Build Your First Big Data Application on AWS

Ben Willett
Sr. Solutions Architect

Pablo Redondo Sanchez
Sr. Solutions Architect

Mike George
Sr. Technical Account Manager

Graham Zulauf
Sr. Solutions Architect

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:

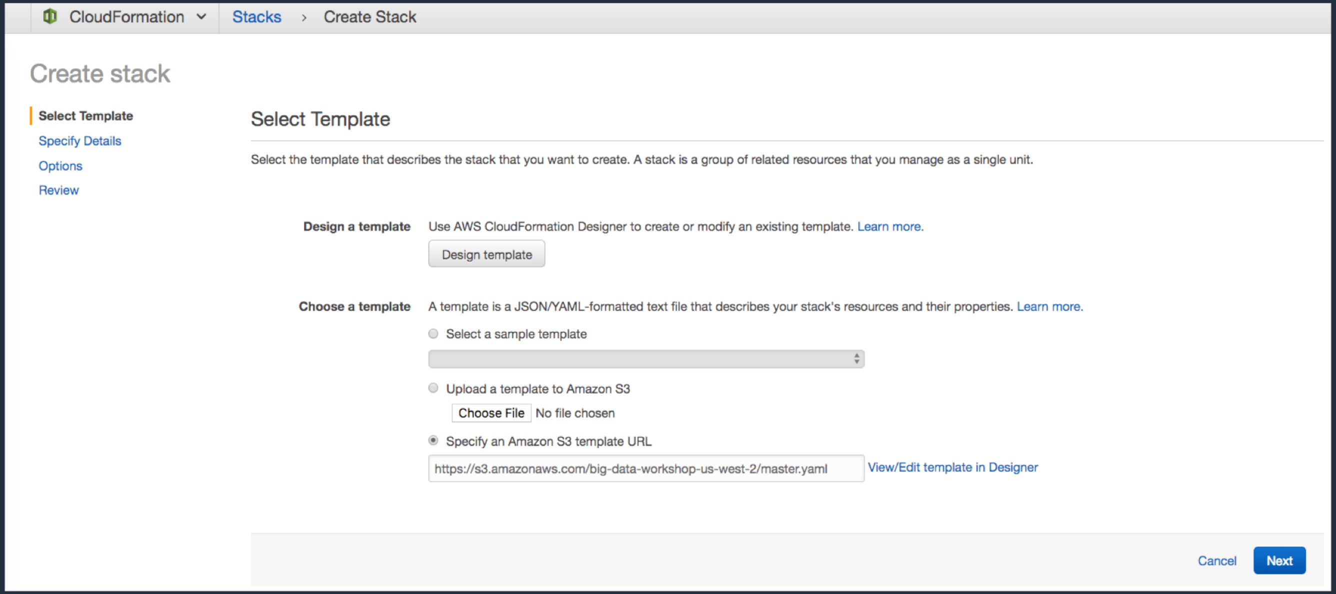
AWS Region	Short name	New VPC	Existing VPC
US East (N. Virginia)	us-east-1	Launch Stack	Launch Stack
US West (Oregon)	us-west-2	Launch Stack	Launch Stack

