entries like below:

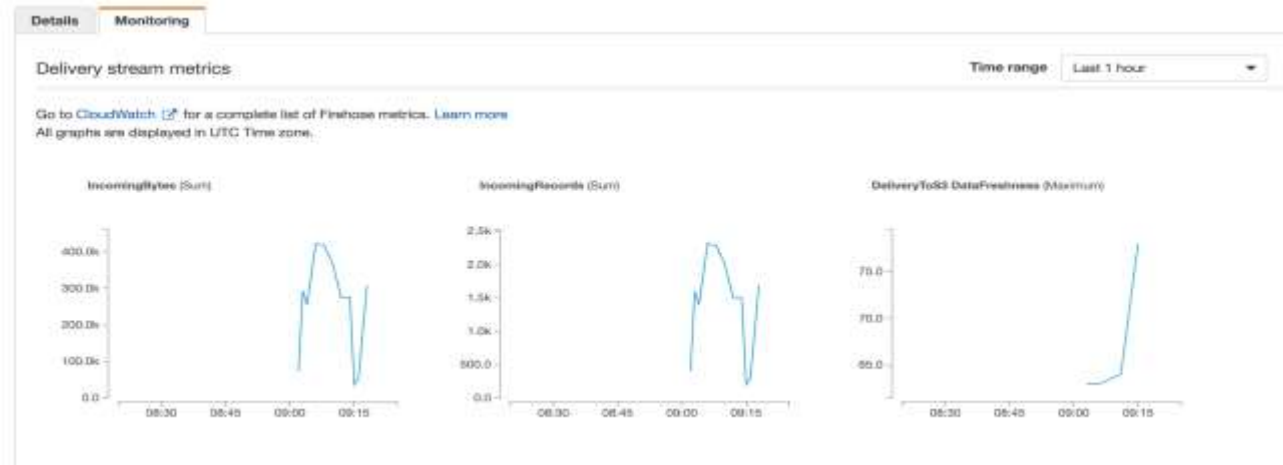


```
Time (UTC +00:00)   Message
2018-09-25
▸ 16:02:43   START RequestId: 6f0337e3-c0dc-11e8-9f7a-71a603fe81fd Version: $LATEST
▸ 16:02:43   169.243.191.25 - - [25/Sep/2018:16:02:43 +0000] "POST /search/tag/list HTTP/1.0" 404 4989 "-" "Mozilla/5.0 (X11; Linux i686) AppleWebKit/5362 (KHTML, like Gecko) Chrome/15.0.826.0 Safari/53
▸ 16:02:43   192.52.244.34 - - [25/Sep/2018:16:02:43 +0000] "GET /wp-admin HTTP/1.0" 200 5026 "-" "Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10_12_0 rv:3.0; tt-RU) AppleWebKit/534.15.5 (KHTML, like Gec
▸ 16:02:43   198.51.98.170 - - [25/Sep/2018:16:02:43 +0000] "POST /wp-content HTTP/1.0" 200 5004 "-" "Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 4.0; Trident/5.1)"
▸ 16:02:43   192.175.63.43 - - [25/Sep/2018:16:02:43 +0000] "DELETE /list HTTP/1.0" 301 4960 "-" "Opera/8.33.(X11; Linux x86_64; hu-HU) Presto/2.9.181 Version/10.00"
▸ 16:02:43   192.0.50.1 - - [25/Sep/2018:16:02:43 +0000] "POST /apps/cart.jsp?appID=8491 HTTP/1.0" 500 5023 "-" "Mozilla/5.0 (Windows; U; Windows NT 5.2) AppleWebKit/532.28.1 (KHTML, like Gecko) Ver
▸ 16:02:43   198.42.197.106 - - [25/Sep/2018:16:02:43 +0000] "POST /explore HTTP/1.0" 200 4968 "-" "Mozilla/5.0 (compatible; MSIE 8.0; Windows NT 5.2; Trident/3.0)"
▸ 16:02:43   169.168.111.2 - - [25/Sep/2018:16:02:43 +0000] "POST /search/tag/list HTTP/1.0" 200 4957 "-" "Mozilla/5.0 (Windows NT 6.1; ms-MY; rv:1.9.2.20) Gecko/2010-03-09 23:35:22 Firefox/3.8"
```

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Review: Monitoring Kinesis Data Firehose delivery to Amazon S3

- Go to the Kinesis console, click on “Data Firehose” and find data stream named like:
  - <StackName>-KinesisStack-xxx-FirehoseDeliveryStream-xxxxxxx
- Go to the Monitoring tab and there should be metrics for delivery to Amazon S3. This might take some time to show up.



Delivery stream metrics

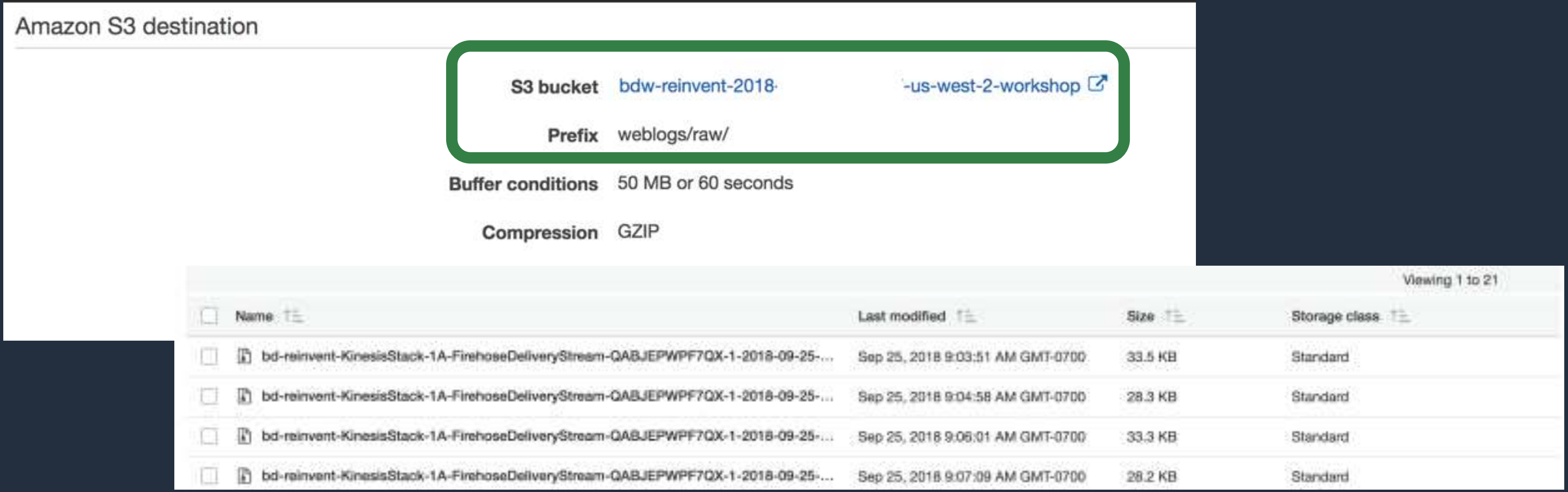
Time range: Last 1 hour

Go to CloudWatch for a complete list of Firehose metrics. Learn more
All graphs are displayed in UTC Time zone.

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Review: Monitoring Kinesis Data Firehose Delivery to Amazon S3

- Go to the Kinesis console and click on “Data Firehose” and find data stream named like:
  - <StackName>-KinesisStack-xxx-FirehoseDeliveryStream-xxxxxxx
- On the details tab you can see the S3 bucket the Apache logs are being streamed to:



Amazon S3 destination

S3 bucket	bdw-reinvent-2018- -us-west-2-workshop
Prefix	weblogs/raw/
Buffer conditions	50 MB or 60 seconds
Compression	GZIP

Viewing 1 to 21

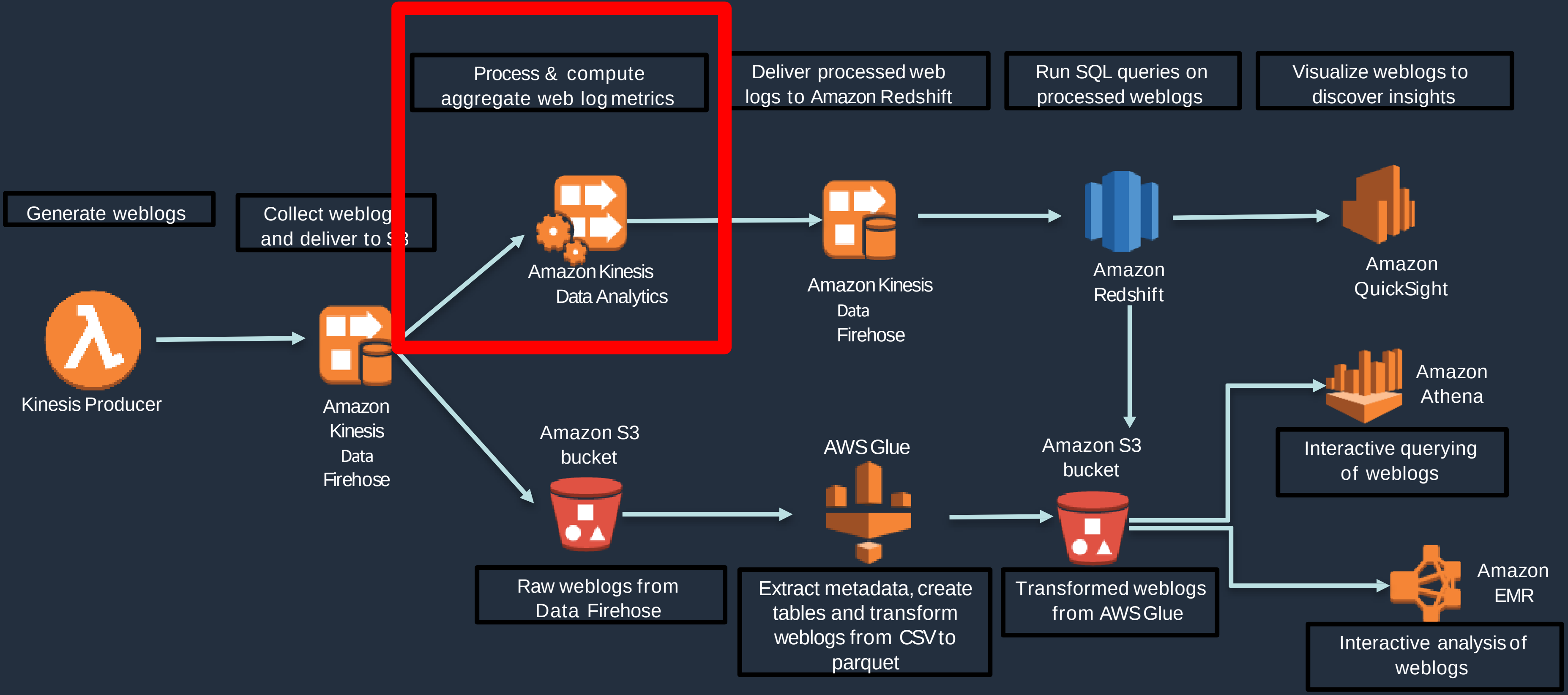
□	Name	Last modified	Size	Storage class
□	bd-reinvent-KinesisStack-1A-FirehoseDeliveryStream-QABJEPWPF7QX-1-2018-09-25-...	Sep 25, 2018 9:03:51 AM GMT-0700	33.5 KB	Standard
□	bd-reinvent-KinesisStack-1A-FirehoseDeliveryStream-QABJEPWPF7QX-1-2018-09-25-...	Sep 25, 2018 9:04:58 AM GMT-0700	28.3 KB	Standard
□	bd-reinvent-KinesisStack-1A-FirehoseDeliveryStream-QABJEPWPF7QX-1-2018-09-25-...	Sep 25, 2018 9:06:01 AM GMT-0700	33.3 KB	Standard
□	bd-reinvent-KinesisStack-1A-FirehoseDeliveryStream-QABJEPWPF7QX-1-2018-09-25-...	Sep 25, 2018 9:07:09 AM GMT-0700	28.2 KB	Standard

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Activity 2
Real-time data processing using Amazon Kinesis Data Analytics

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Your application architecture



Process & compute aggregate web log metrics
Deliver processed web logs to Amazon Redshift
Run SQL queries on processed weblogs
Visualize weblogs to discover insights
Generate weblogs
Collect weblog and deliver to S3
Amazon Kinesis Data Analytics
Amazon Kinesis Data Firehose
Amazon Redshift
Amazon QuickSight
Kinesis Producer
Amazon Kinesis Data Firehose
Amazon Athena
Amazon S3 bucket
AWS Glue
Amazon S3 bucket
Interactive querying of weblogs
Amazon EMR
Raw weblogs from Data Firehose
Extract metadata, create tables and transform weblogs from CSV to parquet
Transformed weblogs from AWS Glue
Interactive analysis of weblogs

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Process data using Kinesis Data Analytics

Time: 20 minutes

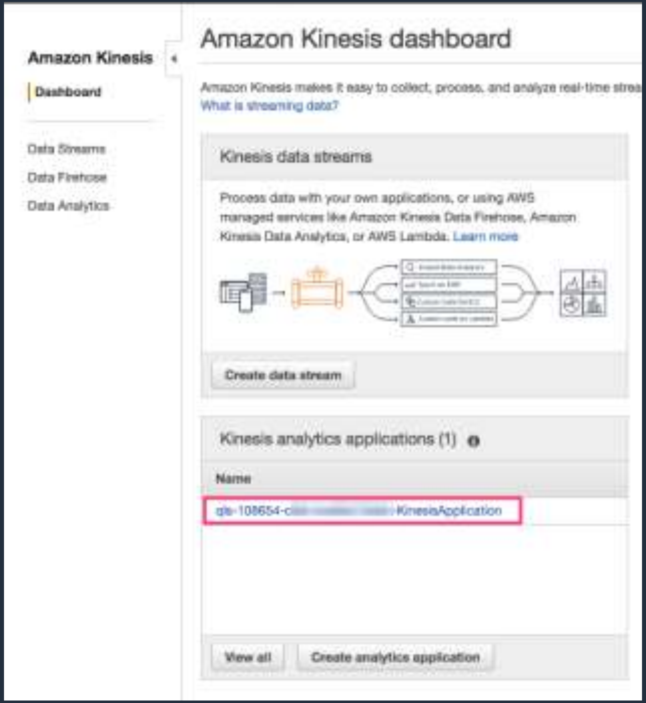
We are going to:

- Write a SQL query to compute an aggregate metric for an interesting statistic on the incoming data
- Write a SQL query using an anomaly detection function

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Activity 2A: Start Amazon Kinesis Data Analytics app

- Navigate to the Kinesis dashboard
- Click on the Kinesis Data Analytics application



Amazon Kinesis dashboard

Amazon Kinesis makes it easy to collect, process, and analyze real-time strea

What is streaming data?

Kinesis data streams

Process data with your own applications, or using AWS managed services like Amazon Kinesis Data Firehose, Amazon Kinesis Data Analytics, or AWS Lambda. Learn more

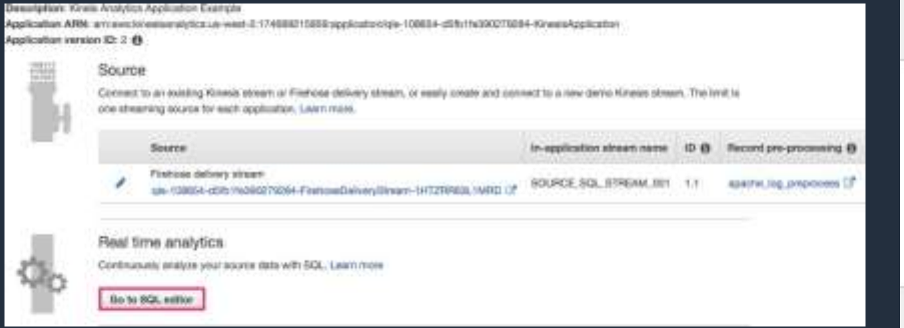
Kinesis analytics applications (1)

Name
qls-108654-c[redacted]-KinesisApplication

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Activity 2A: Start Kinesis app

- Click on **“Go to SQL editor”**
- On the next screen, click on **“Yes, start application”**



Description: Kinesis Analytics Application Example
Application ARN: arn:aws:kinesisanalytics:us-west-2:174668215888:application/qls-108654-d5fb1fe390279284-KinesisApplication
Application version ID: 2

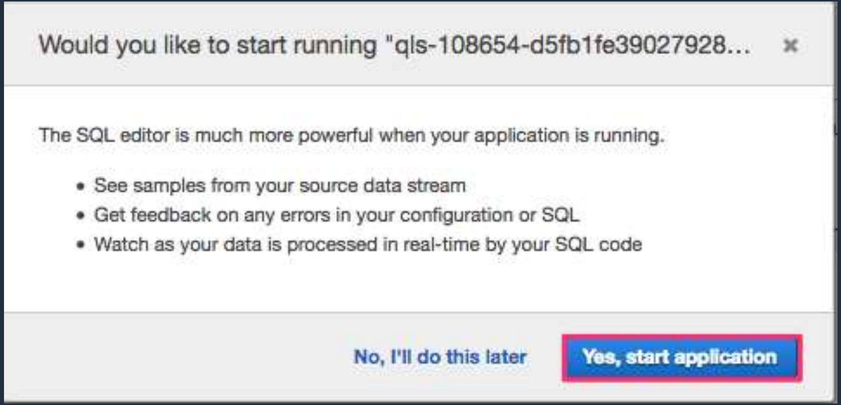
Source

Connect to an existing Kinesis stream or Firehose delivery stream, or easily create and connect to a new demo Kinesis stream. The limit is one streaming source for each application. Learn more.

Source	In-application stream name	ID	Record pre-processing
Firehose delivery stream qls-108654-d5fb1fe390279284-FirehoseDeliveryStream-1HTZRR69L1MRD	SOURCE_SQL_STREAM_001	1.1	apache_log_preprocess

Real time analytics

Continuously analyze your source data with SQL. Learn more



Would you like to start running "qls-108654-d5fb1fe39027928...

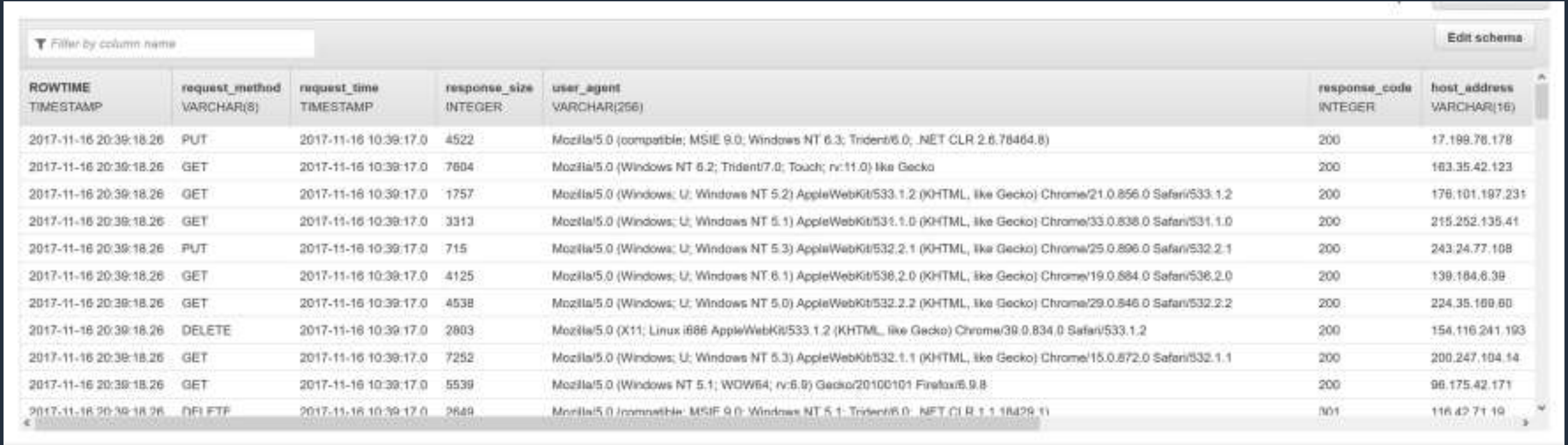
The SQL editor is much more powerful when your application is running.

- See samples from your source data stream
- Get feedback on any errors in your configuration or SQL
- Watch as your data is processed in real-time by your SQL code

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

View sample records in Kinesis app

- Review sample records delivered to the source stream (SOURCE_SQL_STREAM_001)

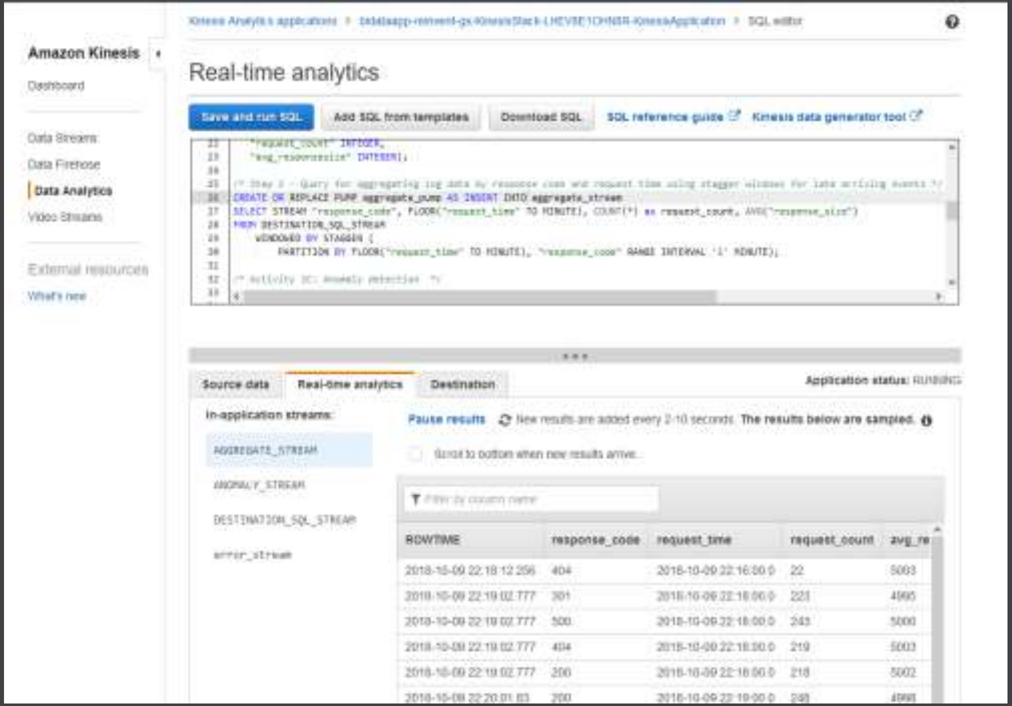


ROWTIME	request_method	request_time	response_size	user_agent	response_code	host_address
TIMESTAMP	VARCHAR(8)	TIMESTAMP	INTEGER	VARCHAR(256)	INTEGER	VARCHAR(16)
2017-11-16 20:39:18.26	PUT	2017-11-16 10:39:17.0	4522	Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.3; Trident/6.0; .NET CLR 2.6.78464.8)	200	17.199.78.178
2017-11-16 20:39:18.26	GET	2017-11-16 10:39:17.0	7604	Mozilla/5.0 (Windows NT 6.2; Trident/7.0; Touch; rv:11.0) like Gecko	200	163.35.42.123
2017-11-16 20:39:18.26	GET	2017-11-16 10:39:17.0	1757	Mozilla/5.0 (Windows; U; Windows NT 5.2) AppleWebKit/533.1.2 (KHTML, like Gecko) Chrome/21.0.856.0 Safari/533.1.2	200	176.101.197.231
2017-11-16 20:39:18.26	GET	2017-11-16 10:39:17.0	3313	Mozilla/5.0 (Windows; U; Windows NT 5.1) AppleWebKit/531.1.0 (KHTML, like Gecko) Chrome/33.0.838.0 Safari/531.1.0	200	215.252.135.41
2017-11-16 20:39:18.26	PUT	2017-11-16 10:39:17.0	715	Mozilla/5.0 (Windows; U; Windows NT 5.3) AppleWebKit/532.2.1 (KHTML, like Gecko) Chrome/25.0.896.0 Safari/532.2.1	200	243.24.77.108
2017-11-16 20:39:18.26	GET	2017-11-16 10:39:17.0	4125	Mozilla/5.0 (Windows; U; Windows NT 6.1) AppleWebKit/536.2.0 (KHTML, like Gecko) Chrome/19.0.884.0 Safari/536.2.0	200	139.184.6.38
2017-11-16 20:39:18.26	GET	2017-11-16 10:39:17.0	4538	Mozilla/5.0 (Windows; U; Windows NT 5.0) AppleWebKit/532.2.2 (KHTML, like Gecko) Chrome/29.0.846.0 Safari/532.2.2	200	224.35.169.60
2017-11-16 20:39:18.26	DELETE	2017-11-16 10:39:17.0	2803	Mozilla/5.0 (X11; Linux i686 AppleWebKit/533.1.2 (KHTML, like Gecko) Chrome/39.0.834.0 Safari/533.1.2	200	154.116.241.193
2017-11-16 20:39:18.26	GET	2017-11-16 10:39:17.0	7252	Mozilla/5.0 (Windows; U; Windows NT 5.3) AppleWebKit/532.1.1 (KHTML, like Gecko) Chrome/15.0.872.0 Safari/532.1.1	200	200.247.104.14
2017-11-16 20:39:18.26	GET	2017-11-16 10:39:17.0	5539	Mozilla/5.0 (Windows NT 5.1; WOW64; rv:6.9) Gecko/20100101 Firefox/6.9.8	200	96.175.42.171
2017-11-16 20:39:18.26	DELETE	2017-11-16 10:39:17.0	2649	Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 5.1; Trident/6.0; .NET CLR 1.1.18429.1)	301	116.42.71.19