© 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

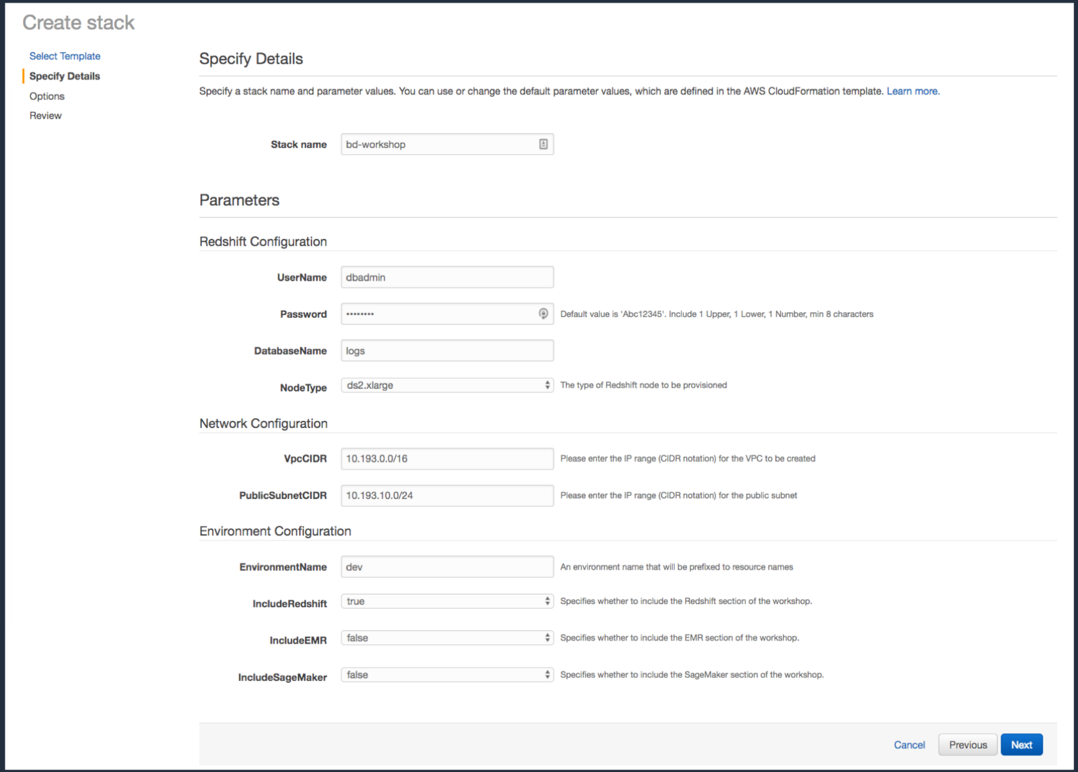
- Verify the Amazon Simple Storage Service (Amazon S3) template URL, and click “Next”



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

Launch AWS CloudFormation template - Continued

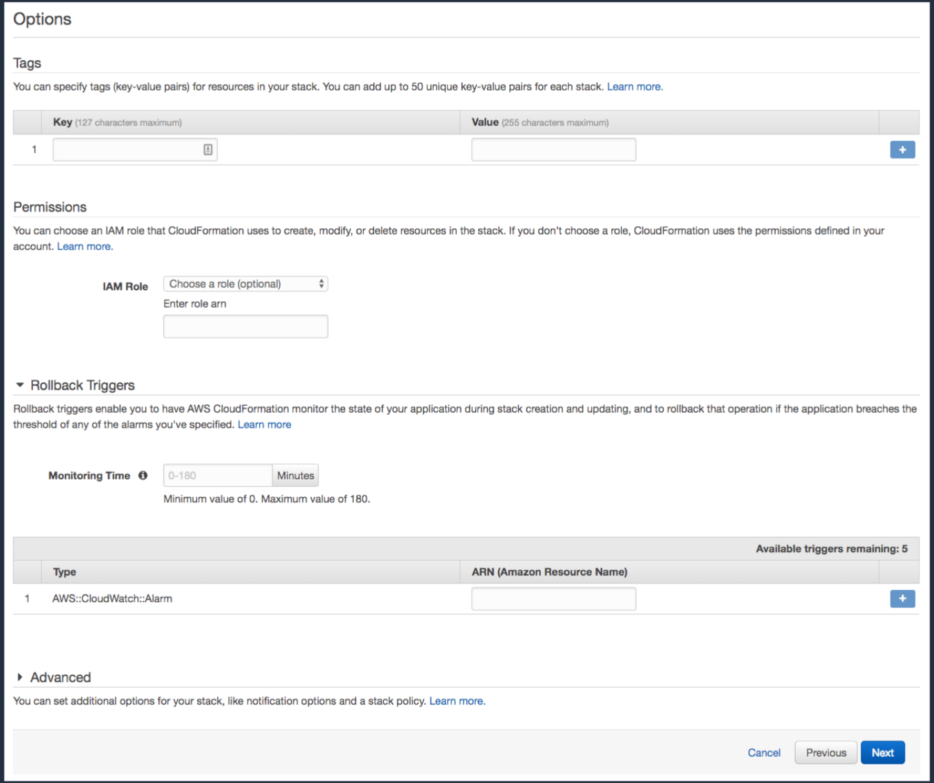
- All defaults can be left as-is for this lab. Depending on if you selected new or existing, the only change is picking a CIDR range for the VPC or selecting an existing VPC in your account.
- If there is a Amazon Redshift password in there, leave as is.
- If there is no password, enter Abc12345 as the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



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

Launch AWS CloudFormation template - Continued

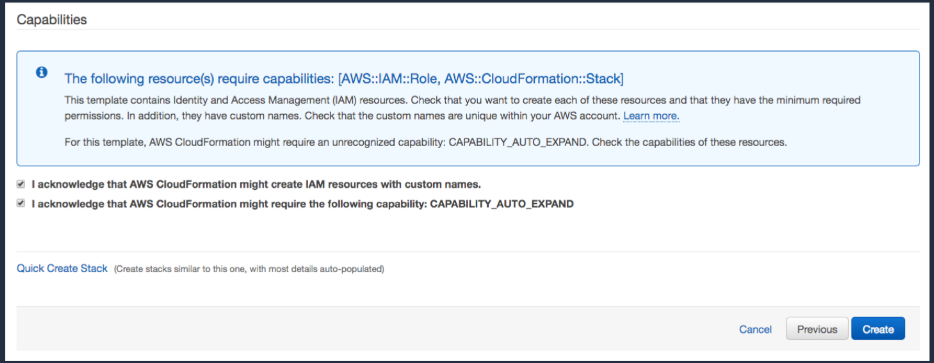
- Leave the Options at their defaults and click "Next".



© 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)



© 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

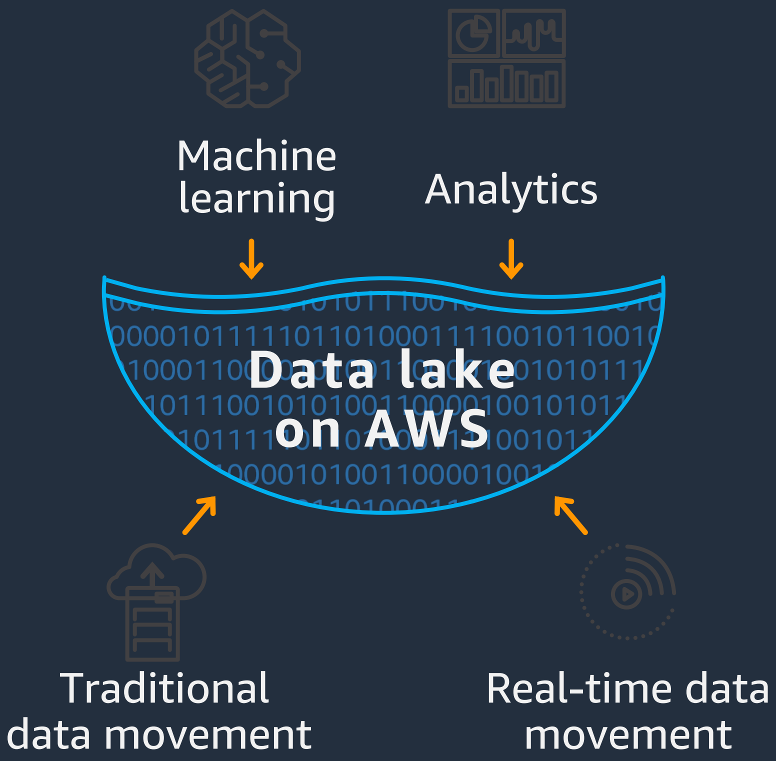
© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Important! – Delete CloudFormation Stacks When Done

- **All AWS resources** created in this lab **should be removed** by deleting the CloudFormation Stacks in your AWS account so **you stop paying for them**
- **If you are not sure, ask for help!**