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Activity 2B: Calculate an aggregate metric

- Open the **KinesisAnalyticsSQL** file located in the **Scripts** section (Workshop Scripts)
- Copy and paste the entire SQL in the SQL editor of your Kinesis Data Analytics application, **OVERWRITING** the existing text
- Click “**Save and run SQL**”



Real-time analytics

```
    "request_count" INTEGER,
    "avg_responsesize" INTEGER);

/* Step 2 - Query for aggregating log data by response code and request time using stagger windows for late arriving events */
CREATE OR REPLACE PUMP aggregate_pump AS INSERT INTO aggregate_stream
SELECT STREAM "response_code", FLOOR("request_time" TO MINUTE), COUNT(*) as request_count, AVG("response_size")
FROM DESTINATION_SQL_STREAM
    WINDOWED BY STAGGER (
        PARTITION BY FLOOR("request_time" TO MINUTE), "response_code" RANGE INTERVAL '1' MINUTE);

/* Activity 2C: Anomaly detection */
```

Application status: RUNNING

In-application streams:

AGGREGATE_STREAM

ANOMALY_STREAM

DESTINATION_SQL_STREAM

error_stream

New results are added every 2-10 seconds. **The results below are sampled.**

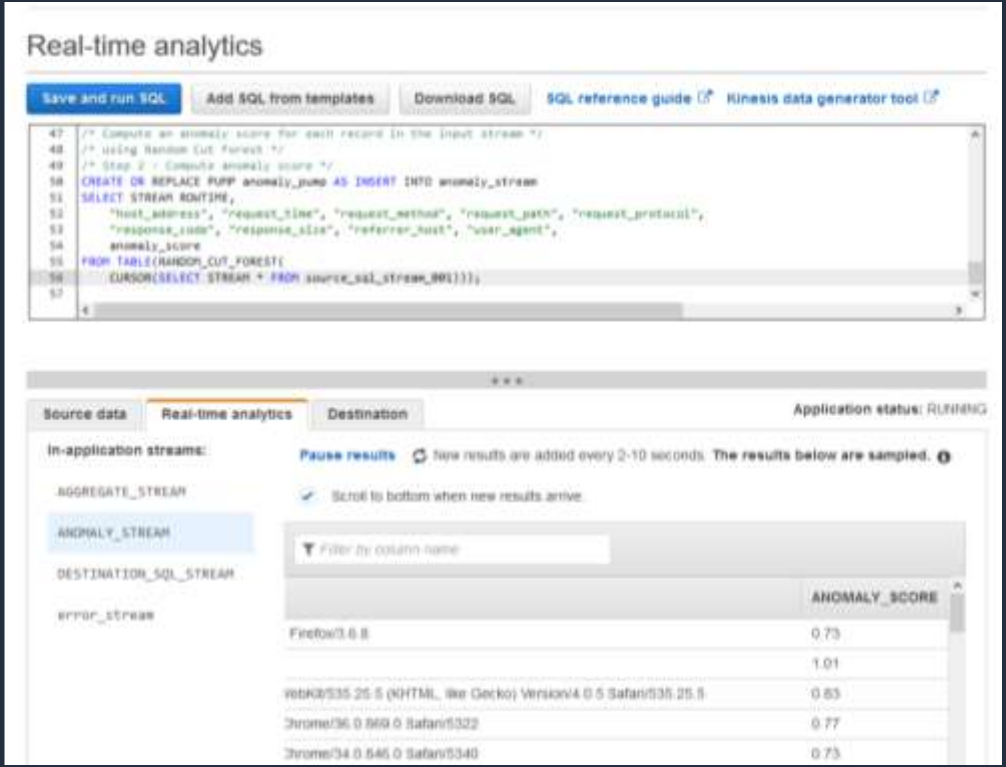
ROWTIME	response_code	request_time	request_count	avg_re
2018-10-09 22:18:12.256	404	2018-10-09 22:16:00.0	22	5003
2018-10-09 22:19:02.777	301	2018-10-09 22:18:00.0	221	4995
2018-10-09 22:19:02.777	500	2018-10-09 22:18:00.0	243	5000
2018-10-09 22:19:02.777	404	2018-10-09 22:18:00.0	219	5003
2018-10-09 22:19:02.777	200	2018-10-09 22:18:00.0	218	5002
2018-10-09 22:20:01.83	200	2018-10-09 22:19:00.0	248	4998

© 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Activity 2C: Anomaly detection

Take a look at the anomaly detection section in the SQL script:

- It creates an anomaly_stream with the attribute anomaly_score
- It calculates the anomaly score for each record in the stream by using the built-in random cut forest function



```
/* Compute an anomaly score for each record in the input stream */
/* using Random Cut Forest */
/* Step 2 : Compute anomaly score */
CREATE OR REPLACE PUMP anomaly_pump AS INSERT INTO anomaly_stream
SELECT STREAM ROWTIME,
    "host_address", "request_time", "request_method", "request_path", "request_protocol",
    "response_code", "response_size", "referrer_host", "user_agent",
    anomaly_score
FROM TABLE(RANDOM_CUT_FOREST(
    CURSOR(SELECT STREAM * FROM source_sql_stream_001)));
```

Application status: RUNNING

In-application streams:

- AGGREGATE_STREAM
- ANOMALY_STREAM
- DESTINATION_SQL_STREAM
- error_stream

New results are added every 2-10 seconds. **The results below are sampled.**

✓ Scroll to bottom when new results arrive

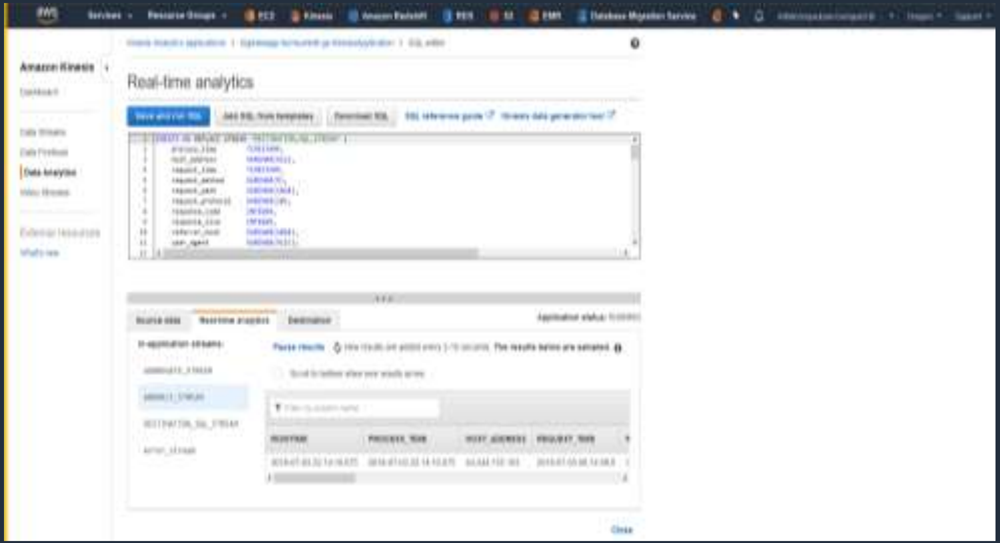
	ANOMALY_SCORE
Firefox/3.6.8	0.73
	1.01
ebKit/535.25.5 (KHTML, like Gecko) Version/4.0.5 Safari/535.25.5	0.83
Chrome/36.0.869.0 Safari/5322	0.77
Chrome/34.0.846.0 Safari/5340	0.73

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Kinesis Data Analytics in-application streams

- In-application streams:

1) Aggregate stream
2) Anomaly stream
3) Destination SQL stream
4) Error stream

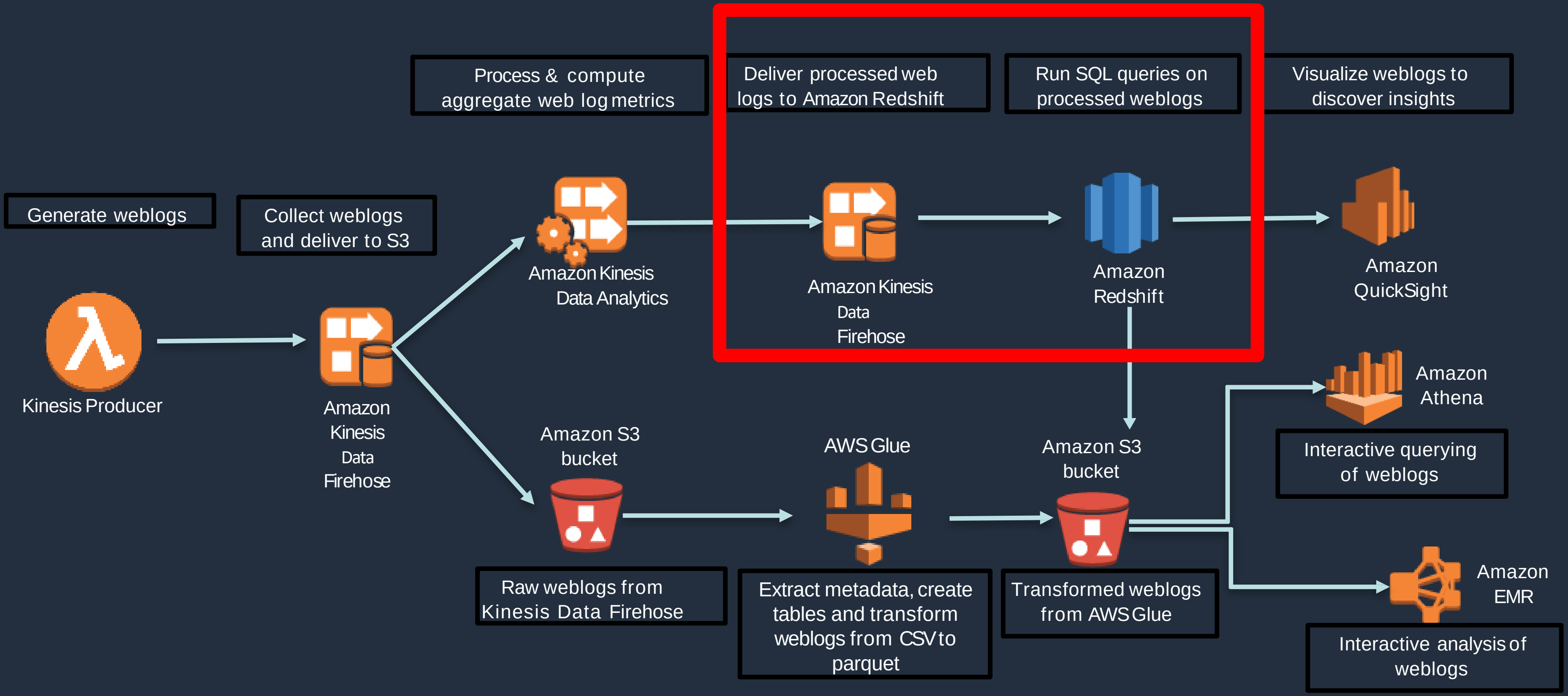


© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Activity 3
Deliver streaming results to Amazon Redshift

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Your application architecture



Generate weblogs

Kinesis Producer

Collect weblogs and deliver to S3

Amazon Kinesis Data Firehose

Process & compute aggregate web log metrics

Amazon Kinesis Data Analytics

Amazon S3 bucket

Raw weblogs from Kinesis Data Firehose

Deliver processed web logs to Amazon Redshift

Amazon Kinesis Data Firehose

AWS Glue

Extract metadata, create tables and transform weblogs from CSV to parquet

Run SQL queries on processed weblogs

Amazon Redshift

Amazon S3 bucket

Transformed weblogs from AWS Glue

Visualize weblogs to discover insights

Amazon QuickSight

Amazon Athena

Interactive querying of weblogs

Amazon EMR

Interactive analysis of weblogs

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Activity 3: Deliver data to Amazon Redshift using Kinesis Data Firehose

Time: 5 minutes

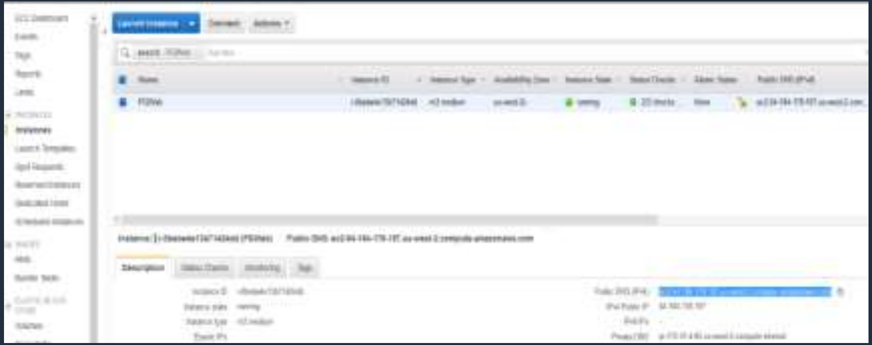
We are going to:

- Connect to Amazon Redshift cluster and create a table to hold web logs data
- Update Kinesis Data Analytics application to send data to Amazon Redshift, via the Data Firehose delivery stream

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Activity 3A: Connect to Amazon Redshift

- You can connect with pgweb
- Already installed for the Amazon Redshift cluster
- Just navigate to pgweb: Services> EC2> Select the pgweb instance> Copy the Public DNS (IPv4)> Paste in your browser> Start interacting with pgweb

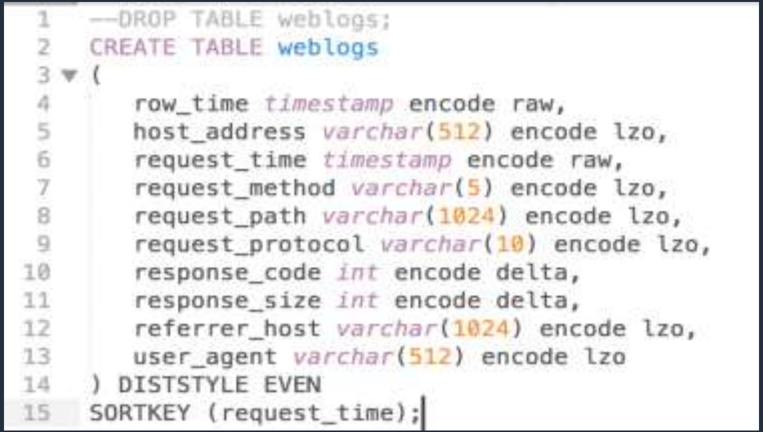


- Or, use any JDBC/ODBC/libpq client
  - Aginity Workbench for Amazon Redshift
  - SQL Workbench/J
  - DBeaver
  - Datagrip
- If you use the above SQL clients, the username/password is in AWS CloudFormation
- Select the big data CFN template and go to the Outputs tab
- You will find the Amazon Redshift cluster end point, username, and password

© 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Activity 3B: Create table in Amazon Redshift

- Make sure you are in the logs database in pgweb
- Create table weblogs to capture incoming data from Kinesis Data Firehose delivery stream
- Use RedshiftSQL.txt from Workshop Scripts for the CREATE TABLE statement
- Paste in the CREATE TABLE SQL statement and click “Run Query” to create the Redshift table

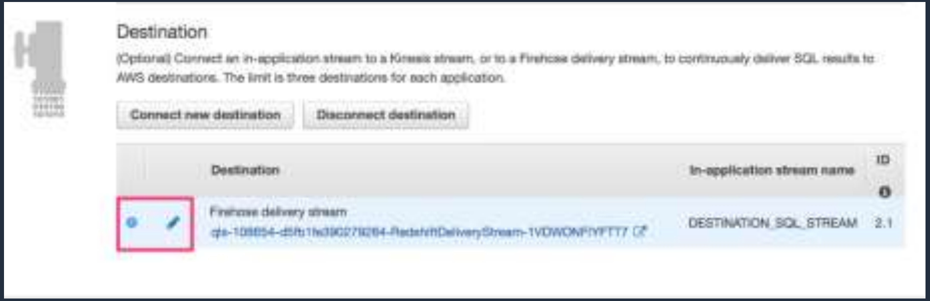


```
--DROP TABLE weblogs;
CREATE TABLE weblogs
(
    row_time timestamp encode raw,
    host_address varchar(512) encode lzo,
    request_time timestamp encode raw,
    request_method varchar(5) encode lzo,
    request_path varchar(1024) encode lzo,
    request_protocol varchar(10) encode lzo,
    response_code int encode delta,
    response_size int encode delta,
    referrer_host varchar(1024) encode lzo,
    user_agent varchar(512) encode lzo
) DISTSTYLE EVEN
SORTKEY (request_time);
```

© 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Activity 3C: Deliver data to Amazon Redshift using Data Firehose

- Update Kinesis Data Analytics application to send data to Kinesis Data Firehose delivery stream. Data Firehose delivers the streaming data to Amazon Redshift.
  1. Go to the **Kinesis Data Analytics console.** Go to Application Details.
  2. Choose the Amazon Redshift delivery stream as destination and click on the edit button (see the pencil icon in the figure below).



Destination

(Optional) Connect an in-application stream to a Kinesis stream, or to a Firehose delivery stream, to continuously deliver SQL results to AWS destinations. The limit is three destinations for each application.

		Destination	In-application stream name	ID
●	✎	Firehose delivery stream qls-108854-d5fb1fe3902792f4-RedshiftDeliveryStream-1VDWONFIYFTT7	DESTINATION_SQL_STREAM	2.1

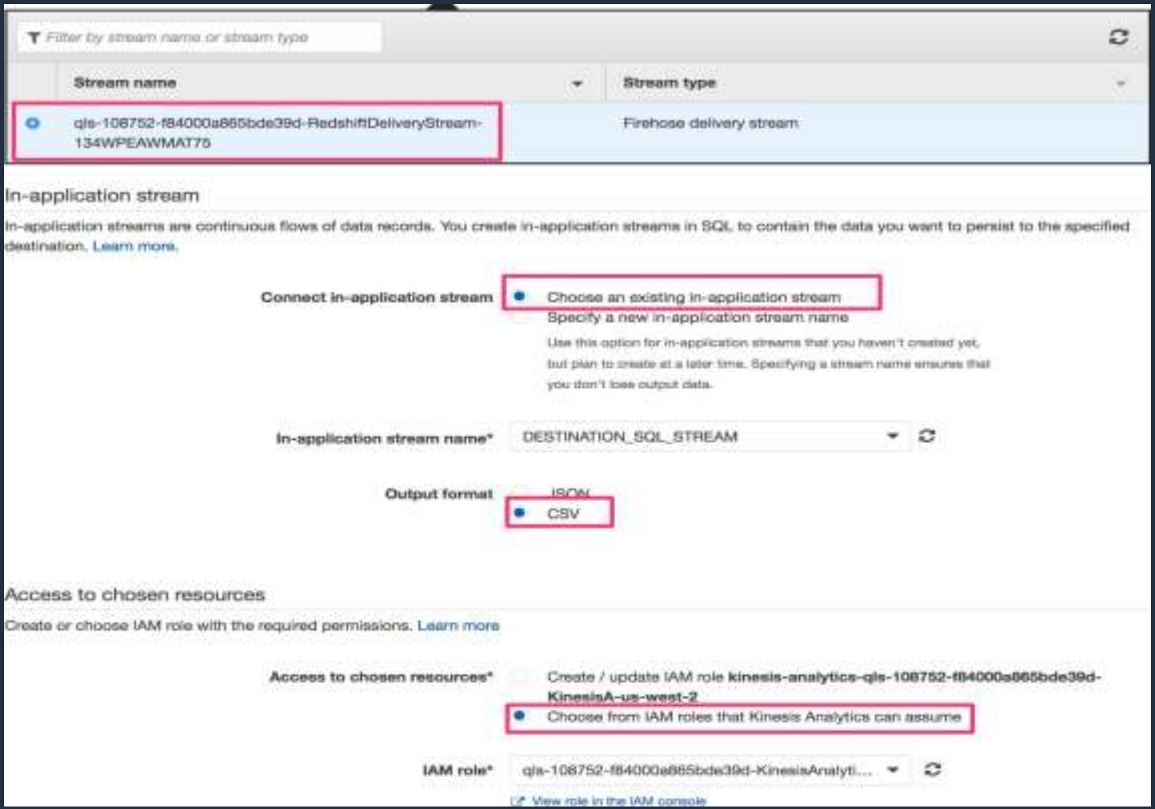
© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Activity 3C: Deliver data to Amazon Redshift using Data Firehose

- Validate your destination
  - Validate that the Kinesis Data Firehose stream is “<stackName>RedshiftStack-xxx-**RedshiftDeliveryStream**-xxxxxxxx”
  - Keep the default for “Choose an existing in-application stream”. DESTINATION_SQL_STREAM
  - Make sure CSV is the “Output format”
  - Validate that “Choose from IAM roles that Kinesis Data Analytics can assume”
  - Click “Save and continue”
- It will take about 1–2 minutes for everything to be updated and for data to start appearing in Amazon Redshift

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Activity 3C: Deliver data to Amazon Redshift using Data Firehose



	Stream name	Stream type
●	qls-108752-f84000a865bde39d-RedshiftDeliveryStream-134WPEAWMAT75	Firehose delivery stream

In-application stream

In-application streams are continuous flows of data records. You create in-application streams in SQL to contain the data you want to persist to the specified destination. Learn more.

Connect in-application stream
● Choose an existing in-application stream
○ Specify a new in-application stream name
Use this option for in-application streams that you haven't created yet, but plan to create at a later time. Specifying a stream name ensures that you don't lose output data.