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

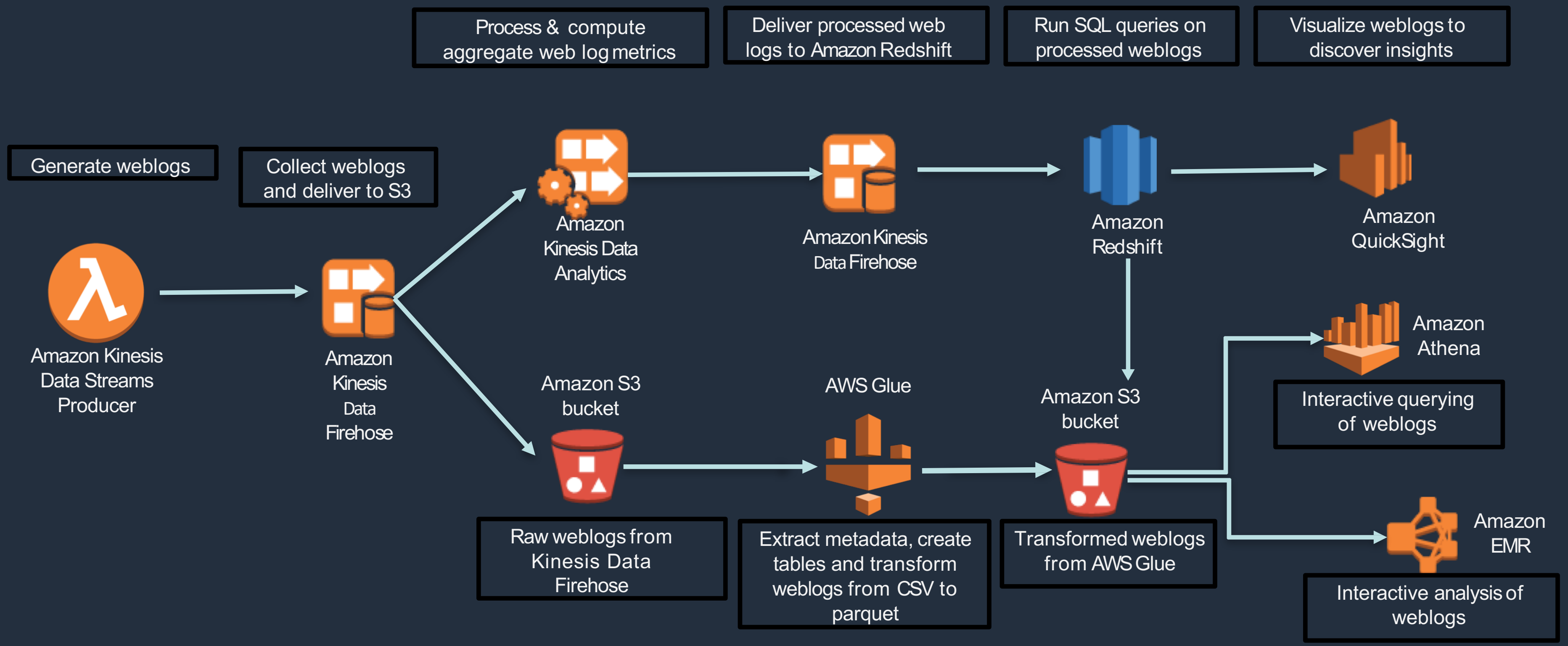
Why analytics on AWS?



- 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



© 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

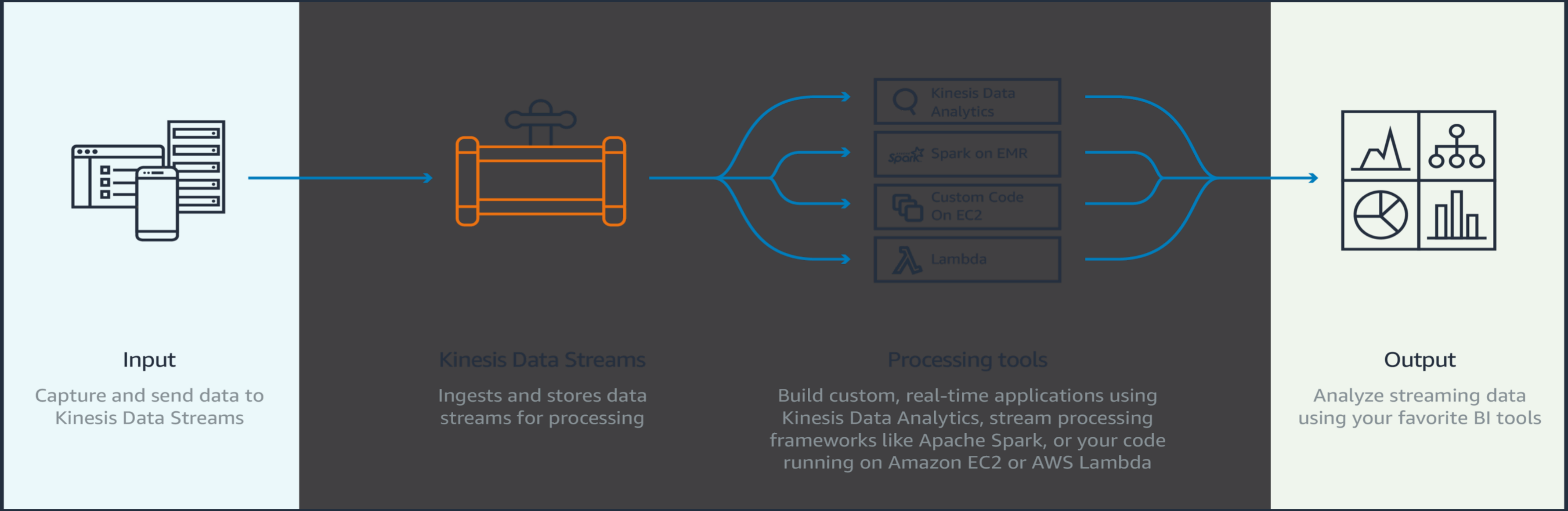
Load data streams into data stores

Kinesis Data Analytics

Analyze data streams with SQL

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

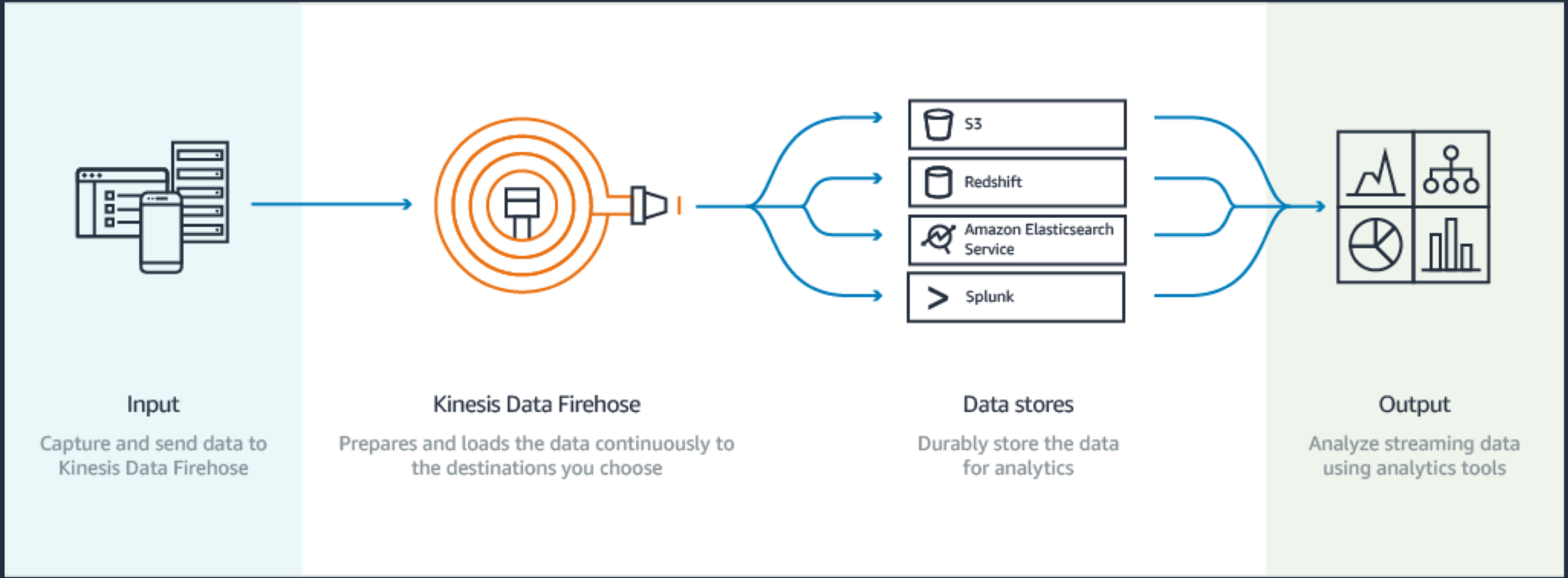
Amazon Kinesis Data Streams