In-application stream name* DESTINATION_SQL_STREAM

Output format
○ JSON
● CSV

Access to chosen resources

Create or choose IAM role with the required permissions. Learn more

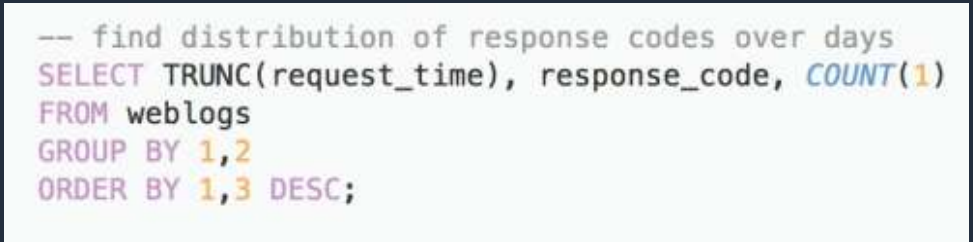
Access to chosen resources*
○ Create / update IAM role **kinesis-analytics-qls-108752-f84000a865bde39d-KinesisA-us-west-2**
● Choose from IAM roles that Kinesis Analytics can assume

IAM role* qls-108752-f84000a865bde39d-KinesisAnalyti...

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

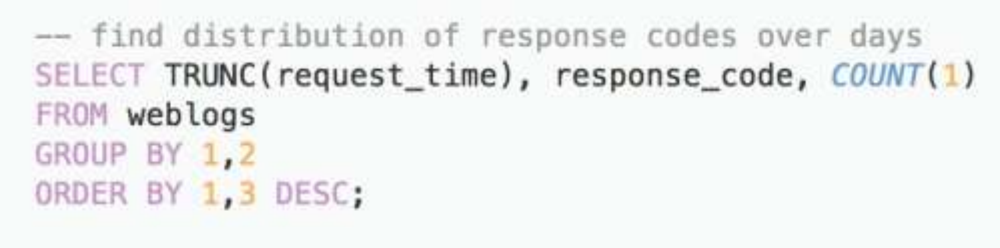
Review: Amazon Redshift test queries

- Find distribution of response codes over days (copy SQL from RedshiftSQL file



```
-- find distribution of response codes over days
SELECT TRUNC(request_time), response_code, COUNT(1)
FROM weblogs
GROUP BY 1,2
ORDER BY 1,3 DESC;
```

- Count the number of 404 response codes

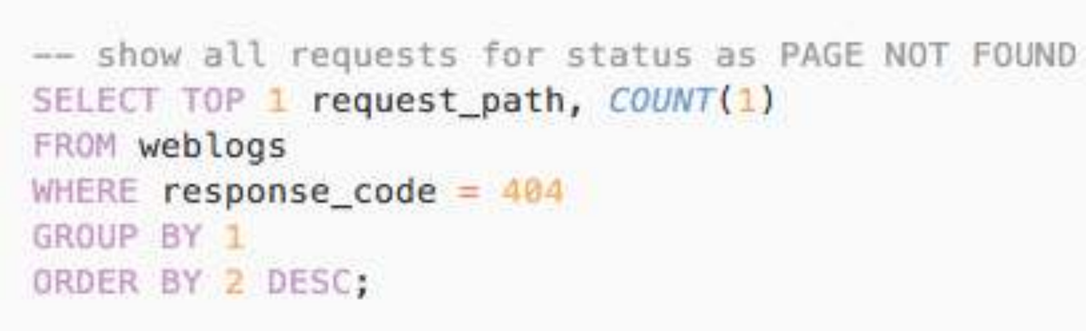


```
-- find distribution of response codes over days
SELECT TRUNC(request_time), response_code, COUNT(1)
FROM weblogs
GROUP BY 1,2
ORDER BY 1,3 DESC;
```

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Review: Amazon Redshift test queries

- Show all requests paths with status “PAGE NOT FOUND"



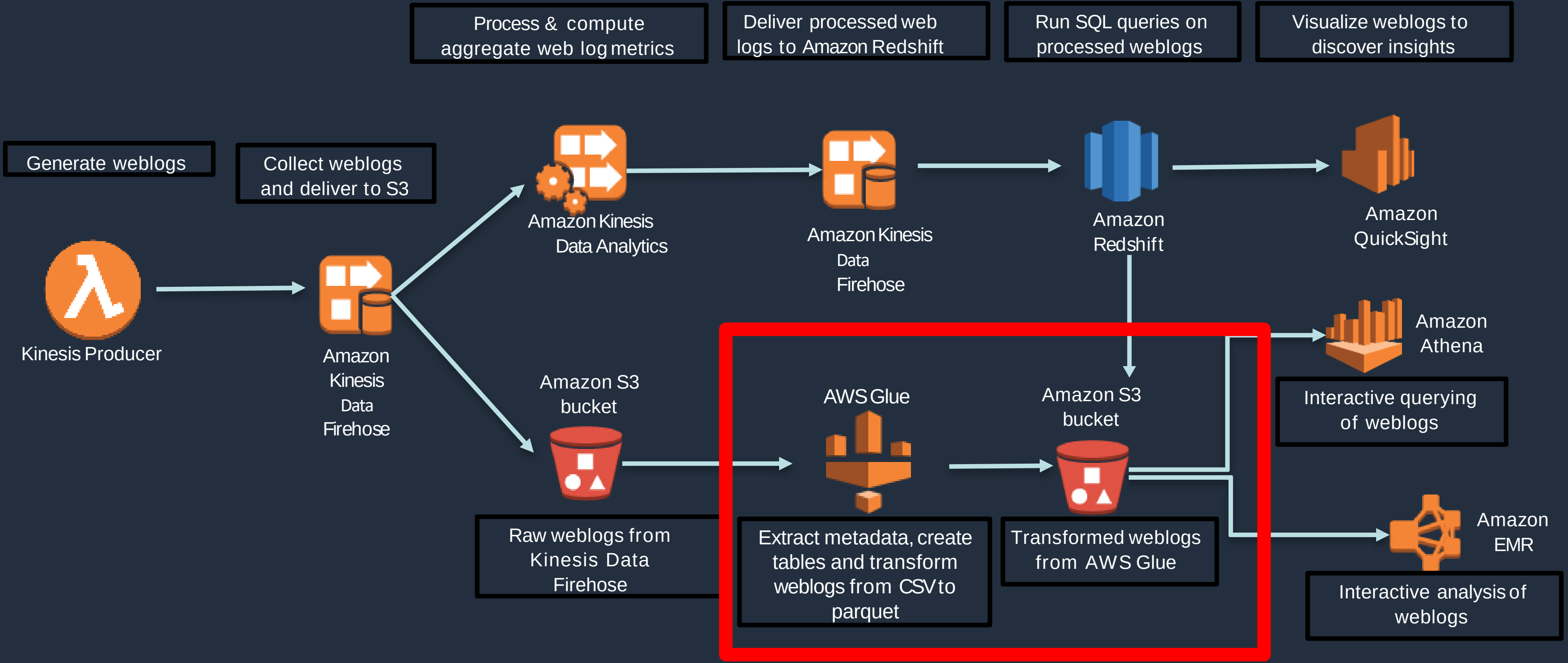
```
-- show all requests for status as PAGE NOT FOUND
SELECT TOP 1 request_path, COUNT(1)
FROM weblogs
WHERE response_code = 404
GROUP BY 1
ORDER BY 2 DESC;
```

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Activity 4
Transform weblogs to parquet format using AWS Glue

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Your application architecture


Process & compute aggregate web log metrics
Deliver processed web logs to Amazon Redshift
Run SQL queries on processed weblogs
Visualize weblogs to discover insights
Generate weblogs
Collect weblogs and deliver to S3
Amazon Kinesis Data Analytics
Amazon Kinesis Data Firehose
Amazon Redshift
Amazon QuickSight
Kinesis Producer
Amazon Kinesis Data Firehose
Amazon Athena
Amazon S3 bucket
AWS Glue
Amazon S3 bucket
Interactive querying of weblogs
Raw weblogs from Kinesis Data Firehose
Extract metadata, create tables and transform weblogs from CSV to parquet
Transformed weblogs from AWS Glue
Amazon EMR
Interactive analysis of weblogs

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Activity: Catalog and perform ETL on weblogs

Time: 30 minutes

We are going to:

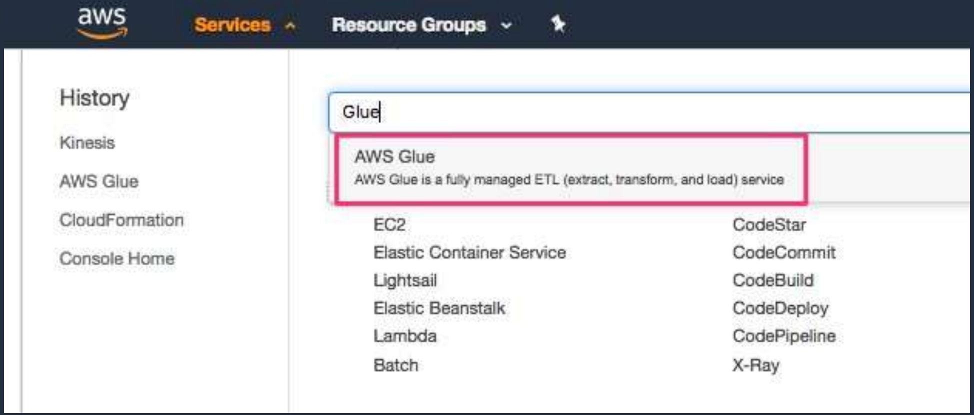
A. Discover and catalog the weblog data deposited into the S3 bucket using AWS Glue crawler

B. Transform weblogs to parquet format using the AWS Glue ETL job authoring tool

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Activity 4A: Discover dataset with AWS Glue

- We use AWS GLUE’s crawler to extract data and metadata. From the AWS Management Console, select AWS Glue. Click on “**Get Started”** on the next screen.



Glue

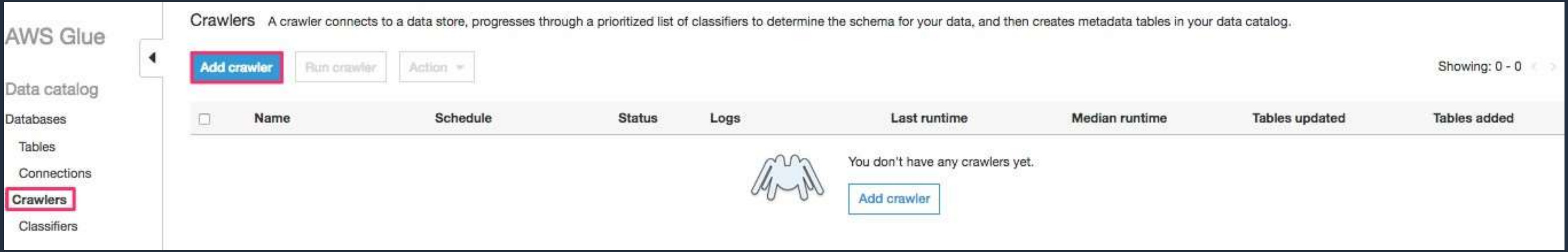
AWS Glue

AWS Glue is a fully managed ETL (extract, transform, and load) service

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Activity 4A: Add crawler using AWS Glue

- Select "**Crawlers"** section on the left and click on "**Add crawler"**



Crawlers A crawler connects to a data store, progresses through a prioritized list of classifiers to determine the schema for your data, and then creates metadata tables in your data catalog.

□	Name	Schedule	Status	Logs	Last runtime	Median runtime	Tables updated	Tables added

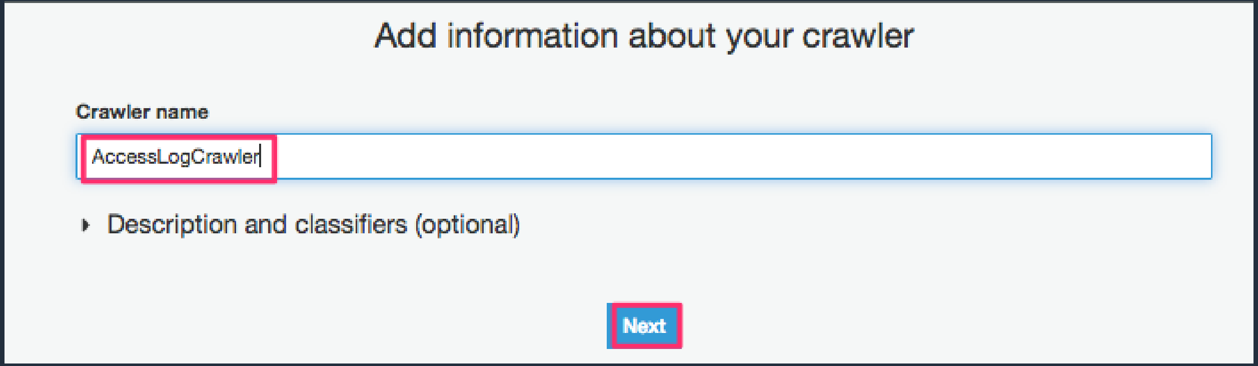
You don't have any crawlers yet.

Add crawler

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Activity 4A: ETL with AWS Glue

- Specify a name for the crawler. Click "**Next**"



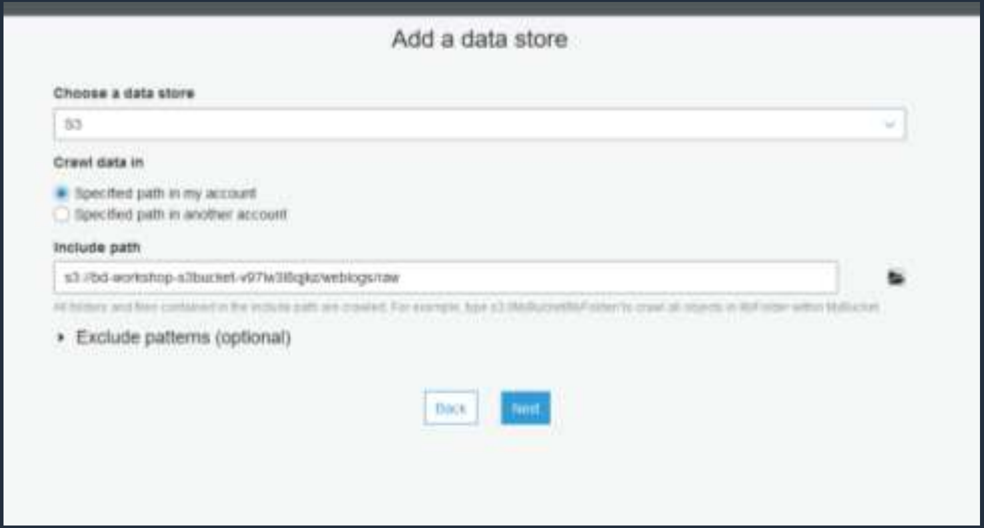
Add information about your crawler

Crawler name: AccessLogCrawler

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Activity 4A: ETL with AWS Glue

- Provide S3 path location where the raw weblogs were placed (navigate to S3 path: s3://<S3 bucket from template>/weblogs/raw)
- Click ”**Next**"



Add a data store

Choose a data store

S3

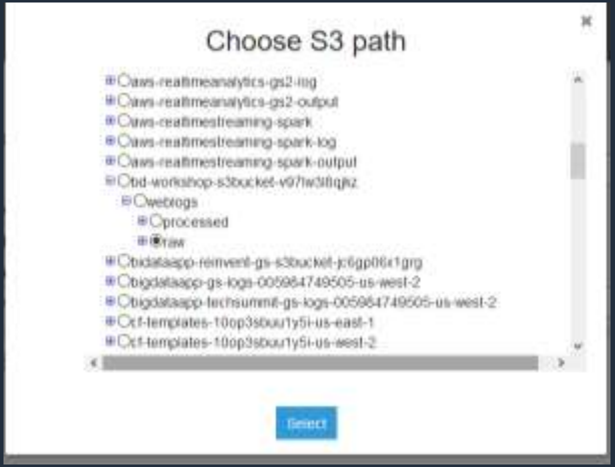
Crawl data in

● Specified path in my account
○ Specified path in another account

Include path

s3://bd-workshop-s3bucket-v97lw3l8qjkz/weblogs/raw

All folders and files contained in the include path are crawled. For example, type s3://MyBucket/MyFolder/ to crawl all objects in MyFolder within MyBucket.



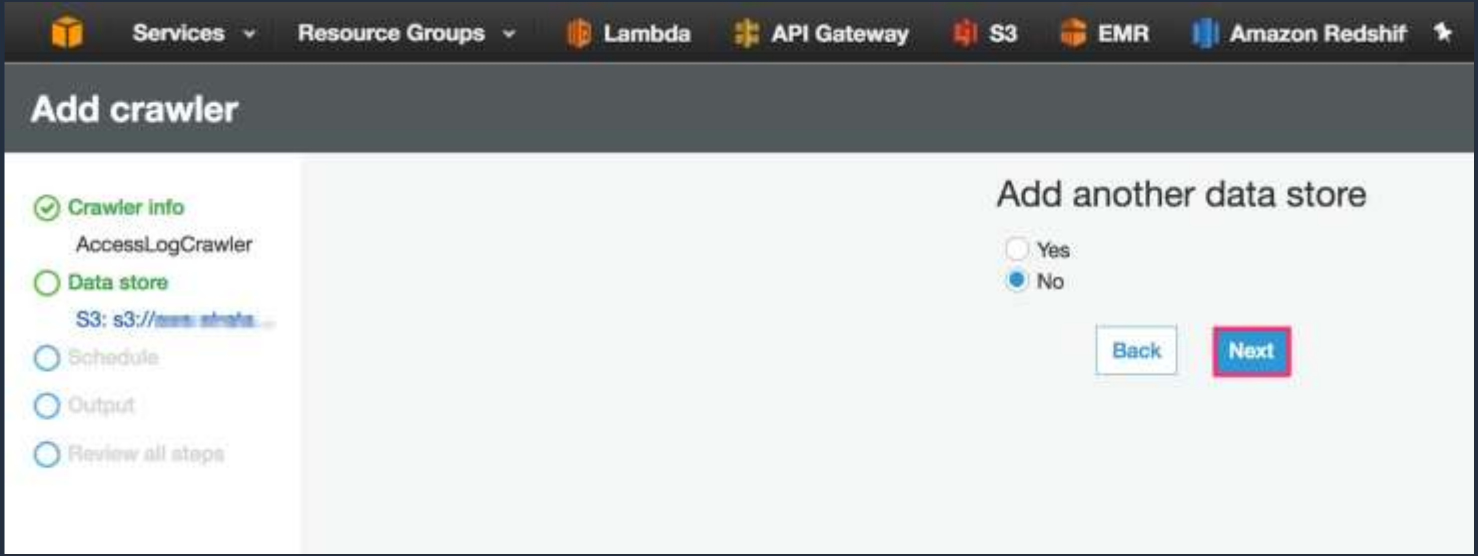
Choose S3 path

```
⊞○aws-realtimeanalytics-gs2-log
⊞○aws-realtimeanalytics-gs2-output
⊞○aws-realtimestreaming-spark
⊞○aws-realtimestreaming-spark-log
⊞○aws-realtimestreaming-spark-output
⊟○bd-workshop-s3bucket-v97lw3l8qjkz
  ⊟○weblogs
    ⊞○processed
    ⊞●raw
⊞○bidataapp-reinvent-gs-s3bucket-jc6gp06x1grg
⊞○bigdataapp-gs-logs-005984749505-us-west-2
⊞○bigdataapp-techsummit-gs-logs-005984749505-us-west-2
⊞○cf-templates-10op3sbuu1y5i-us-east-1
⊞○cf-templates-10op3sbuu1y5i-us-west-2
```

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Activity 4A: ETL with AWS Glue

- Click "**Next**" on the next screen to not add another data store



Add crawler

AccessLogCrawler

S3: s3://[redacted]

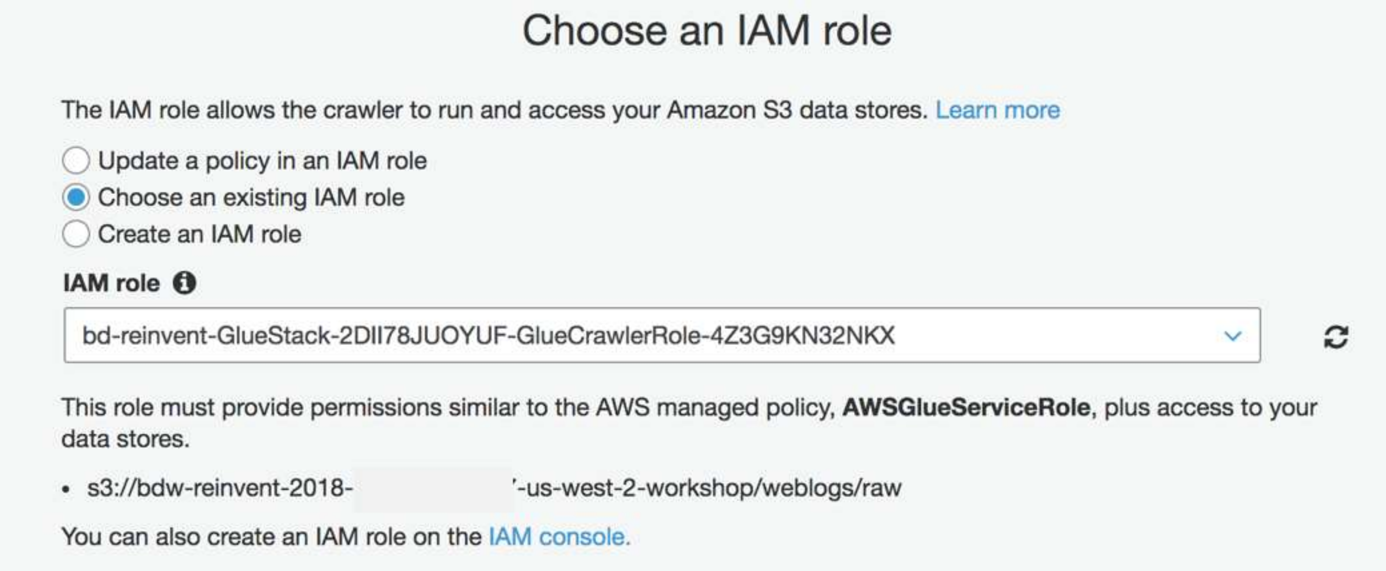
Add another data store

○ Yes
● No

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Activity 4A: ETL with AWS Glue

- In the IAM role section, select “Choose an existing IAM role”
- Select role <StackName>-GlueStack-xxx-GlueCrawlerRole-xxxxxxx as the IAM role and click “Next”



Choose an IAM role

The IAM role allows the crawler to run and access your Amazon S3 data stores. Learn more

○ Update a policy in an IAM role
● Choose an existing IAM role
○ Create an IAM role

IAM role

bd-reinvent-GlueStack-2DII78JUOYUF-GlueCrawlerRole-4Z3G9KN32NKX

This role must provide permissions similar to the AWS managed policy, **AWSGlueServiceRole**, plus access to your data stores.

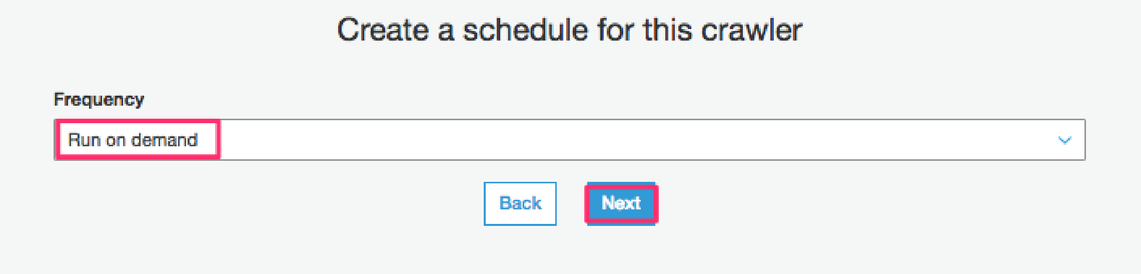
- s3://bdw-reinvent-2018-[redacted]-us-west-2-workshop/weblogs/raw

You can also create an IAM role on the IAM console.