- 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

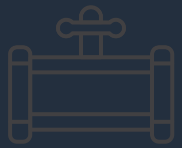
Amazon Kinesis Data Firehose



- 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

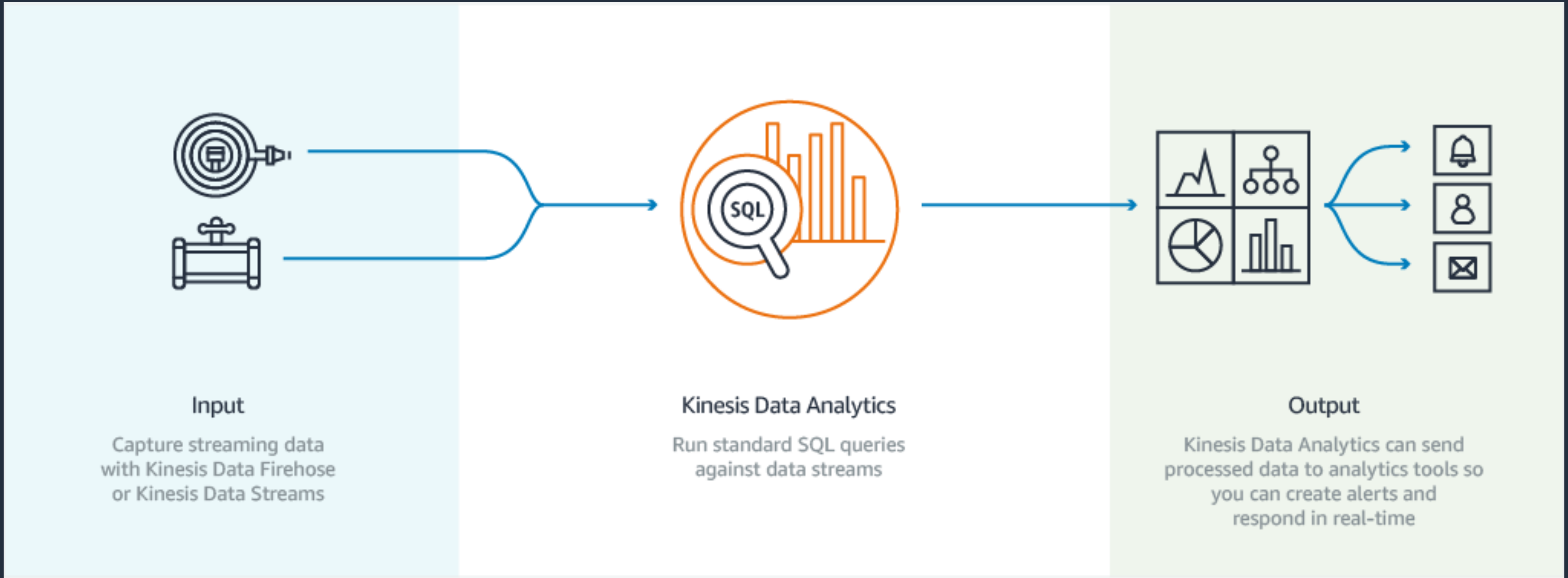
Amazon Kinesis Data Streams is for use cases that require custom processing, per incoming record, with sub-1 second processing latency, and a choice of stream processing frameworks

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

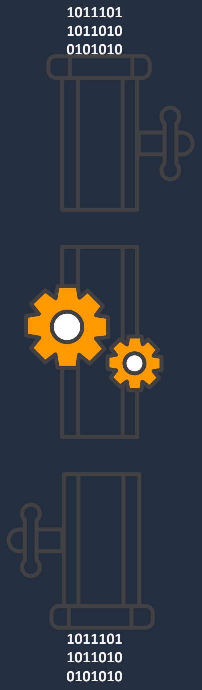
Amazon Kinesis Data Analytics



- 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

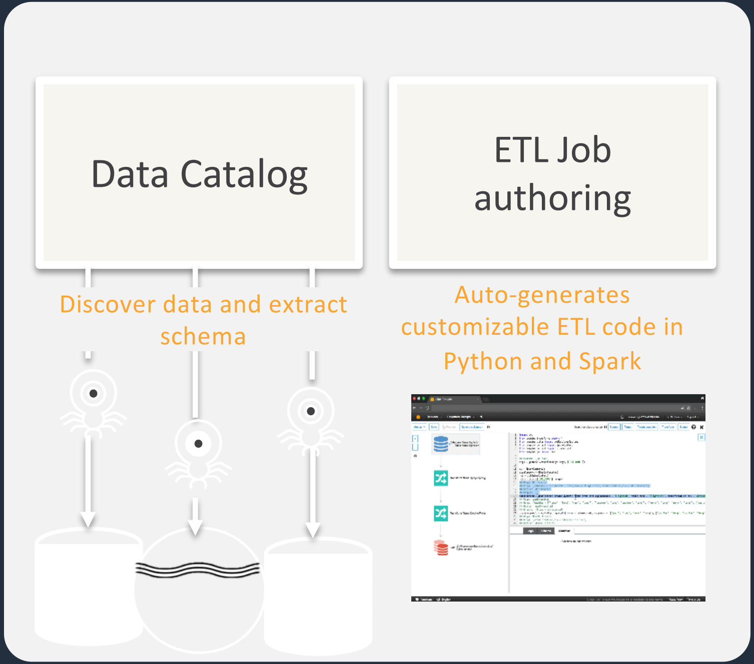
© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