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Activity 4A: Add crawler with AWS Glue

- Choose "**Run on demand**" to run the crawler now, and click **"Next"**



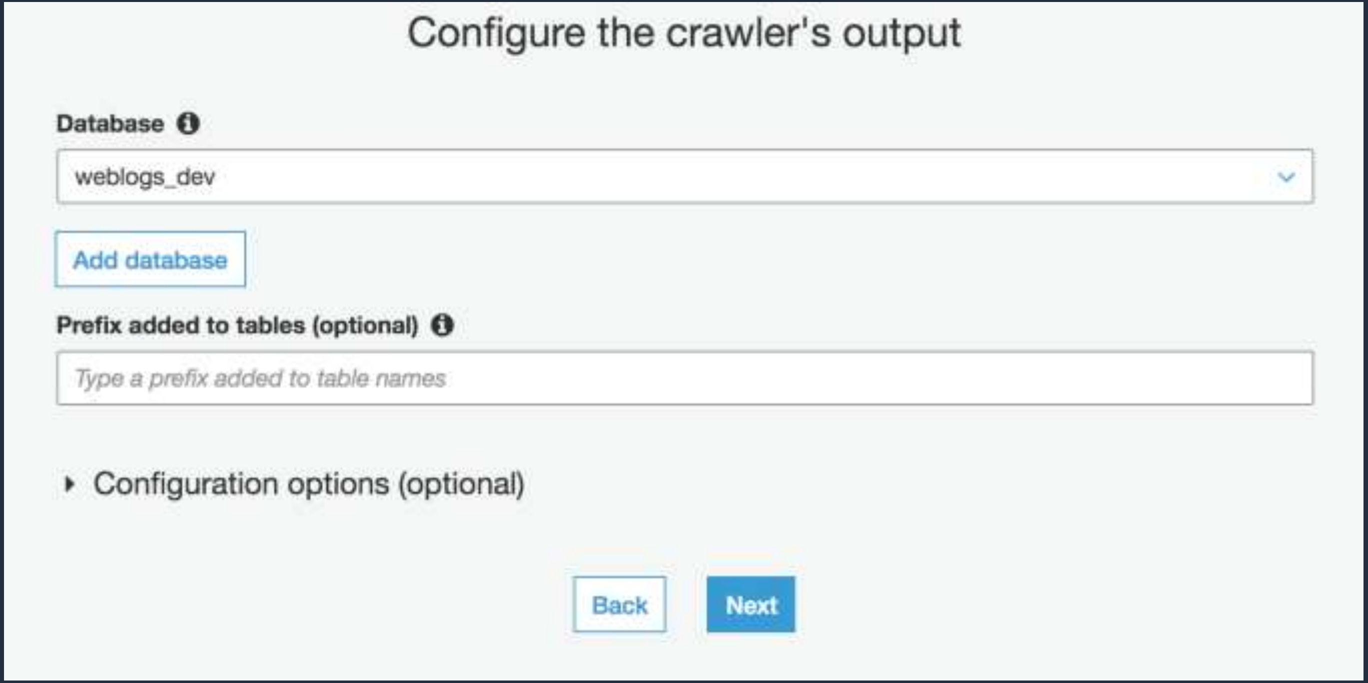
Create a schedule for this crawler

Frequency: Run on demand

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Activity 4A: Add crawler with AWS Glue

- On the next screen, drop down Database and select “weblogs_dev”
- Click “Next”



Configure the crawler's output

Database

weblogs_dev

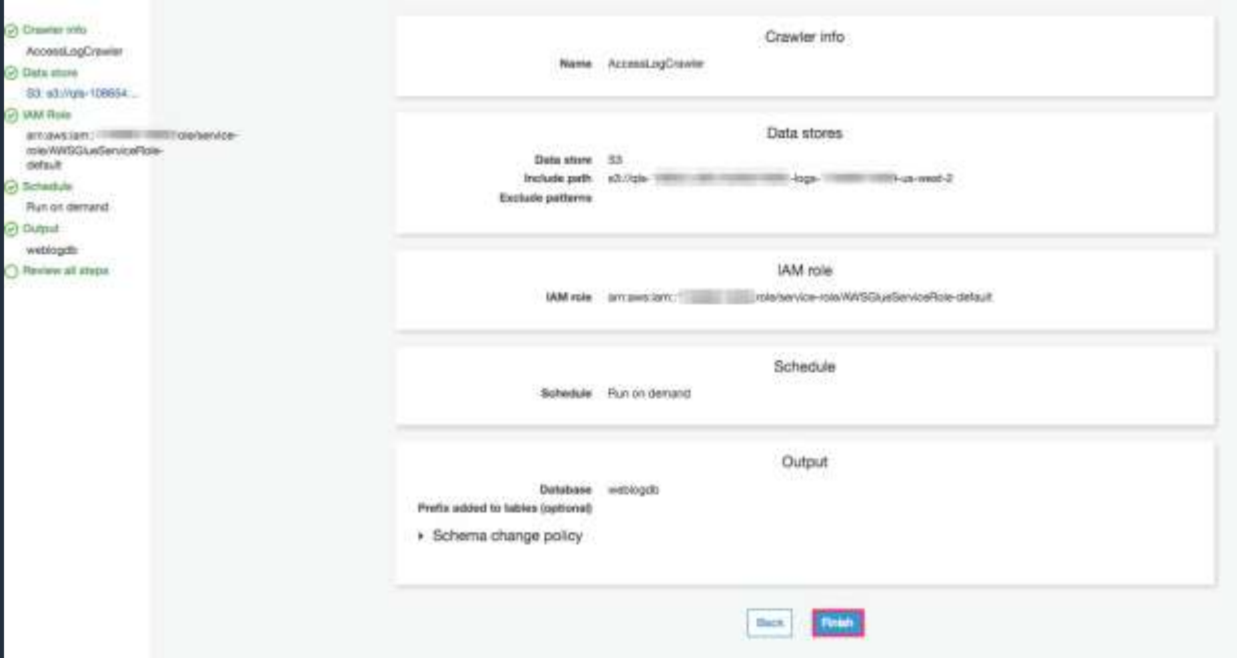
Prefix added to tables (optional)

Type a prefix added to table names

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Activity 4A: Add crawler with AWS Glue

- Review and click "**Finish**" to create a crawler



Crawler info
AccessLogCrawler
Data store
S3: s3://qls-108654...
IAM Role
arn:aws:iam::[redacted]ole/service-role/AWSGlueServiceRole-default
Schedule
Run on demand
Output
weblogdb
Review all steps

Crawler info

Name: AccessLogCrawler

Data stores

Data store: S3
Include path: s3://qls-[redacted]-logs-[redacted]-us-west-2
Exclude patterns:

IAM role

IAM role: arn:aws:iam::[redacted]role/service-role/AWSGlueServiceRole-default

Schedule

Schedule: Run on demand

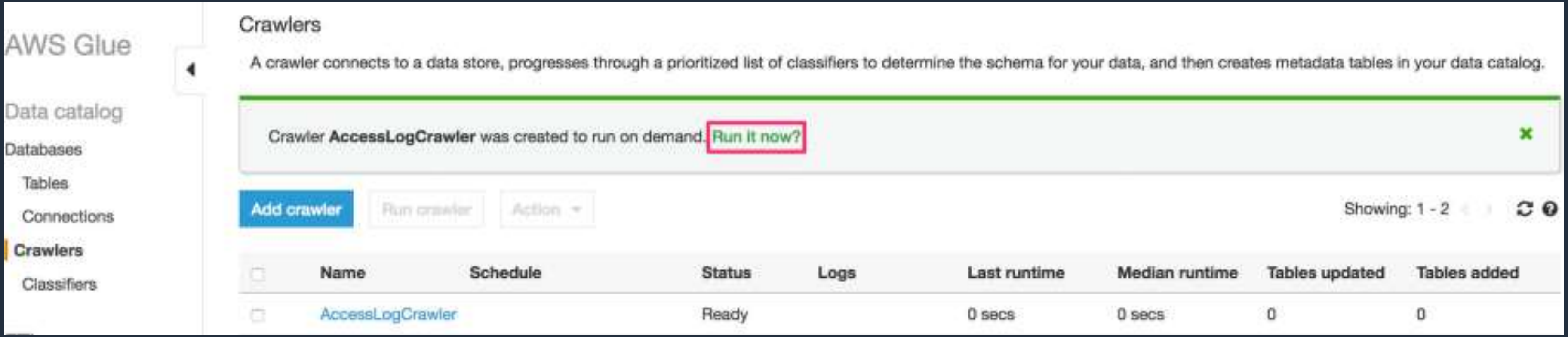
Output

Database: weblogdb
Prefix added to tables (optional):

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Activity 4A: Add crawler with AWS Glue

- Click on "**Run it now?**" link to run the crawler



Crawlers

A crawler connects to a data store, progresses through a prioritized list of classifiers to determine the schema for your data, and then creates metadata tables in your data catalog.

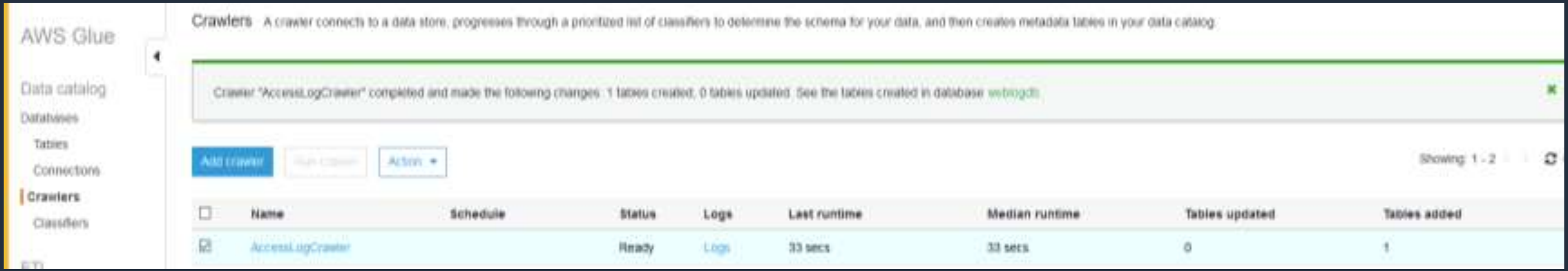
Crawler **AccessLogCrawler** was created to run on demand. Run it now?

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

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Activity 4A: Add crawler with AWS Glue

- Crawler shows a **Ready** status when it is finished running



Crawlers A crawler connects to a data store, progresses through a prioritized list of classifiers to determine the schema for your data, and then creates metadata tables in your data catalog.

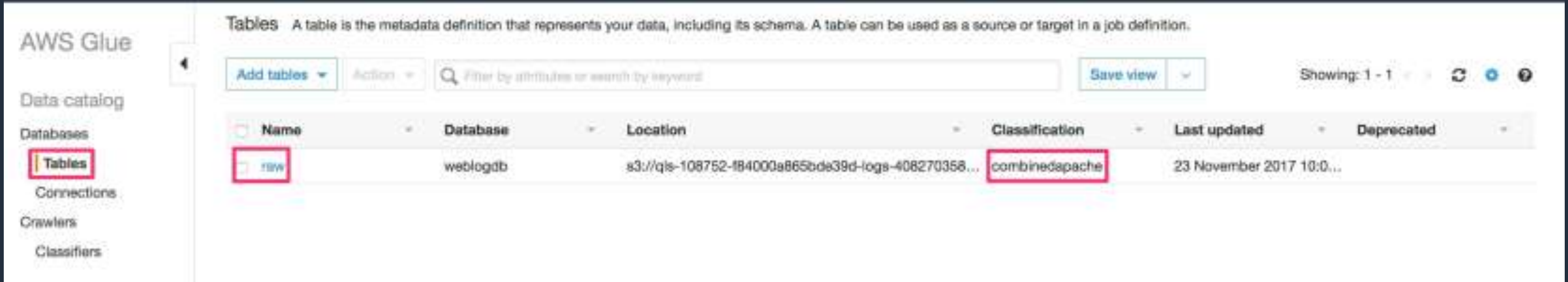
Crawler "AccessLogCrawler" completed and made the following changes: 1 tables created, 0 tables updated. See the tables created in database weblogdb

☐	Name	Schedule	Status	Logs	Last runtime	Median runtime	Tables updated	Tables added
☑	AccessLogCrawler		Ready	Logs	33 secs	33 secs	0	1

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Activity 4A: Table creation in AWS Glue

- Observe that the crawler has created a table for your dataset
- The crawler automatically classified the dataset as **combinedapache** log format
- Click the table to take a look at the properties



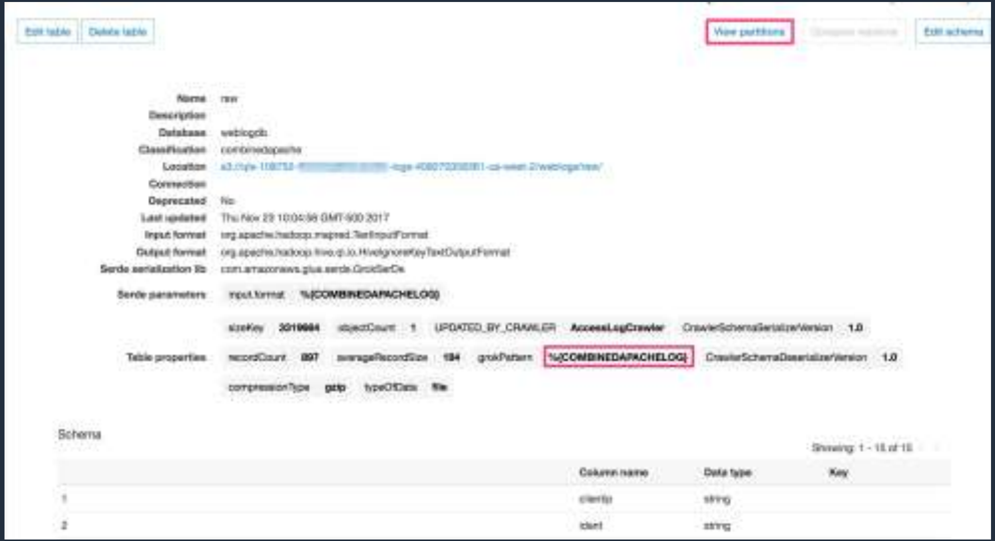
Tables A table is the metadata definition that represents your data, including its schema. A table can be used as a source or target in a job definition.

Name	Database	Location	Classification	Last updated	Deprecated
raw	weblogdb	s3://qls-108752-f84000a865bde39d-logs-408270358...	combinedapache	23 November 2017 10:0...	

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Activity 4A: Table creation in AWS Glue

- AWS Glue used the **GrokSerDe** (Serializer/Deserializer) to correctly interpret the weblogs
- You can click on the "**View partitions**" link to look at the partitions in the dataset



Name	raw
Description	
Database	weblogdb
Classification	combinedapache
Location	s3://ylv-108752-[redacted]-logs-408072050881-us-west-2/weblogs/raw/
Connection	
Deprecated	No
Last updated	Thu Nov 23 10:04:38 GMT-800 2017
Input format	org.apache.hadoop.mapred.TextInputFormat
Output format	org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat
Serde serialization lib	com.amazonaws.glue.serde.GrokSerDe
Serde parameters	input.format **%{COMBINEDAPACHELOG}**
	sizeKey **3019684** objectCount **1** UPDATED_BY_CRAWLER **AccessLogCrawler** CrawlerSchemaSerializerVersion **1.0**
Table properties	recordCount **897** averageRecordSize **184** grokPattern **%{COMBINEDAPACHELOG}** CrawlerSchemaDeserializerVersion **1.0**
	compressionType **gzip** typeOfData **file**

Schema

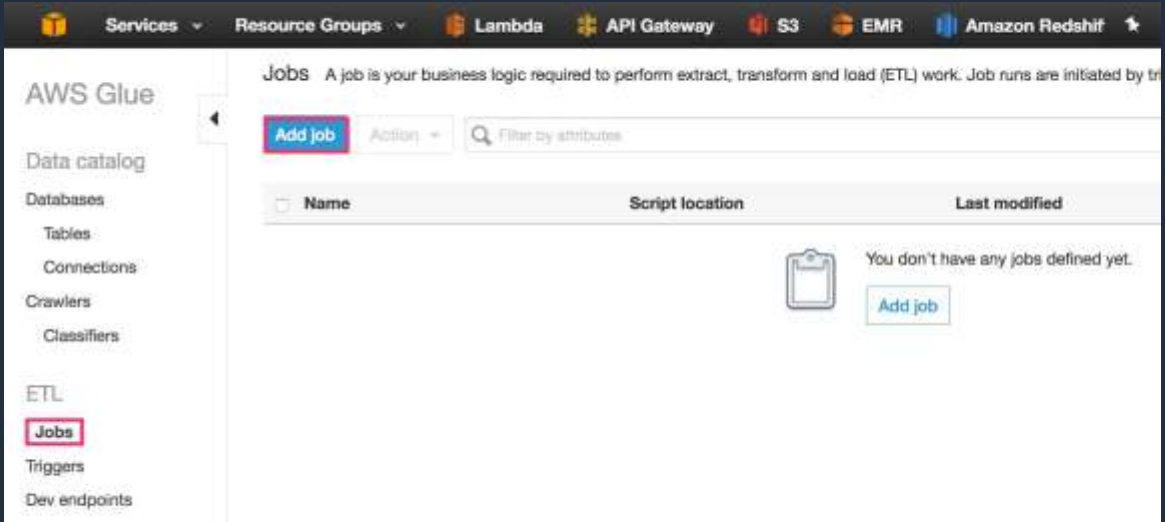
Showing: 1 - 15 of 15

	Column name	Data type	Key
1	clientip	string	
2	ident	string	

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Activity 4A: Create ETL job in AWS Glue

- With the dataset cataloged and table created, we are now ready to convert the weblogs–based Apache combined log format to a more optimal parquet format for querying.
- Click on "**Add job**" to begin creating the ETL job



Jobs A job is your business logic required to perform extract, transform and load (ETL) work. Job runs are initiated by tr

Name	Script location	Last modified

You don't have any jobs defined yet.

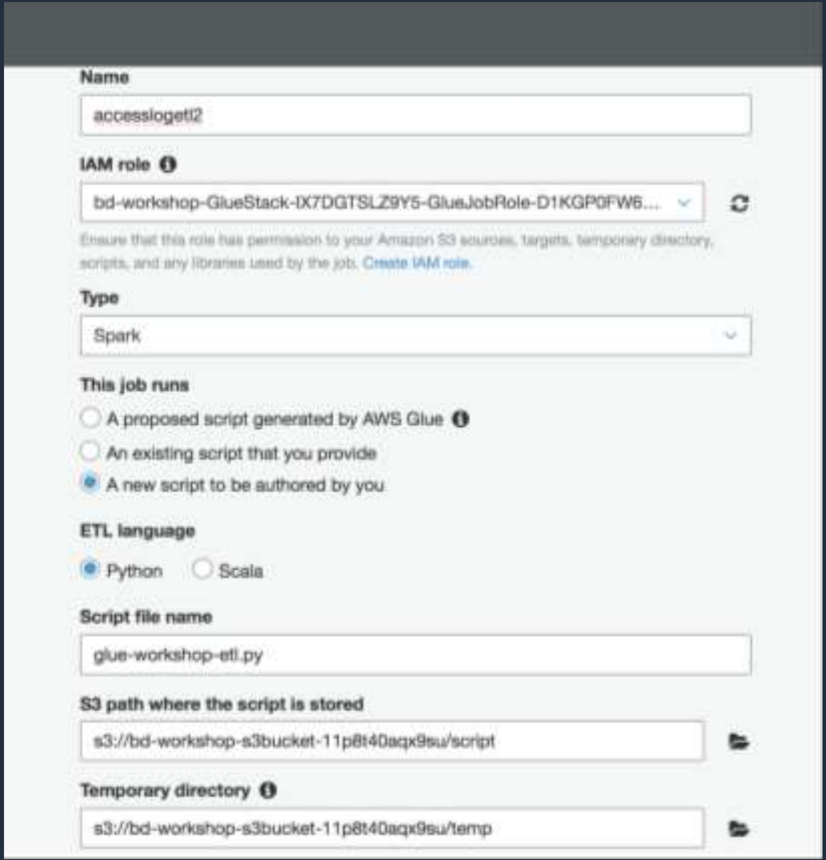
© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Activity 4B: ETL job in AWS Glue

- Job name: **accesslogetl2**
- Select the IAM role <StackName>-GlueStack-xxx-GlueJobRole-xxxxxxx
- Select Type as **Spark**
- Select “a new script to be authored by you”
- ETL language as **Python**

- Script file name: **glue-workshop-etl.py**
- For the S3 path where the script will be stored, use path **s3://<S3 bucket from template>/script**
- For the Temporary Directory, use path **s3://<S3 bucket from template>/temp**

- **DO NOT CLICK NEXT JUST YET**



Name

accesslogetl2

IAM role

bd-workshop-GlueStack-IX7DGTSLZ9Y5-GlueJobRole-D1KGP0FW6...

Ensure that this role has permission to your Amazon S3 sources, targets, temporary directory, scripts, and any libraries used by the job. Create IAM role.