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

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

AWS Glue - Components

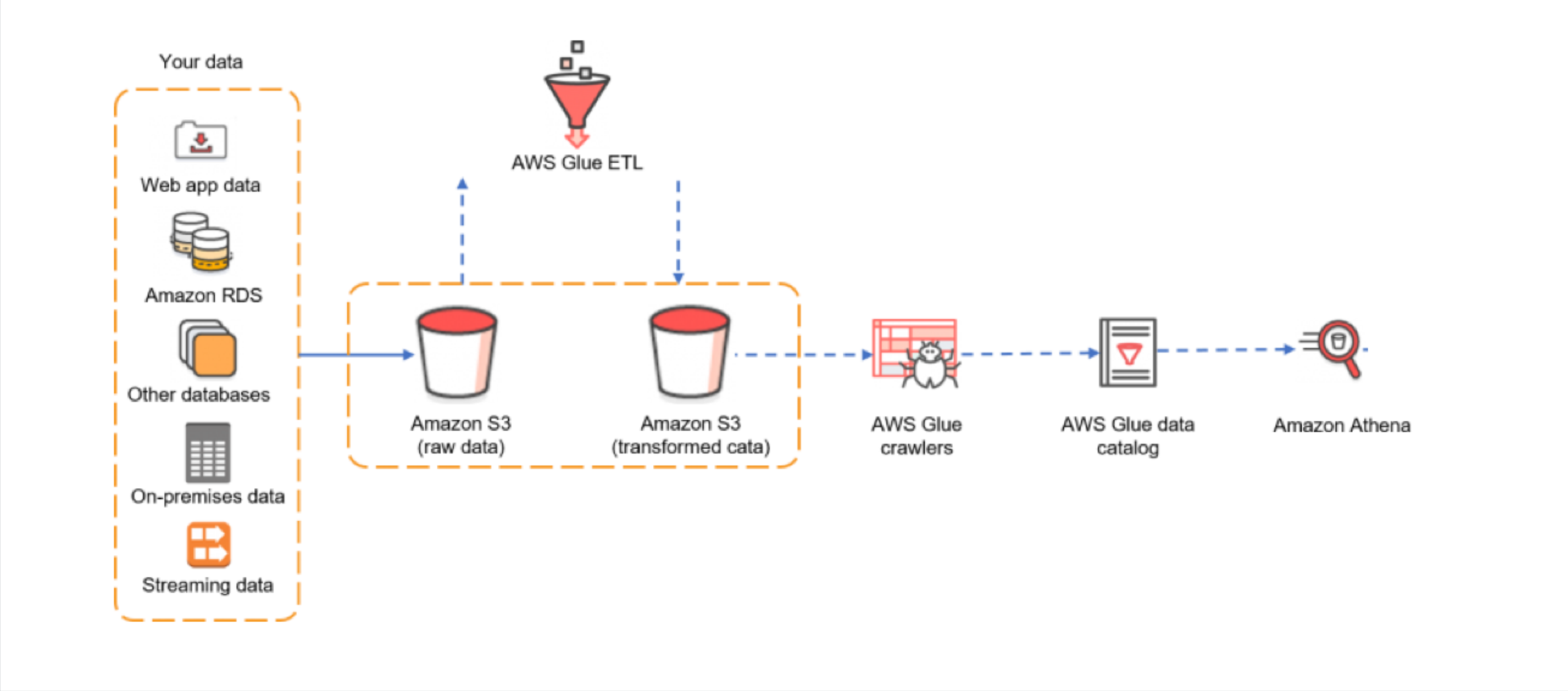
Make data discoverable



- 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

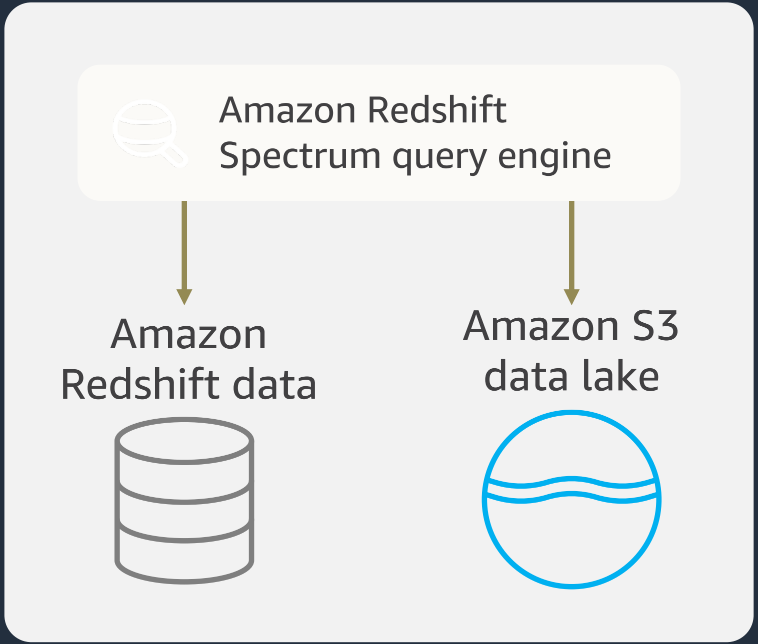
Dataset	Size on Amazon S3	Query run time	Data scanned	Cost
Data stored as text files	1 TB	236 seconds	1.15 TB	$5.75
Data stored in Apache parquet format*	130 GB	6.78 seconds	2.51 GB	$0.013
Savings / Speedup	87% less with parquet	34x faster	99% less data scanned	99.7% savings

(*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 Glue Data Catalog or your own Apache Hive Metastore

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

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

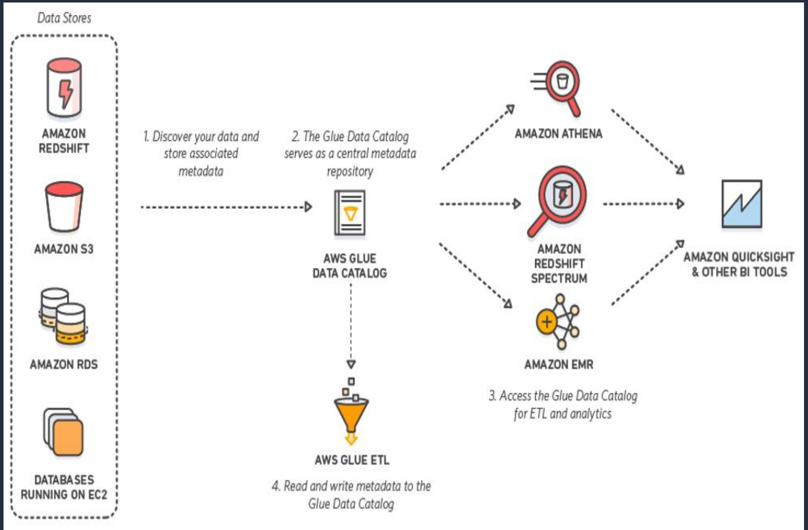
- Register external tables using AWS Glue Data Catalog, your Hive Metastore client, or from Amazon Redshift CREATE EXTERNAL TABLE syntax

  ```
  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



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

Workshop activities schedule