Type

Spark

This job runs

○ A proposed script generated by AWS Glue

○ An existing script that you provide

● A new script to be authored by you

ETL language

● Python ○ Scala

Script file name

glue-workshop-etl.py

S3 path where the script is stored

s3://bd-workshop-s3bucket-11p8t40aqx9su/script

Temporary directory

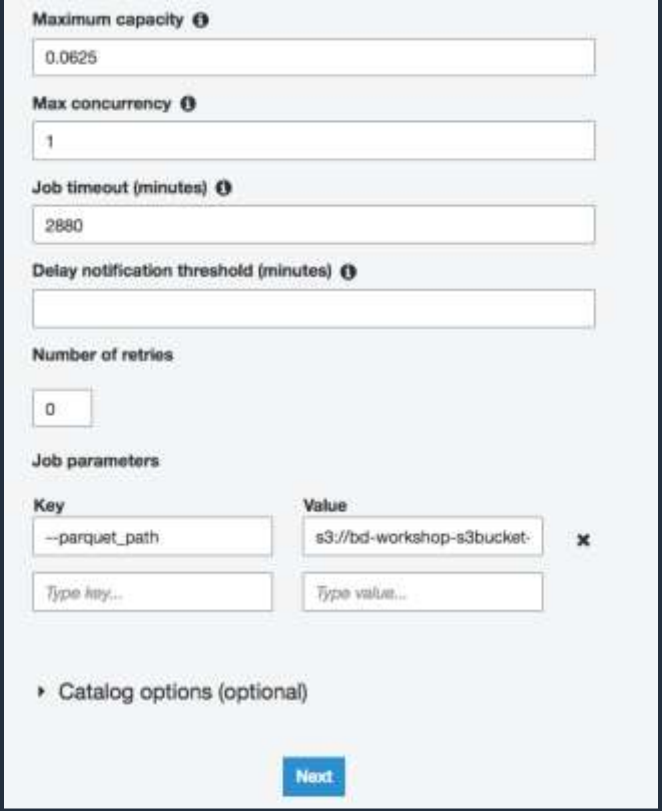
s3://bd-workshop-s3bucket-11p8t40aqx9su/temp

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Activity 4B: ETL job in AWS Glue

- Expand **Security configuration, script libraries, and job parameters** section, and set the Maximum capacity to 10

- Let's pass a job parameter to send the S3 path where parquet files will be deposited.
- Specify the following values for Key and Value
- **Key: --parquet_path** (notice the 2 hyphens at the beginning and underscore between "parquet" and "path")
- **Value: s3://<S3 bucket from template>/weblogs/parquet**
- **Note:** Value is the S3 path we stored from the previous slide



Maximum capacity: 0.0625

Max concurrency: 1

Job timeout (minutes): 2880

Delay notification threshold (minutes):

Number of retries: 0

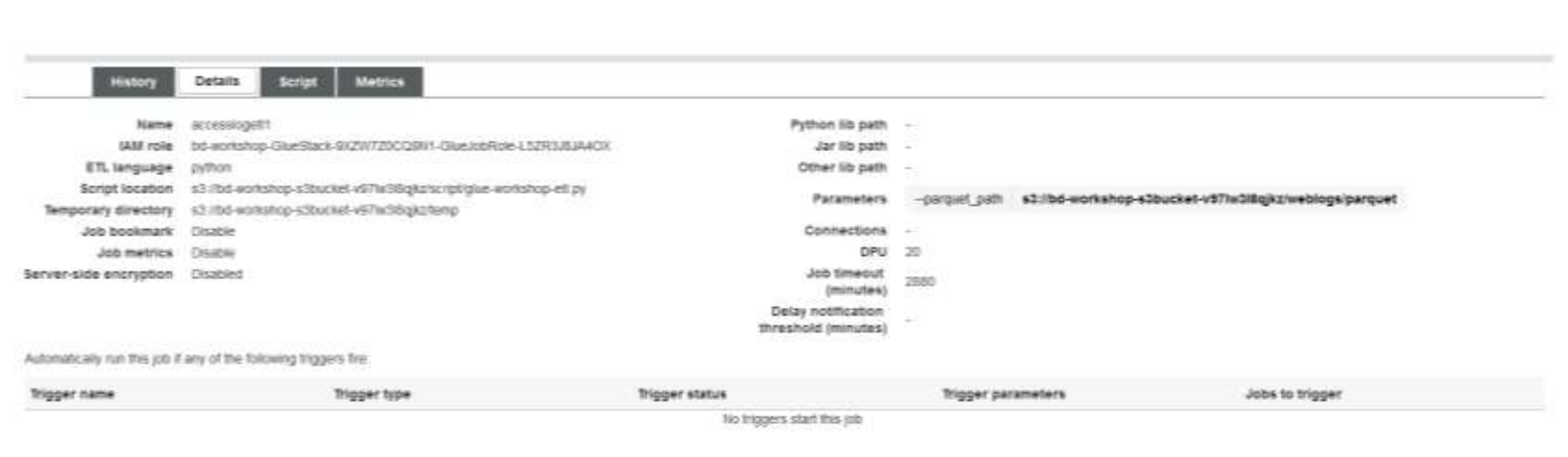
Job parameters

Key	Value
--parquet_path	s3://bd-workshop-s3bucket-
Type key...	Type value...

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Activity 4B: ETL job in AWS Glue

- Click "**Next**" in the following screen
- Review and click "**Save Job and Edit Script**" to create the job



Name accesslogetl1
IAM role bd-workshop-GlueStack-9XZW7Z0CQ9I1-GlueJobRole-L5ZR3J8JA4OX
ETL language python
Script location s3://bd-workshop-s3bucket-v97lw38qjkz/script/glue-workshop-etl.py
Temporary directory s3://bd-workshop-s3bucket-v97lw38qjkz/temp
Job bookmark Disable
Job metrics Disable
Server-side encryption Disabled

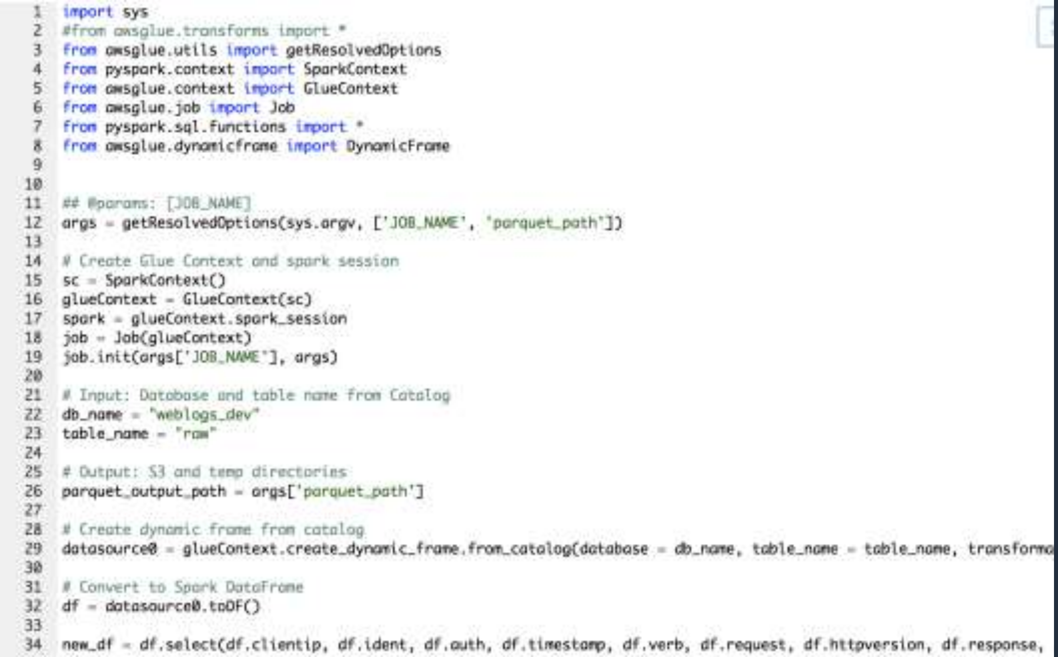
Python lib path -
Jar lib path -
Other lib path -
Parameters --parquet_path s3://bd-workshop-s3bucket-v97lw38qjkz/weblogs/parquet
Connections -
DPU 20
Job timeout (minutes) 2880
Delay notification threshold (minutes) -

Automatically run this job if any of the following triggers fire:

Trigger name	Trigger type	Trigger status	Trigger parameters	Jobs to trigger
		No triggers start this job		

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Activity 4B: ETL job in AWS Glue



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


## @params: [JOB_NAME]
args = getResolvedOptions(sys.argv, ['JOB_NAME', 'parquet_path'])

# Create Glue Context and spark session
sc = SparkContext()
glueContext = GlueContext(sc)
spark = glueContext.spark_session
job = Job(glueContext)
job.init(args['JOB_NAME'], args)

# Input: Database and table name from Catalog
db_name = "weblogs_dev"
table_name = "raw"

# Output: S3 and temp directories
parquet_output_path = args['parquet_path']

# Create dynamic frame from catalog
datasource0 = glueContext.create_dynamic_frame.from_catalog(database = db_name, table_name = table_name, transforma

# Convert to Spark DataFrame
df = datasource0.toDF()

new_df = df.select(df.clientip, df.ident, df.auth, df.timestamp, df.verb, df.request, df.httpversion, df.response,
```

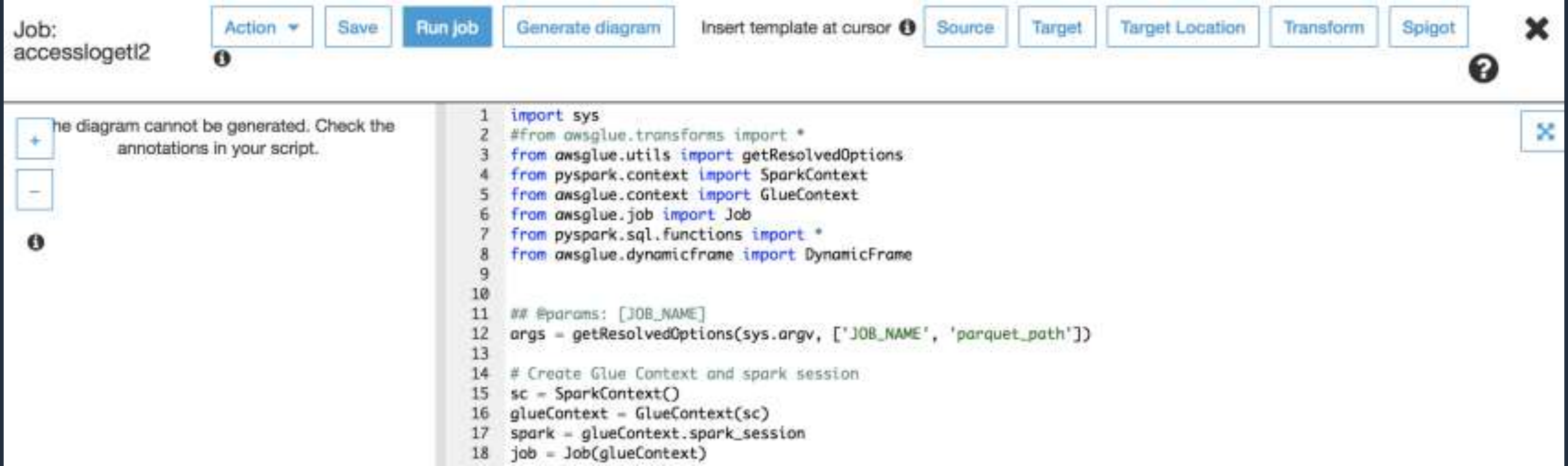
- Close script editor tips window (if it appears).
- In the AWS Glue script editor, copy the ETL code in glue-workshop-etl.py from BigDataWorkshop.zip and paste. Overwrite; don’t append.

- Ensure that the db_name and table_name statements reflect the database and table name created by the AWS Glue crawler.

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Activity 4B: ETL job in AWS Glue

- Click "**Save**" and then "**Run job**" button to execute your ETL



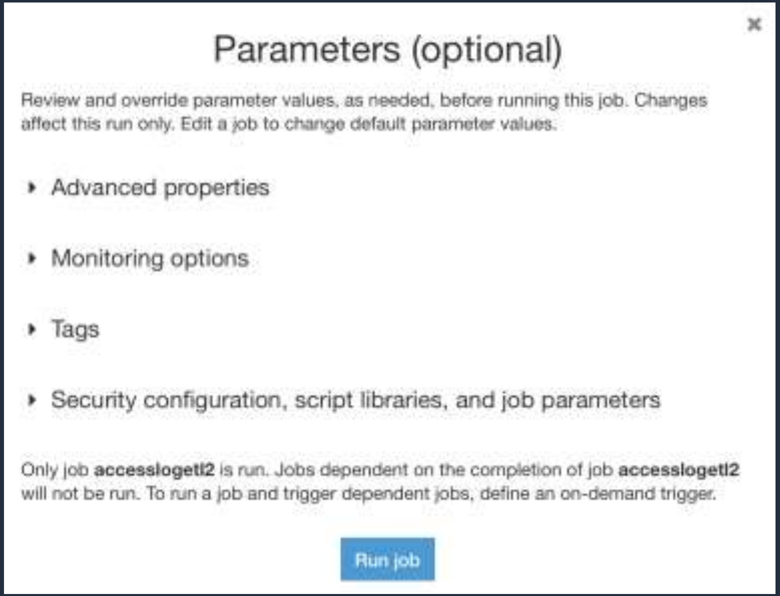
he diagram cannot be generated. Check the annotations in your script.

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


## @params: [JOB_NAME]
args = getResolvedOptions(sys.argv, ['JOB_NAME', 'parquet_path'])

# Create Glue Context and spark session
sc = SparkContext()
glueContext = GlueContext(sc)
spark = glueContext.spark_session
job = Job(glueContext)
```

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Activity 4B: ETL job in AWS Glue

- Check that the --parquet_path is correctly set in the job parameters.
- Check that the **Python library path** set to your bucket directing you to the glue-workshop-etl.py script
    - s3:**//<S3 bucket from template>**/script/glue-workshop-etl.py
- Click "**Run job**" to continue. This might take a few minutes. When the job finishes, weblogs will be transformed to parquet format.
- Go to s3://bd-workshop-s3bucket-xxxxxxxxxx/weblogs/parquet and verify that you see parquet.snappy files in there**.**



Parameters (optional)

Review and override parameter values, as needed, before running this job. Changes affect this run only. Edit a job to change default parameter values.

▸ Advanced properties
▸ Monitoring options
▸ Tags
▸ Security configuration, script libraries, and job parameters
Run job

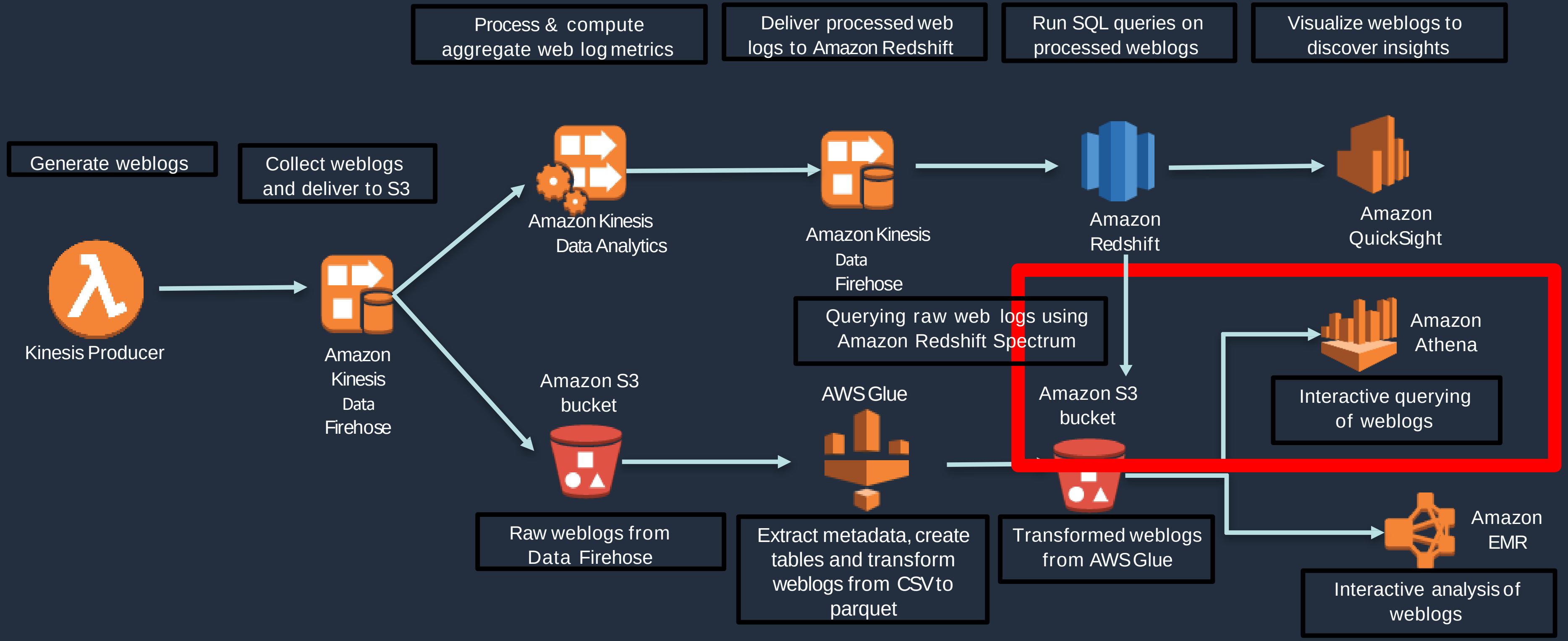
Only job **accesslogetl2** is run. Jobs dependent on the completion of job **accesslogetl2** will not be run. To run a job and trigger dependent jobs, define an on-demand trigger.

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Activity 5
Amazon Redshift Spectrum and interactive querying with Amazon Athena

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Your application architecture



Generate weblogs

Kinesis Producer

Collect weblogs and deliver to S3

Amazon Kinesis Data Firehose

Process & compute aggregate web log metrics

Amazon Kinesis Data Analytics

Deliver processed web logs to Amazon Redshift

Amazon Kinesis Data Firehose

Run SQL queries on processed weblogs

Amazon Redshift

Visualize weblogs to discover insights

Amazon QuickSight

Querying raw web logs using Amazon Redshift Spectrum

Amazon S3 bucket

Raw weblogs from Data Firehose

AWS Glue

Extract metadata, create tables and transform weblogs from CSV to parquet

Amazon S3 bucket

Transformed weblogs from AWS Glue

Amazon Athena

Interactive querying of weblogs

Amazon EMR

Interactive analysis of weblogs

© 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Activity: Querying data in Amazon S3 using Redshift Spectrum

Time: 30 minutes

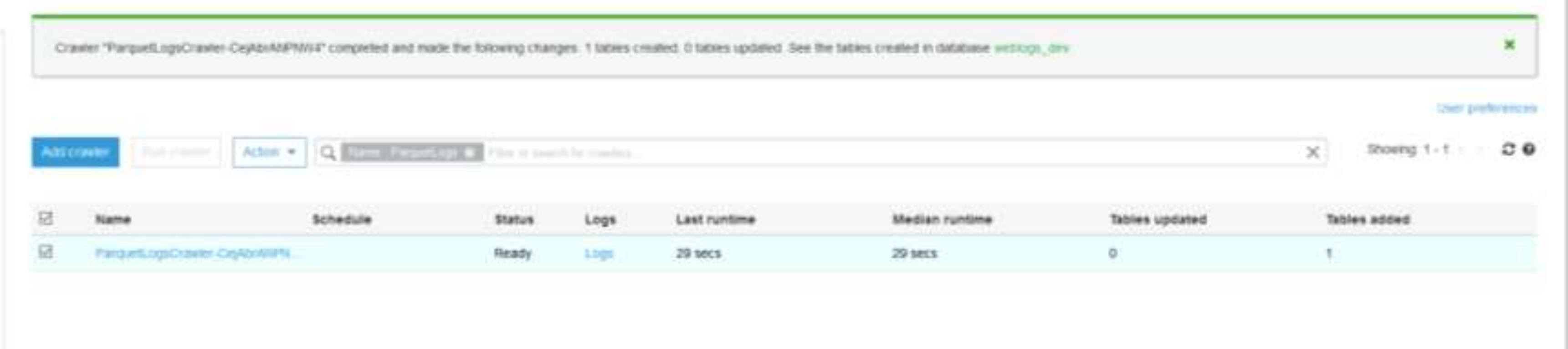
We are going to:

A. Create a table over the processed weblogs in Amazon S3 using an AWS Glue crawler. These are the parquet files created by AWS Glue ETL job in the previous section.
B. Run queries from Amazon Redshift on the parquet weblogs in Amazon S3 using Amazon Redshift Spectrum.
C. Run interactive queries from Athena on parquet weblogs in S3.

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Activity 5A: Set up AWS Glue crawler for processed parquet data

- Go to Crawlers in AWS Glue and search for ParquetLogs.
- Select the ParquetLogsCrawler-xxxxxxx. Go to details tab and check that it’s pointing to the right parquet location in Amazon S3.
- Run crawler. Should take about 30 secs for the crawler to finish running.



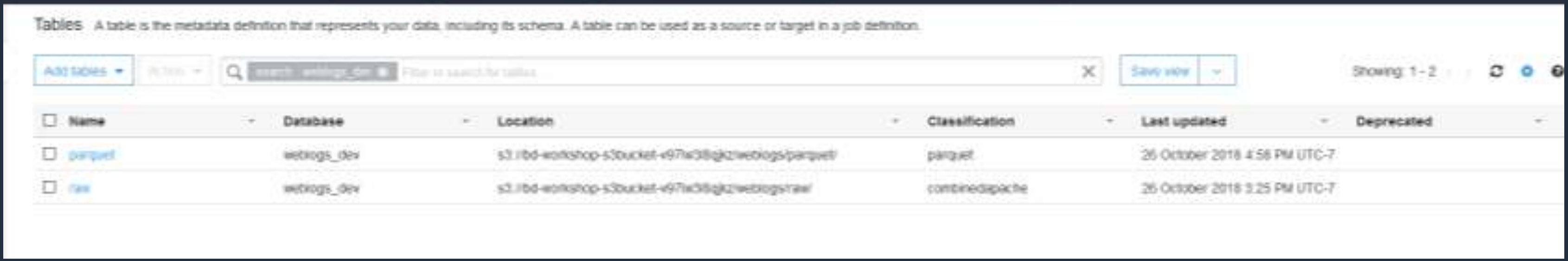
Crawler "ParquetLogsCrawler-CeyAbiAtiPNW4" completed and made the following changes: 1 tables created, 0 tables updated. See the tables created in database weblogs_dev

☑	Name	Schedule	Status	Logs	Last runtime	Median runtime	Tables updated	Tables added
☑	ParquetLogsCrawler-CeyAbiAtiPN...		Ready	Logs	29 secs	29 secs	0	1

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Activity 5A: Set up AWS Glue crawler for processed parquet data

- Navigate to Databases-Tables in AWS Glue. Search weblogs_<environment>.
- Make sure you see both the “raw” and the newly created “parquet” table.



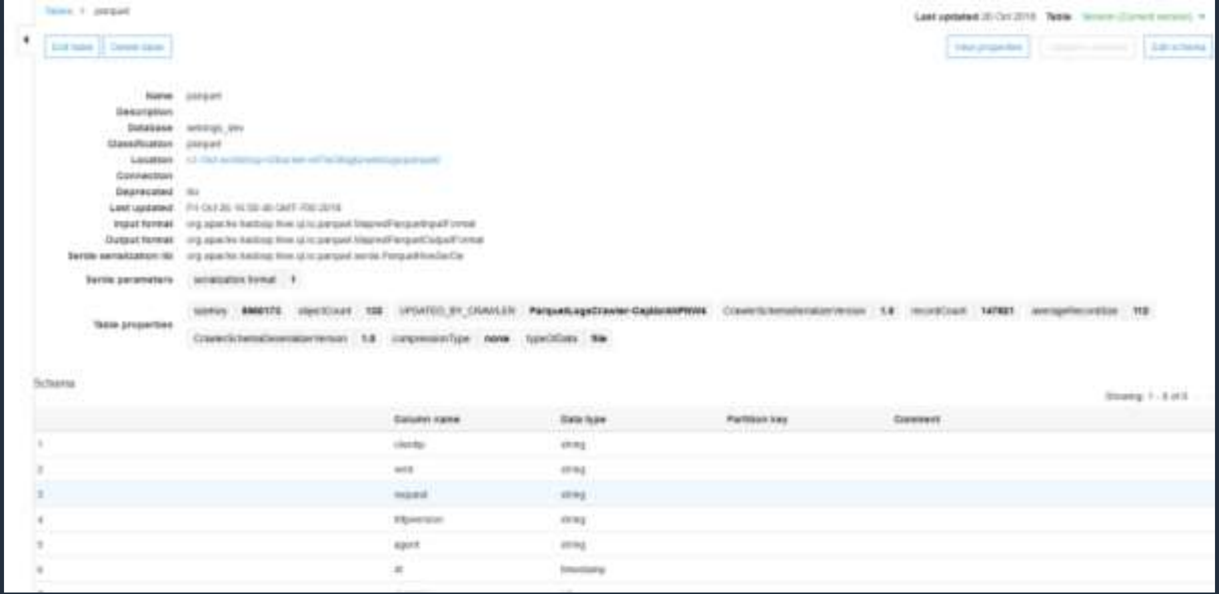
Tables A table is the metadata definition that represents your data, including its schema. A table can be used as a source or target in a job definition.

□ Name	Database	Location	Classification	Last updated	Deprecated
□ parquet	weblogs_dev	s3://bd-workshop-s3bucket-v97lw38qjkz/weblogs/parquet/	parquet	26 October 2018 4:58 PM UTC-7	
□ raw	weblogs_dev	s3://bd-workshop-s3bucket-v97lw38qjkz/weblogs/raw/	combinedapache	26 October 2018 3:25 PM UTC-7	

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Activity 5A: Set up AWS Glue crawler for parquet data

- Take a look at the schema of the parquet table
- Should have eight columns
- Classification should be parquet



Name	parquet
Description	
Database	weblogs_dev
Classification	parquet
Location	s3://[illegible]
Connection	
Deprecated	No
Last updated	Fri Oct 26 [illegible] GMT-700 2018
Input format	org.apache.hadoop.hive.ql.io.parquet.MapredParquetInputFormat
Output format	org.apache.hadoop.hive.ql.io.parquet.MapredParquetOutputFormat
Serde serialization lib	org.apache.hadoop.hive.ql.io.parquet.serde.ParquetHiveSerDe
Serde parameters	serialization.format 1
Table properties	sizeKey 8860172 objectCount 132 UPDATED_BY_CRAWLER ParquetLogsCrawler-[illegible] CrawlerSchemaSerializerVersion 1.0 recordCount 147821 averageRecordSize 112 CrawlerSchemaDeserializerVersion 1.0 compressionType none typeOfData file

Schema

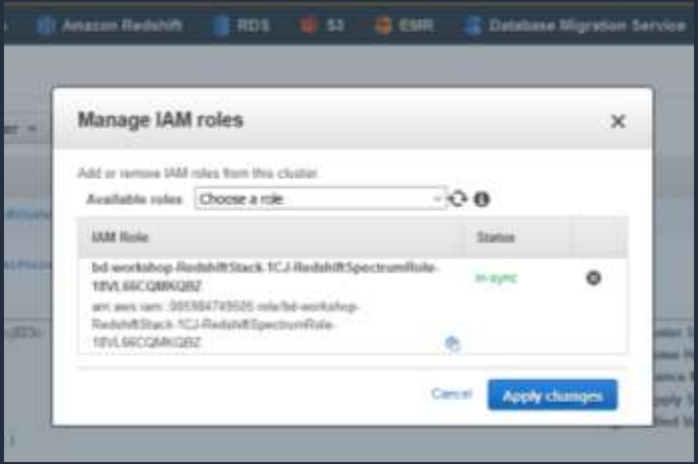
Showing: 1 - 8 of 8

	Column name	Data type	Partition key	Comment
1	clientip	string		
2	verb	string		
3	request	string		
4	httpversion	string		
5	agent	string		
6	dt	timestamp		

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Activity 5B: Query using Amazon Redshift Spectrum

- Navigate to Amazon Redshift-Clusters. Select the cluster with the <stack-name>
- Click on “Manage IAM roles”.
- Validate that IAM role shows bd-workshop-RedshiftStack-xxx-RedshiftSpectrumRole-xxxxxxxxxxx.



Manage IAM roles

Add or remove IAM roles from this cluster.

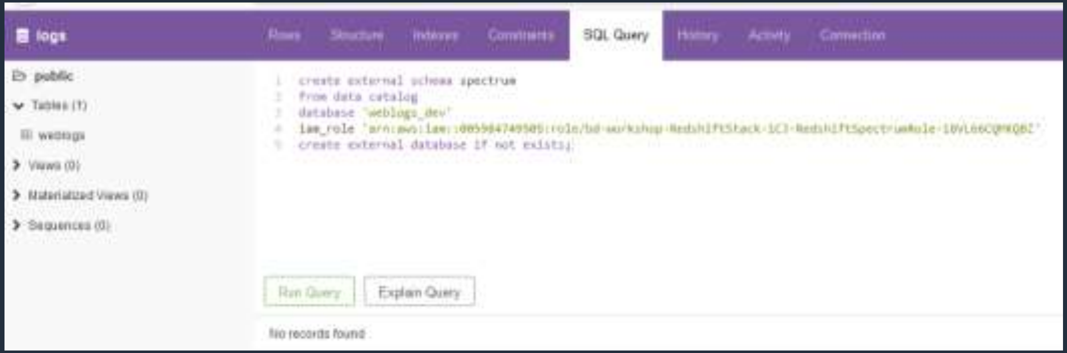
Available roles: Choose a role

IAM Role	Status
bd-workshop-RedshiftStack-1CJ-RedshiftSpectrumRole-18VL66CQMKQBZ arn:aws:iam::985384749505:role/bd-workshop-RedshiftStack-1CJ-RedshiftSpectrumRole-18VL66CQMKQBZ	in-sync

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Activity 5B: Query using Amazon Redshift Spectrum

- Navigate to pgWeb. If you need the URL, check public IP of the pgweb EC2.
- Get the RedshiftSpectrumSetup.sql from the BigDataWorkshop.zip.
  - Make sure you replace the role ARN with your SpectrumRole ARN.
  - Make sure you are pointing to the weblogs_dev database.
  - This will create an external schema in Amazon Redshift called “Spectrum”.



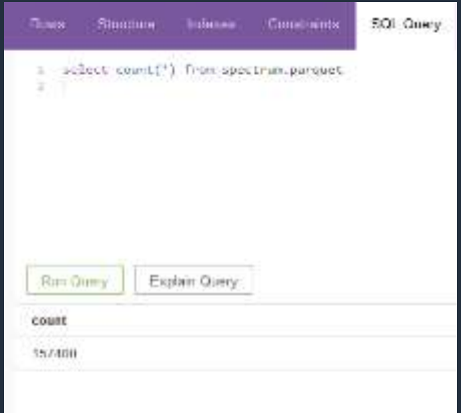
```
create external schema spectrum
from data catalog
database 'weblogs_dev'
iam_role 'arn:aws:iam::005984749505:role/bd-workshop-RedshiftStack-1C3-RedshiftSpectrumRole-10VL66CQMXQBZ'
create external database if not exists;
```

No records found

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

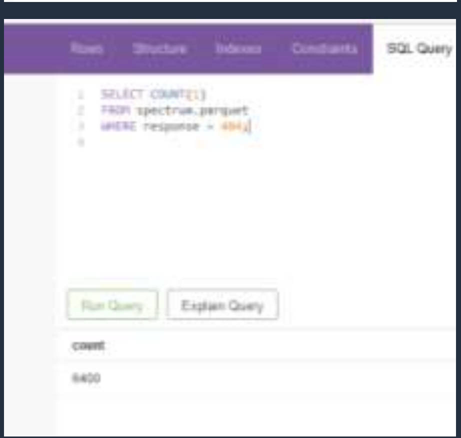
Activity 5B: Query using Amazon Redshift Spectrum

- Let’s run a couple of simple queries against Amazon S3 from Amazon Redshift using Amazon Redshift Spectrum
    - Count of all records in the parquet location in Amazon S3
    - Select count(*) from spectrum.parquet
    - Count of all records in the parquet location in Amazon S3 where the response is 404
    - Select count(1) from spectrum.parquet where response=404;



```
select count(*) from spectrum.parquet
```

```
count
157400
```



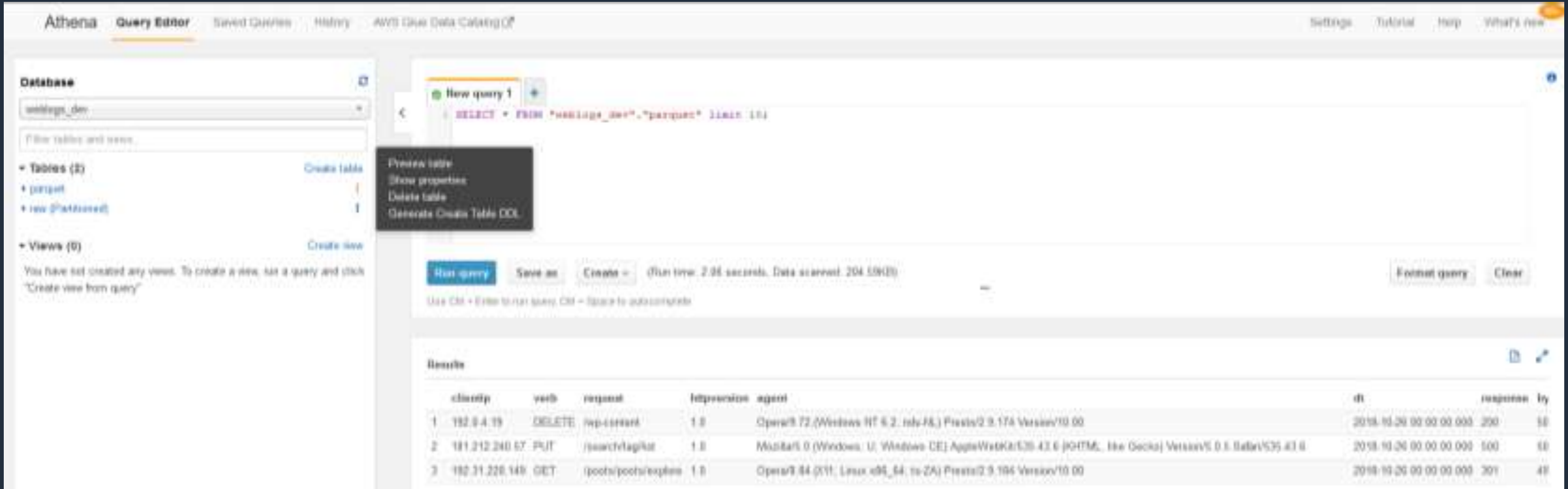
```
SELECT COUNT(1)
FROM spectrum.parquet
WHERE response = 404;
```

```
count
6400
```

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Activity 5C: Querying using Amazon Athena

- Go to the Athena console. Close the tutorial that comes up.
- In the Athena console, choose the "**weblogs_<environment>**" on the database dropdown.
- Select "**parquet**“ from the tables section and click on the **three stacked dots** icon to preview/sample a few rows of the S3 data.



You have not created any views. To create a view, run a query and click "Create view from query"

```
SELECT * FROM "weblogs_dev"."parquet" limit 10;
```

(Run time: 2.06 seconds, Data scanned: 204.59KB)

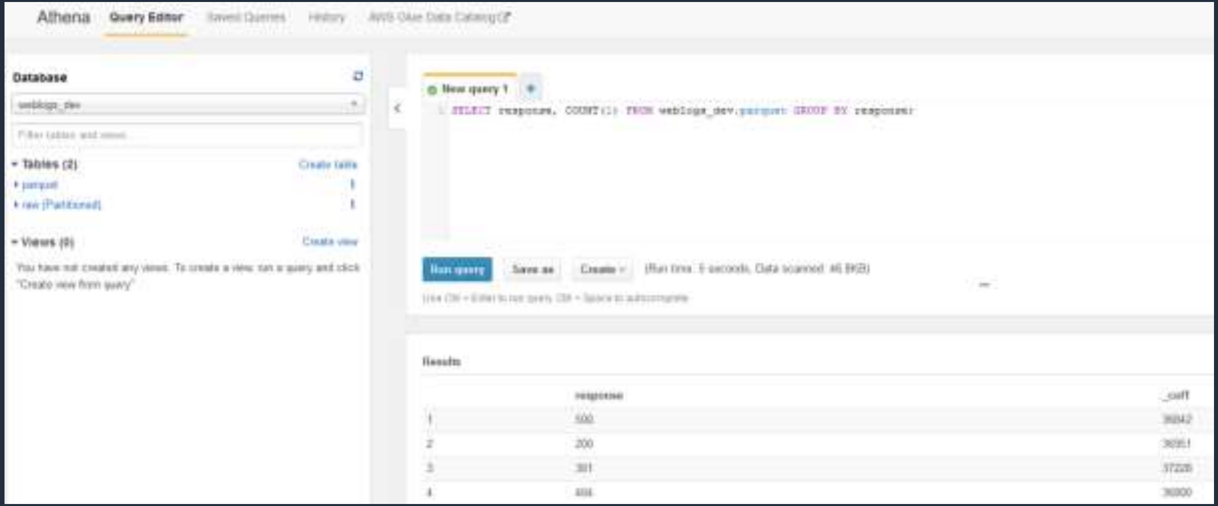
Use Ctrl + Enter to run query, Ctrl + Space to autocomplete

	clientip	verb	request	httpversion	agent	dt	response	by
1	192.0.4.19	DELETE	/wp-content	1.0	Opera/9.72 (Windows NT 6.2; nds-NL) Presto/2.9.174 Version/10.00	2018-10-26 00:00:00.000	200	[illegible]
2	181.212.240.57	PUT	/search/tag/list	1.0	Mozilla/5.0 (Windows; U; Windows CE) AppleWebKit/535.43.6 (KHTML, like Gecko) Version/5.0.5 Safari/535.43.6	2018-10-26 00:00:00.000	500	[illegible]
3	192.31.228.149	GET	/posts/posts/explore	1.0	Opera/8.84 (X11; Linux x86_64; ts-ZA) Presto/2.9.184 Version/10.00	2018-10-26 00:00:00.000	301	[illegible]

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Activity 5C: Querying using Amazon Athena

- Athena allows you to run interactive queries against parquet data in Amazon S3
- Run the queries from the AthenaSQL from the BigDataWorkshop.zip
  - Count of each clientIP
  - Count of each response code



Database: weblogs_dev

You have not created any views. To create a view, run a query and click "Create view from query"

```
SELECT response, COUNT(1) FROM weblogs_dev.parquet GROUP BY response;
```

(Run time: 5 seconds, Data scanned: 46.8KB)

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

	response	_col1
1	500	36842
2	200	36951
3	301	37226
4	404	36930

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Congratulations!

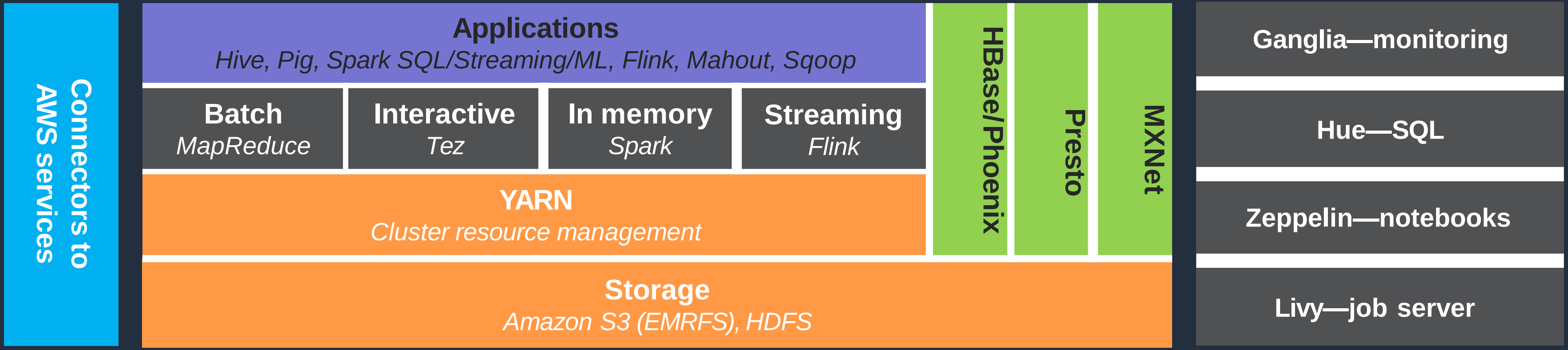
SHUT DOWN YOUR LAB!

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Optional exercise: Data processing with Amazon EMR

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Amazon EMR service



Connectors to AWS services

Applications
Hive, Pig, Spark SQL/Streaming/ML, Flink, Mahout, Sqoop

Batch
MapReduce

Interactive
Tez

In memory
Spark

Streaming
Flink

YARN
Cluster resource management

HBase/Phoenix

Presto

MXNet

Storage
Amazon S3 (EMRFS), HDFS

Ganglia—monitoring

Hue—SQL

Zeppelin—notebooks

Livy—job server

Amazon EMR service

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

On-cluster UIs

HUE

SQL editor, workflow designer, metastore browser

Notebooks

Design and execute queries and workloads

APACHE Spark

hadoop

APACHE HBASE

TEZ

Flink

Manage applications

jupyter

R Studio

And more using bootstrap actions!

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

The Hadoop ecosystem can run in Amazon EMR

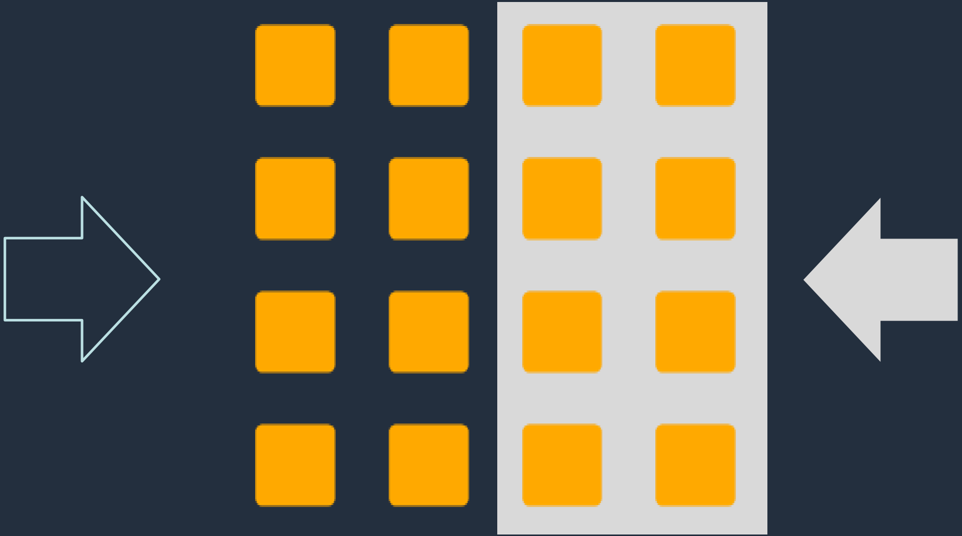
Spark
TEZ
MapReduce
hadoop HDFS
FLUME
Flink
MAPR
AVRO
APACHE Phoenix
APACHE HBASE
Apache Ranger
presto
cascading
R
chukwa
HIVE
mahout

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Easy-to-use Amazon EC2 Spot Instances

- Meet SLA at predictable cost
  - On-demand for core nodes
    - Standard Amazon Elastic Compute Cloud (Amazon EC2) pricing for on-demand capacity

Exceed SLA at lower cost



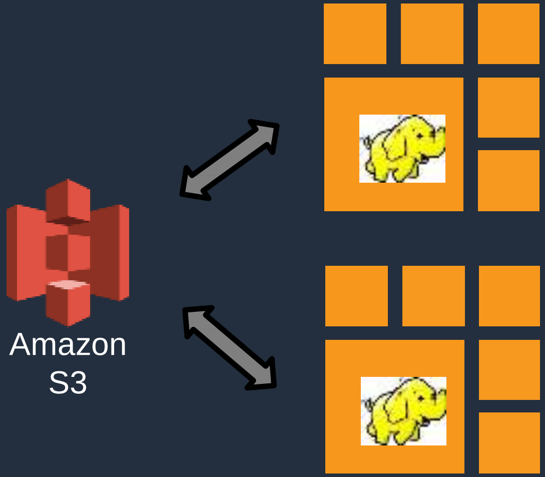
Spot Instances for task nodes

Up to 90% off Amazon EC2 on-demand pricing

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Amazon S3 as your persistent data store

- Separate compute and storage
- Resize and shut down Amazon EMR clusters with no data loss
- Point multiple Amazon EMR clusters at same data in Amazon S3



© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

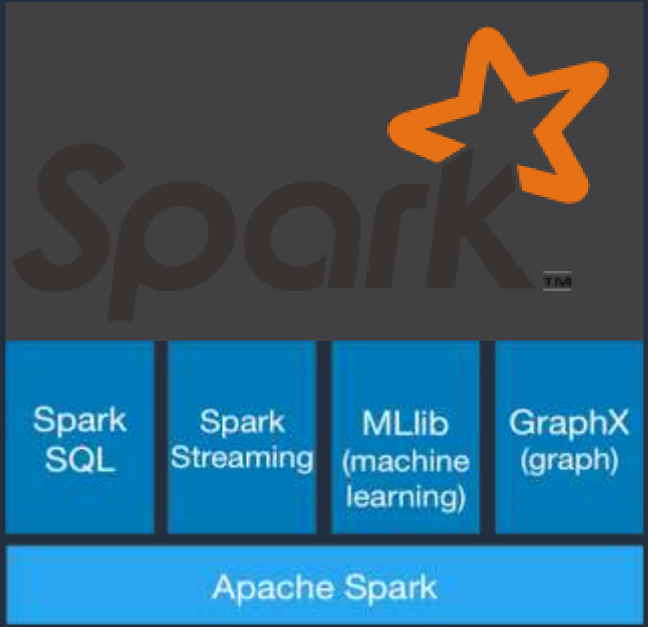
EMRFS makes it easier to leverage Amazon S3

- Transparent to applications—Use “s3://”
- Support for Amazon S3 server-side and client-side encryption
- Faster listing using EMRFS metadata
- Makes it easier to secure your clusters (fine-grained access control, Kerberos, security configurations)
  - New feature! EMR 5.14.0+ supports the ability to audit users who ran queries that accessed data in Amazon S3 through EMRFS and pushes user/group information to audit logs like AWS CloudTrail.

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Apache Spark

- Fast, general-purpose engine for large-scale data processing
- Write applications quickly in Java, Scala, or Python
- Combine SQL, streaming, and complex analytics

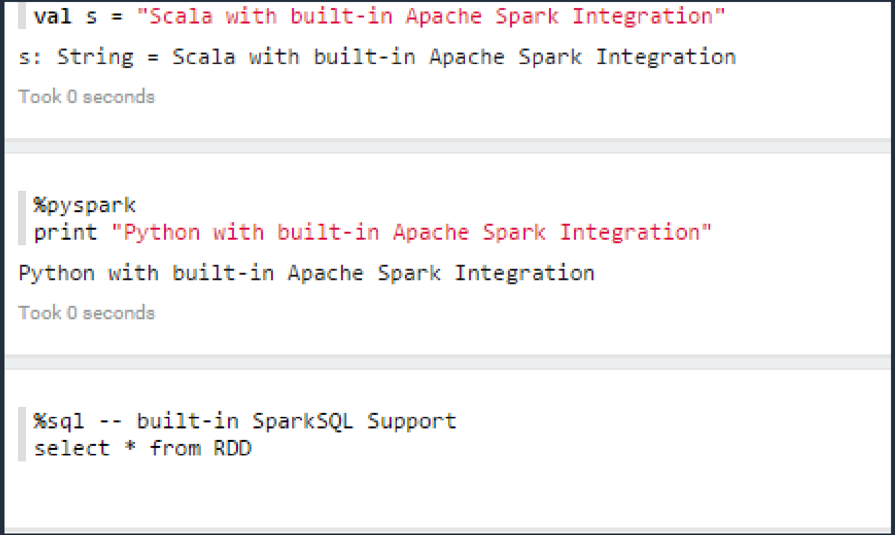


© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Apache Zeppelin

- Web-based notebook for interactive analytics
- Multiple language back end
- Apache Spark integration
- Data visualization
- Collaboration

- https://zeppelin.apache.org/



```
val s = "Scala with built-in Apache Spark Integration"
s: String = Scala with built-in Apache Spark Integration
Took 0 seconds
```

```
%pyspark
print "Python with built-in Apache Spark Integration"
Python with built-in Apache Spark Integration
Took 0 seconds
```

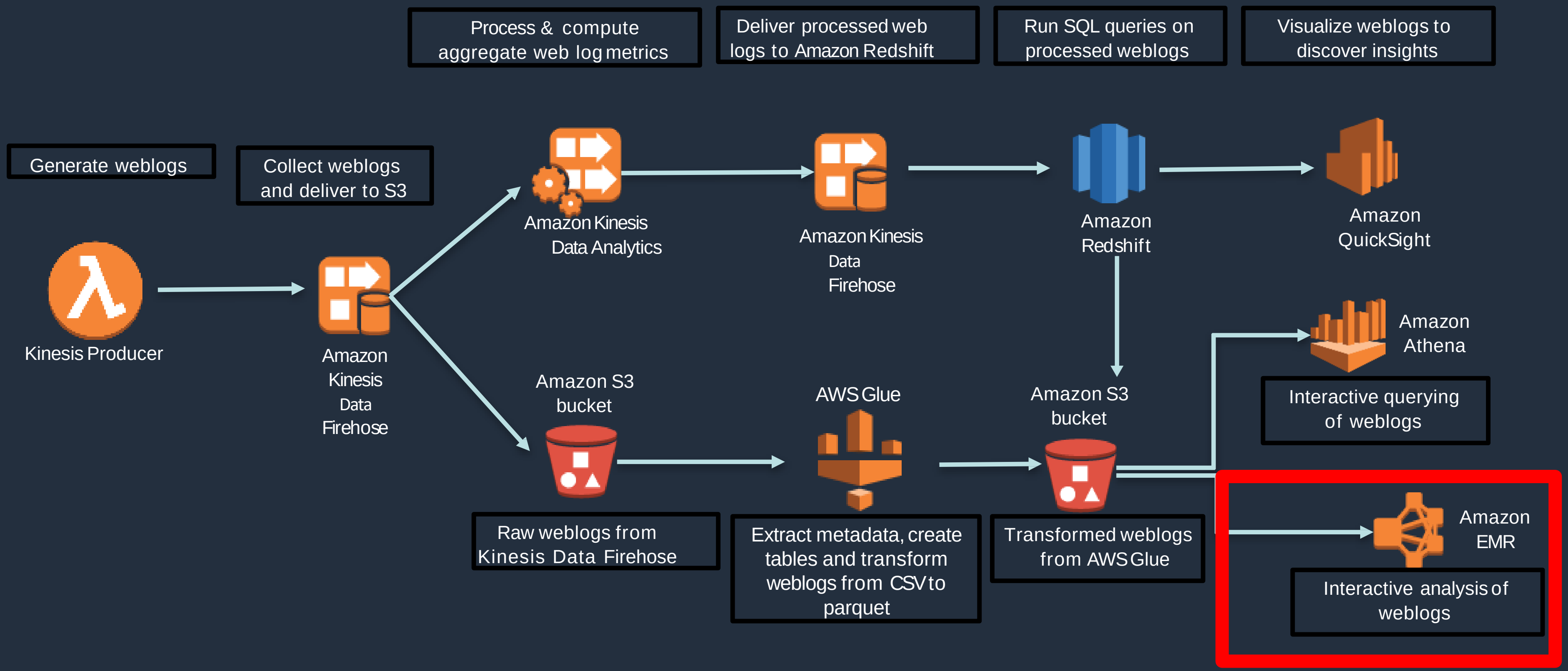
```
%sql -- built-in SparkSQL Support
select * from RDD
```

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Activity 6
Interactive analysis using Amazon EMR

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Your application architecture


Process & compute aggregate web log metrics
Deliver processed web logs to Amazon Redshift
Run SQL queries on processed weblogs
Visualize weblogs to discover insights
Generate weblogs
Collect weblogs and deliver to S3
Amazon Kinesis Data Analytics
Amazon Kinesis Data Firehose
Amazon Redshift
Amazon QuickSight
Kinesis Producer
Amazon Kinesis Data Firehose
Amazon Athena
Interactive querying of weblogs
Amazon S3 bucket
AWS Glue
Amazon S3 bucket
Amazon EMR
Interactive analysis of weblogs
Raw weblogs from Kinesis Data Firehose
Extract metadata, create tables and transform weblogs from CSV to parquet
Transformed weblogs from AWS Glue

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Activity 6: Process and query data with Amazon EMR

Time: 20 minutes

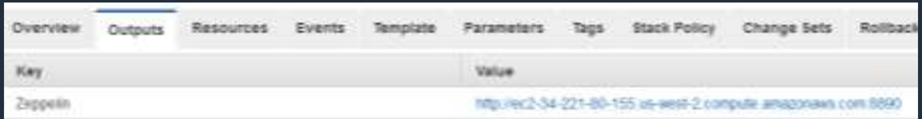
We are going to:

- Use a Zeppelin notebook to interact with Amazon EMR cluster
- Process the data in Amazon S3 using Apache Spark
- Query the data processed in the earlier stage and create simple charts

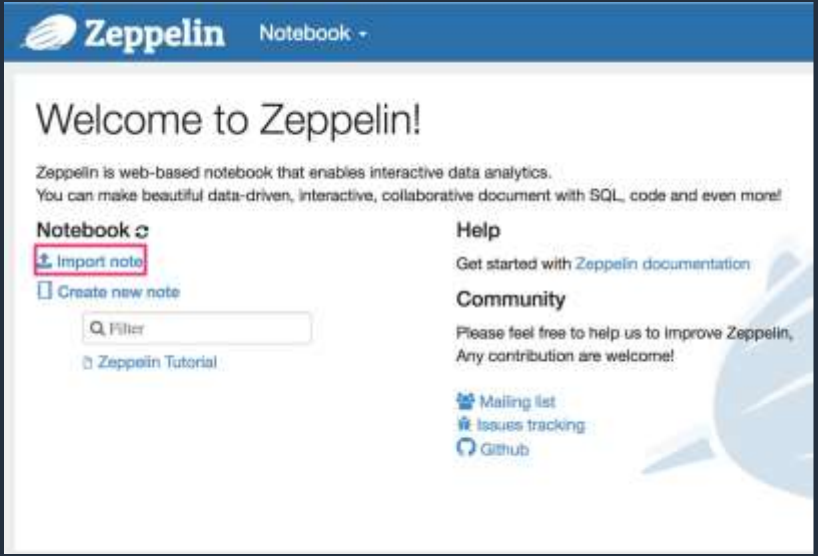
© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Activity 6A: Open the Zeppelin interface

- Copy the Zeppelin end point in **the AWS CloudFormation** output section
- Open the Zeppelin link in a new browser window
- Download the First Big Data Application json file from the BigDataWorkshop.zip
- Import the notebook using the "Import Note" link on Zeppelin interface
- **Note: Disconnect from VPN or the page will not load**



Key	Value
Zeppelin	http://ec2-34-221-80-155.us-west-2.compute.amazonaws.com:8890



Welcome to Zeppelin!

Zeppelin is web-based notebook that enables interactive data analytics.
You can make beautiful data-driven, interactive, collaborative document with SQL, code and even more!

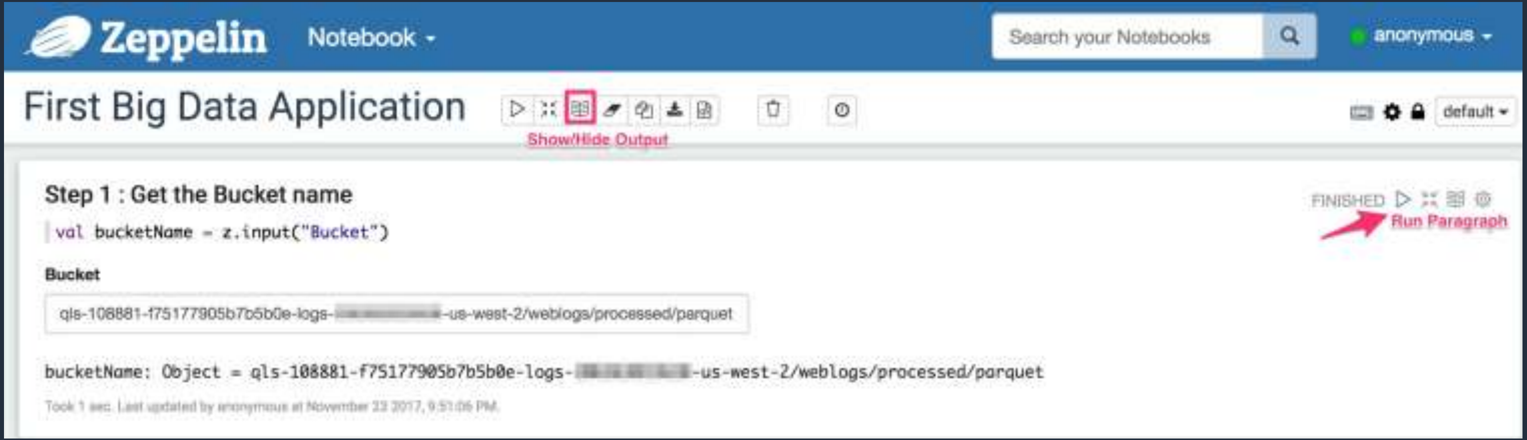
Get started with Zeppelin documentation

Please feel free to help us to improve Zeppelin,
Any contribution are welcome!

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Activity 6A: Open the Zeppelin interface

- Step 1: Run the first paragraph
- Enter **<stackname>-logs-<account>-us-west-2/processed/parquet** in the Bucket field as input
- This is where the processed parquet files were stored earlier in your S3 bucket



First Big Data Application

Step 1 : Get the Bucket name

```
val bucketName = z.input("Bucket")
```

Bucket: qls-108881-f75177905b7b5b0e-logs-[redacted]-us-west-2/weblogs/processed/parquet

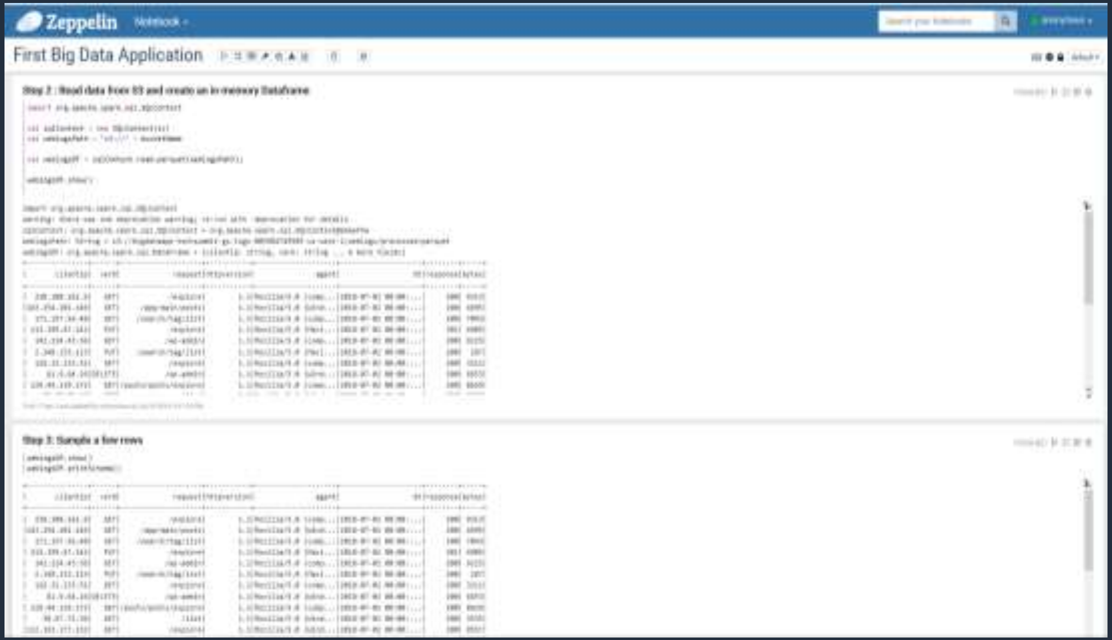
```
bucketName: Object = qls-108881-f75177905b7b5b0e-logs-[redacted]-us-west-2/weblogs/processed/parquet
```

Took 1 sec. Last updated by anonymous at November 23 2017, 9:51:06 PM.

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Activity 6B: Run the notebook

- Execute Step 2
  - Create a data frame with the parquet files from the AWS Glue ETL job
- Execute Step 3
  - Sample a few rows



First Big Data Application

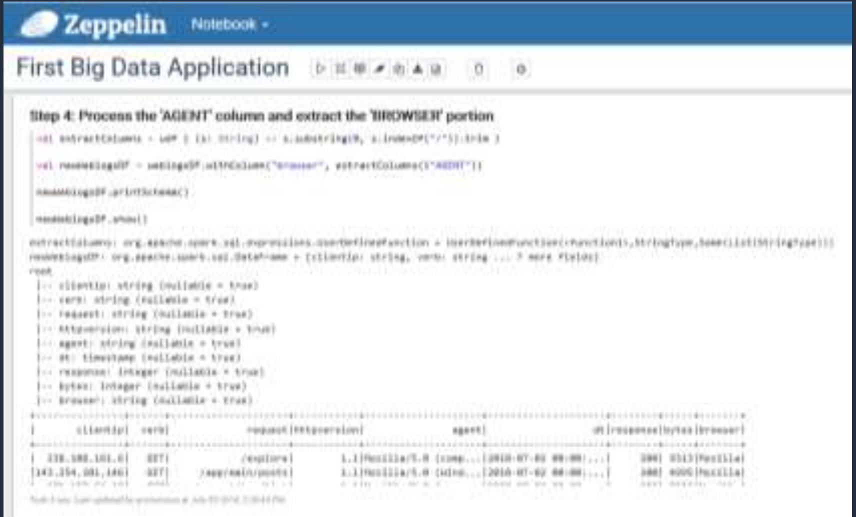
Step 2 : Read data from S3 and create an in-memory Dataframe

Step 3: Sample a few rows

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Activity 6B: Run the notebook

- Execute Step 4 to process the data
  - Notice how the “AGENT” field consists of the “BROWSER” at the beginning of the column value. Let’s extract the browser from the agent field.
- Create a UDF to extract the browser and add to the data frame
- Print the new data frame



First Big Data Application

Step 4: Process the 'AGENT' column and extract the 'BROWSER' portion

```
val extractColumn = udf { (s: String) => s.substring(0, s.indexOf("/")).trim }

val newWeblogsDF = weblogsDF.withColumn("browser", extractColumn($"AGENT"))

newWeblogsDF.printSchema()

newWeblogsDF.show()
```

```
extractColumn: org.apache.spark.sql.expressions.UserDefinedFunction = UserDefinedFunction(<function1>,StringType,Some(List(StringType)))
newWeblogsDF: org.apache.spark.sql.DataFrame = [clientip: string, verb: string ... 7 more fields]
root
 |-- clientip: string (nullable = true)
 |-- verb: string (nullable = true)
 |-- request: string (nullable = true)
 |-- httpversion: string (nullable = true)
 |-- agent: string (nullable = true)
 |-- dt: timestamp (nullable = true)
 |-- response: integer (nullable = true)
 |-- bytes: integer (nullable = true)
 |-- browser: string (nullable = true)
+---------------+----+------------------+-----------+--------------------+--------------------+--------+-----+-------+
|       clientip|verb|           request|httpversion|               agent|                  dt|response|bytes|browser|
+---------------+----+------------------+-----------+--------------------+--------------------+--------+-----+-------+
|  238.188.161.6| GET|          /explore|        1.1|Mozilla/5.0 (comp...|2018-07-03 00:00:...|     200| 5513|Mozilla|
|143.254.281.146| GET|   /app/main/posts|        1.1|Mozilla/5.0 (Wind...|2018-07-02 00:00:...|     200| 4006|Mozilla|
```

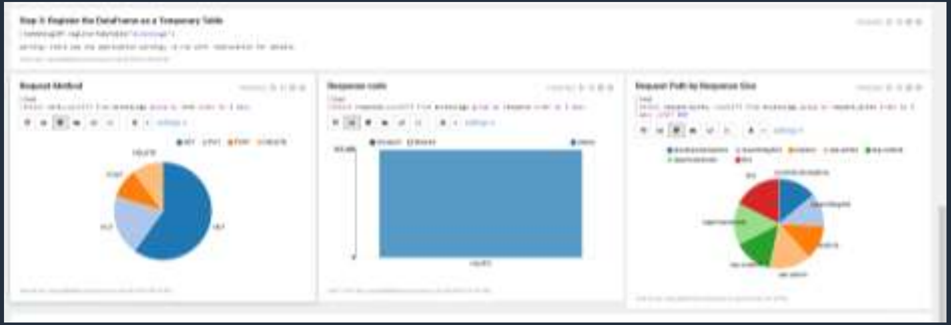
[illegible]

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Activity 6B: Run the notebook

- Execute Step 6
  - Register the data frame as a temporary table
  - Now you can run SQL queries on the temporary tables

- Execute the next 3 steps and observe the charts created
  - What did you learn about the data set?



© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Review: Interactive analysis using Amazon EMR

- You just learned on how to process and query data using Amazon EMR with Apache Spark
- Amazon EMR has many other frameworks available for you to use
  - Hive, Presto, Flink, Pig, MapReduce
  - Hue, Oozie, HBase

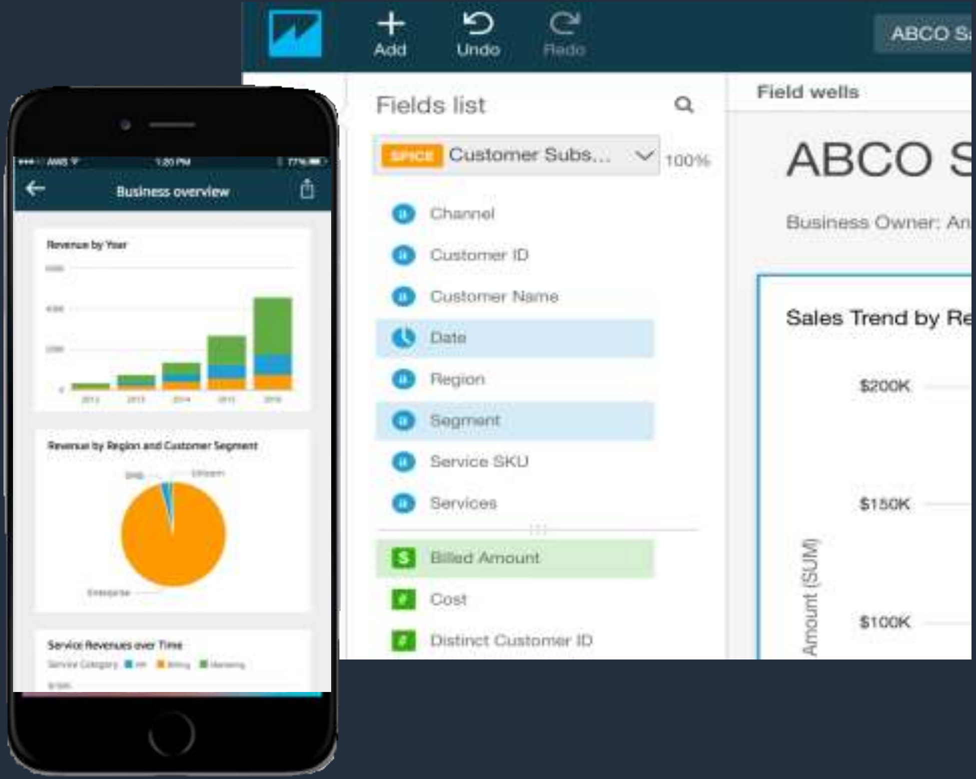
© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Optional exercise: Data visualization with Amazon QuickSight

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Amazon QuickSight

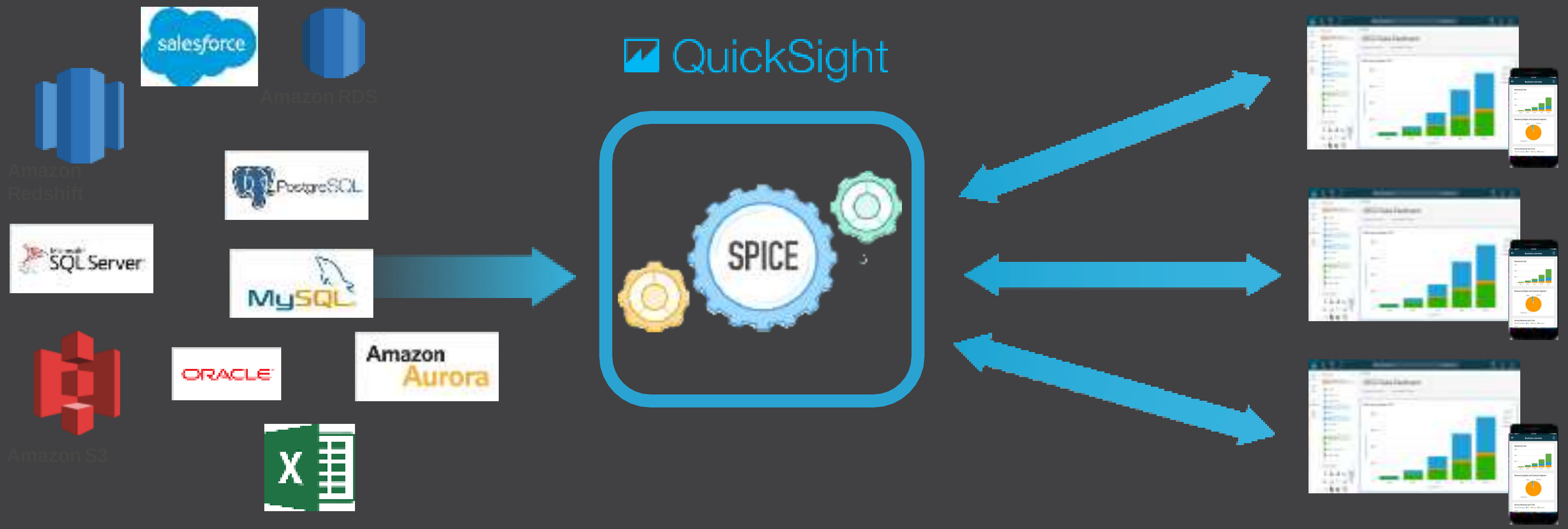
- **Fast, easy interactive analytics for anyone, everywhere**
- Ease of use targeted at business users
- Blazing-fast performance powered by SPICE
- Broad connectivity with AWS data services, on-premises data, files, and business applications
- Cloud-native solution that scales automatically
- 1/10th the cost of traditional BI solutions
- Create, share, and collaborate with anyone in your organization, on the web, or on mobile



© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Connect, SPICE, analyze

- Amazon QuickSight allows you to connect to data from a wide variety of AWS, third-party, and on-premises sources and import it to SPICE or query directly
- Users can then easily explore, analyze, and share their insights with anyone

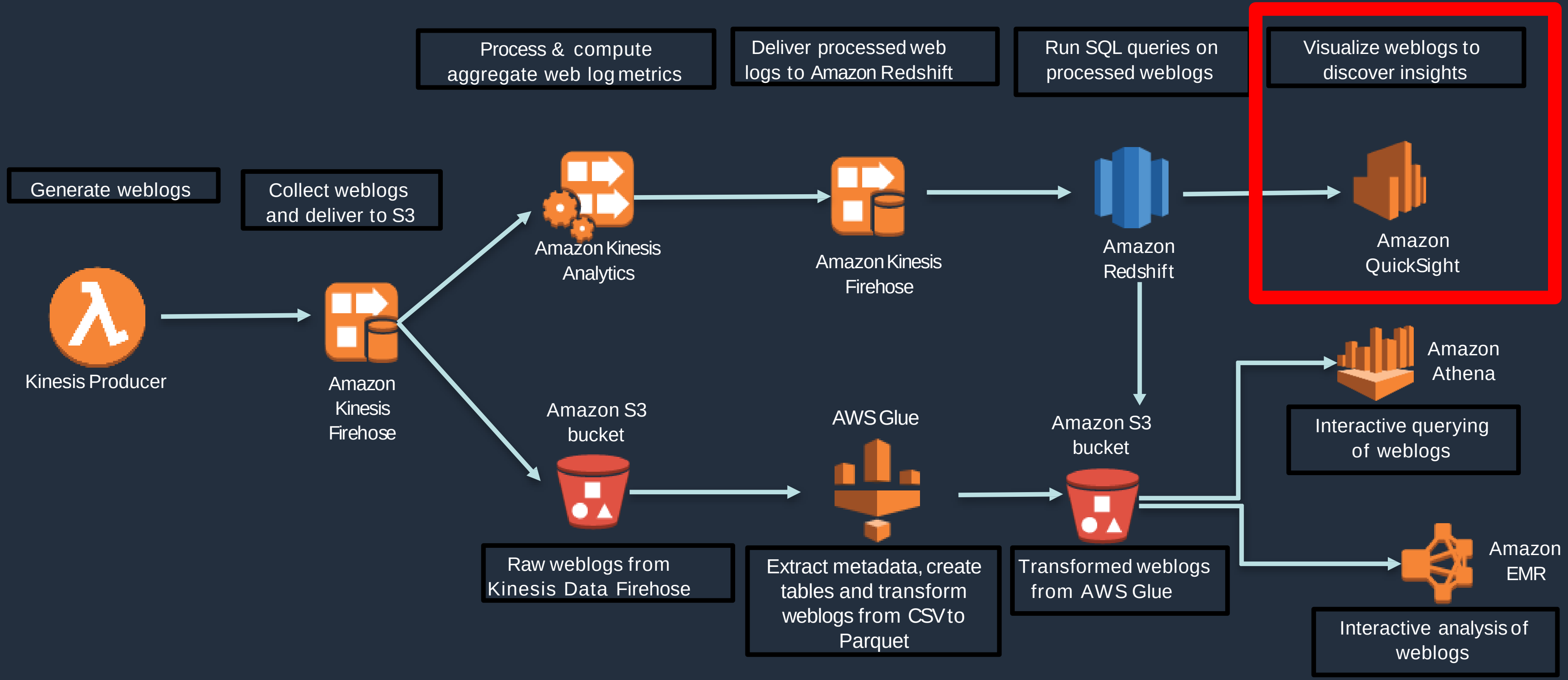


© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Activity 7
Visualize results in Amazon QuickSight

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Your application architecture



Generate weblogs

Collect weblogs and deliver to S3

Process & compute aggregate web log metrics

Deliver processed web logs to Amazon Redshift

Run SQL queries on processed weblogs

Visualize weblogs to discover insights

Raw weblogs from Kinesis Data Firehose

Extract metadata, create tables and transform weblogs from CSV to Parquet

Transformed weblogs from AWS Glue

Interactive querying of weblogs

Interactive analysis of weblogs

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Activity 7: Visualization with Amazon QuickSight

Time: 10 mins

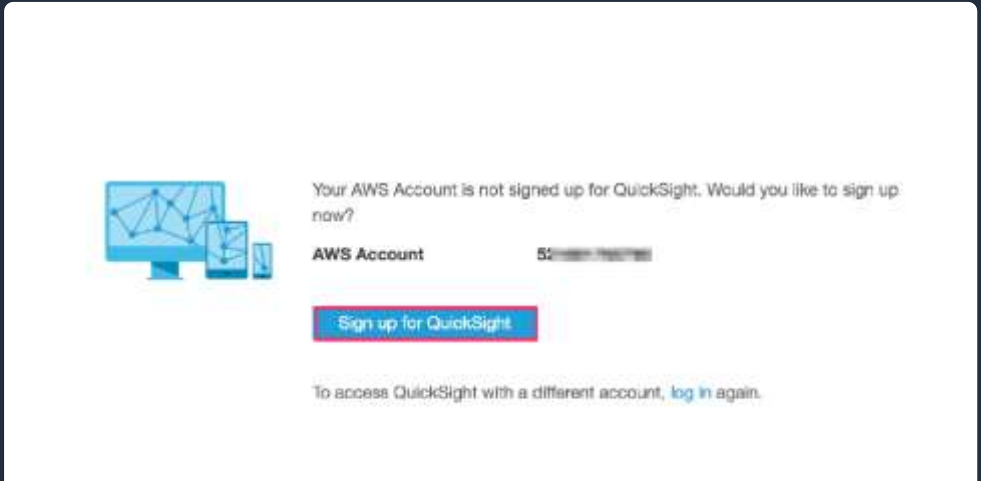
We are going to:

A. Register for an Amazon QuickSight account
B. Connect to the Amazon Redshift cluster
C. Create visualizations for analysis to answer questions like:
    A. What are the most common http requests and how successful (response code of 200) are they?
    B. Which are the most requested URIs?

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Activity 7A: Amazon QuickSight registration

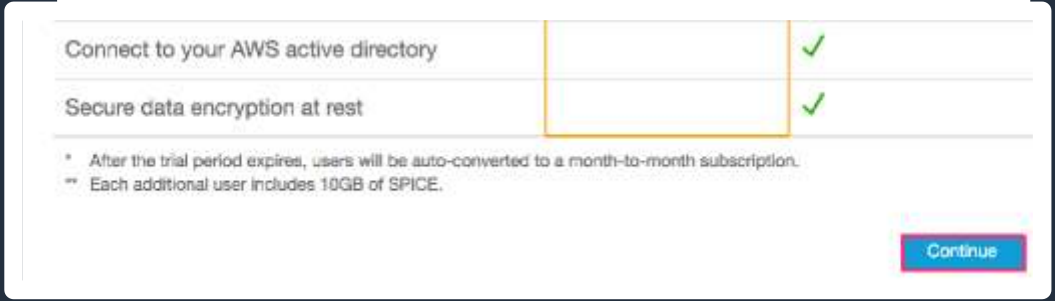
- Go to AWS console, click on Amazon QuickSight from the analytics section
- Click on "Sign up" in the next window
- Make sure the subscription type is standard and click "Continue" on the next screen



Your AWS Account is not signed up for QuickSight. Would you like to sign up now?

AWS Account [redacted]

To access QuickSight with a different account, log in again.



Connect to your AWS active directory		✓
Secure data encryption at rest		✓

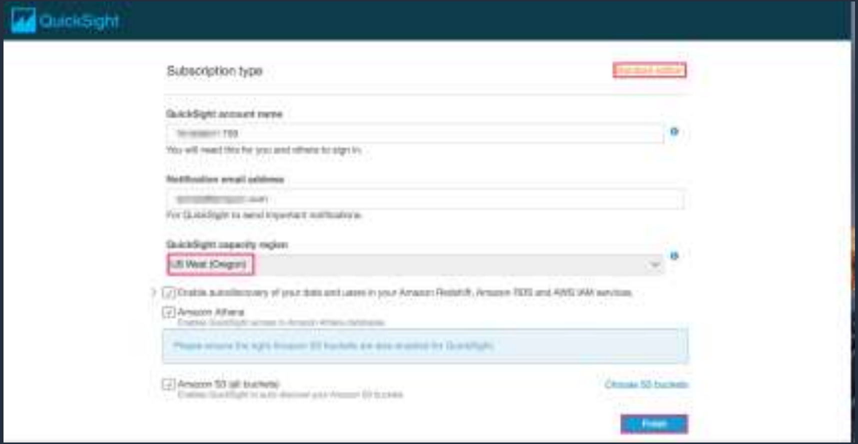
* After the trial period expires, users will be auto-converted to a month-to-month subscription.

** Each additional user includes 10GB of SPICE.

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Activity 7A: Amazon QuickSight registration

- On the "Subscription type" page, enter the account name (see **note** below)
- Enter your **email address**
- Select **US West region**
- Check the "**Amazon S3 (all buckets)**" box
- **Note: Amazon QuickSight account name is your AWS account number**



QuickSight

Subscription type [illegible]

QuickSight account name

[redacted] 789

You will need this for you and others to sign in.

Notification email address

[redacted].com

For QuickSight to send important notifications.

QuickSight capacity region

US West (Oregon)

☑ Enable autodiscovery of your data and users in your Amazon Redshift, Amazon RDS and AWS IAM services.

☑ Amazon Athena
Enables QuickSight access to Amazon Athena databases

Please ensure the right Amazon S3 buckets are also enabled for QuickSight.

☑ Amazon S3 (all buckets)
Enables QuickSight to auto-discover your Amazon S3 buckets

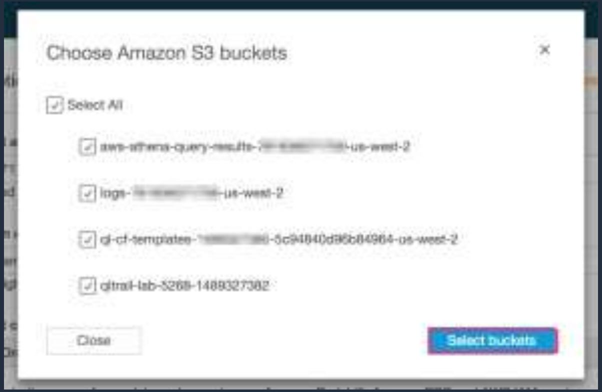
Choose S3 buckets

Finish

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Activity 7A: Amazon QuickSight registration

- If a popup box to choose Amazon S3 buckets appears, click "**Select buckets**"
- Click on "**Go to Amazon QuickSight**"
- Dismiss welcome screen
- Make sure you are in us-west-2 (Oregon)



Choose Amazon S3 buckets ×

☑ Select All

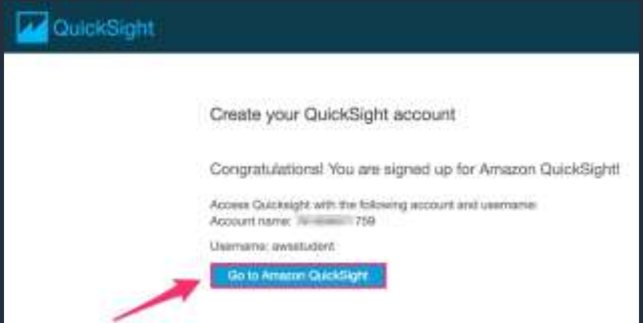
☑ aws-athena-query-results-[redacted]-us-west-2

☑ logs-[redacted]-us-west-2

☑ ql-cf-templates-[redacted]-5c94840d96b84964-us-west-2

☑ qltrail-lab-5268-1489327382

Close | Select buckets



QuickSight

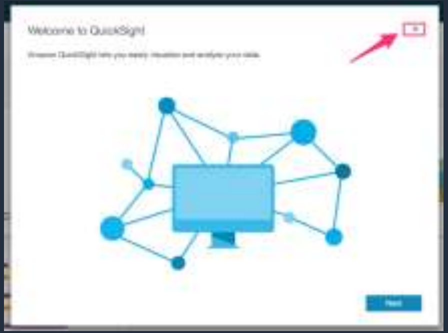
Create your QuickSight account

Congratulations! You are signed up for Amazon QuickSight!

Access Quicksight with the following account and username:
Account name: [redacted]759

Username: awsstudent

Go to Amazon QuickSight



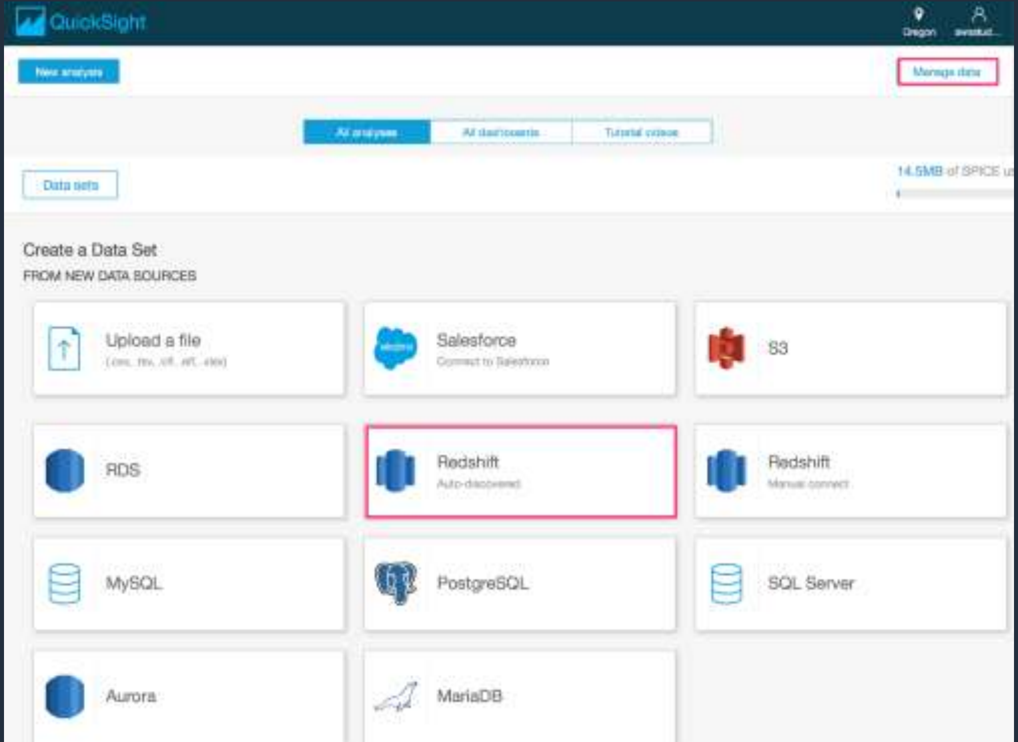
Welcome to QuickSight

Amazon QuickSight lets you easily visualize and analyze your data.

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

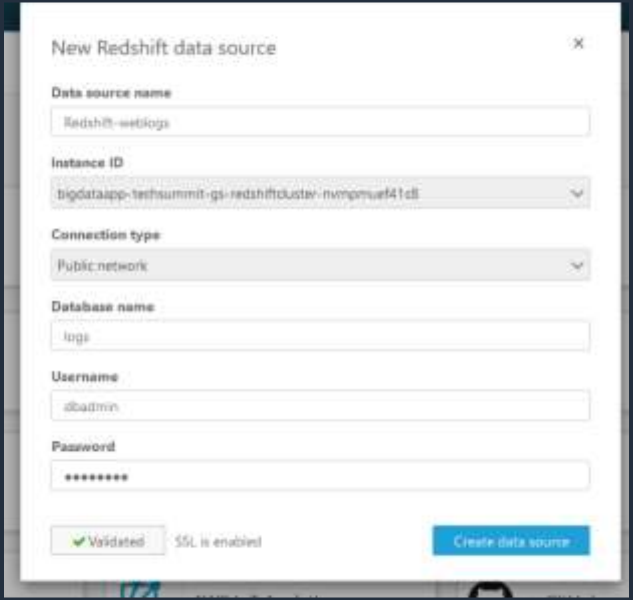
Activity 7B: Connect to data source

- Click on "**Manage Data**" and then select "**New Data set**" to create a new data set in Amazon QuickSight.
- Choose "Redshift (Auto-discovered)" as the data source. Amazon QuickSight auto-discovers databases associated with your AWS account (Amazon Redshift database in this case).



© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Activity 7B: Connect to Amazon Redshift



New Redshift data source

Data source name: Redshift-weblogs

Instance ID: bigdataapp-techsummit-gs-redshiftcluster-nvmpmuef41c8

Connection type: Public network

Database name: logs

Username: dbadmin

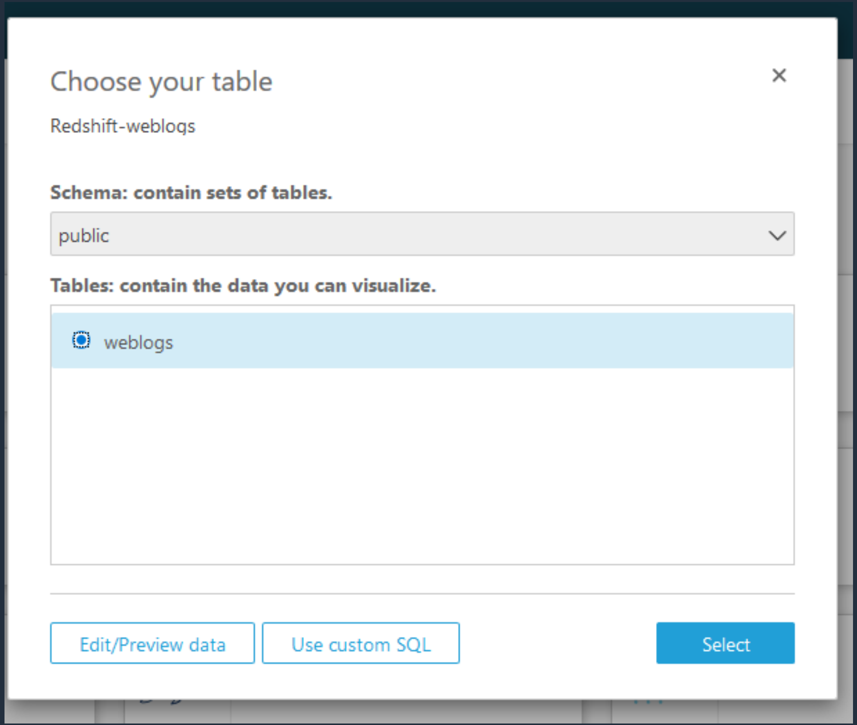
Password: •••••••••

Note: Use ”dbadmin” as the username. You can get the Amazon Redshift database password by navigating to the AWS CloudFormation outputs section.

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Activity 7C: Choose your weblogs Amazon Redshift table

- Select “public” schema
- Select “weblogs” in tables



Choose your table

Redshift-weblogs

Schema: contain sets of tables.

public

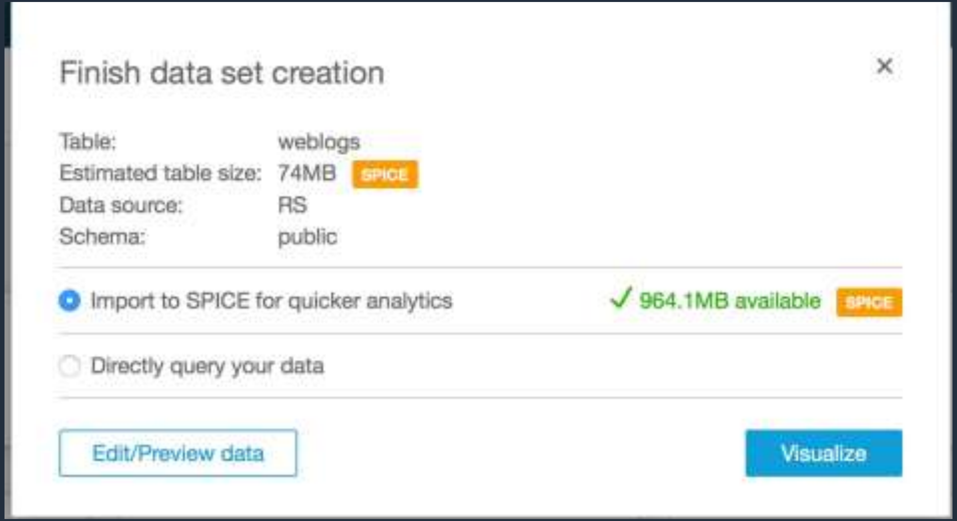
Tables: contain the data you can visualize.

● weblogs

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Activity 7D: Ingest data into SPICE

- SPICE is Amazon QuickSight's in-memory optimized calculation engine, designed specifically for fast, interactive data visualization
- You can improve the performance of database data sets by importing the data into SPICE instead of using a direct query to the database



Finish data set creation

Table: weblogs
Estimated table size: 74MB SPICE
Data source: RS
Schema: public

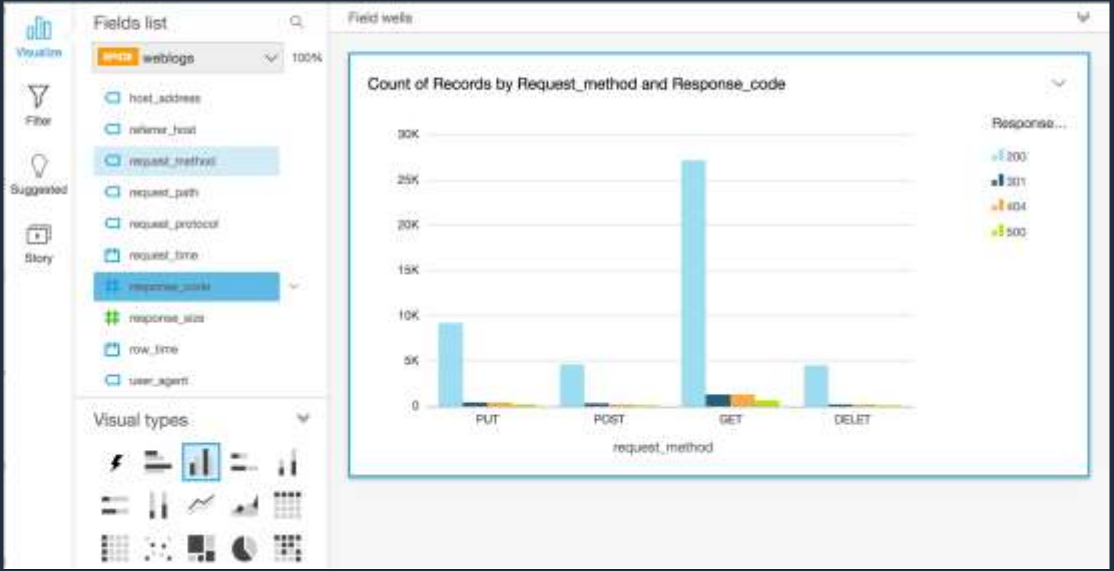
● Import to SPICE for quicker analytics ✓ 964.1MB available SPICE

○ Directly query your data

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Activity 7E: Creating your first analysis

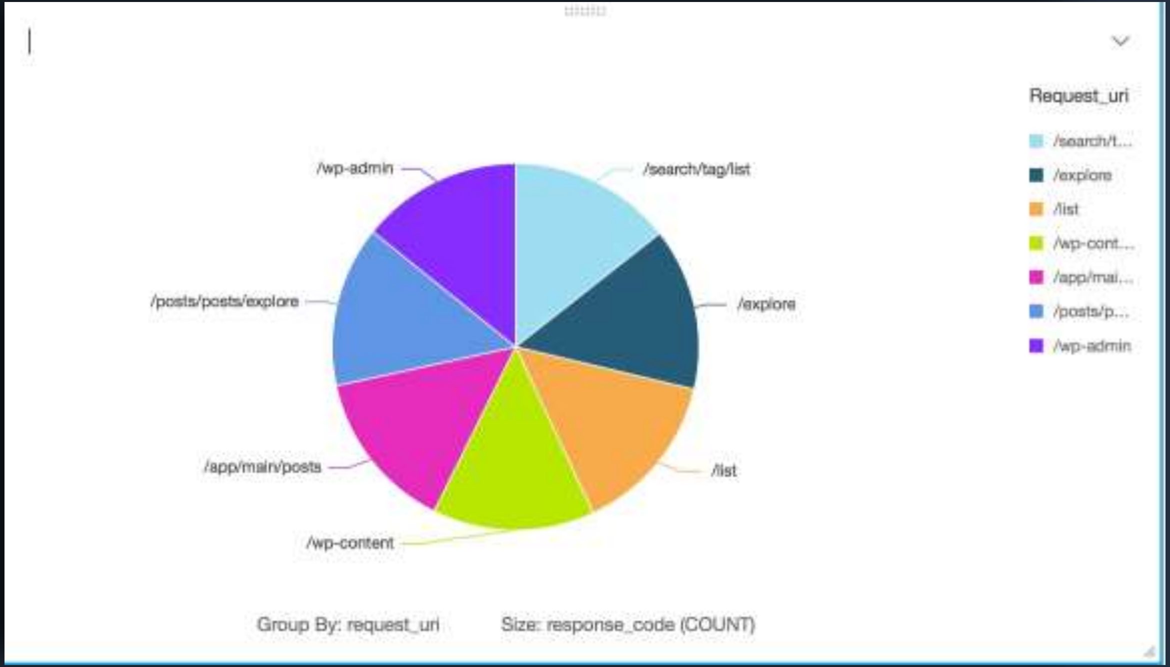
- What are the most requested http request types and their corresponding response codes for this site?
- Simply select request, response, and let AUTOGRAPH create the optimal visualization



© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Review - Creating your analysis

- Exercise: Add a visual to demonstrate which URI are the most requested



© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved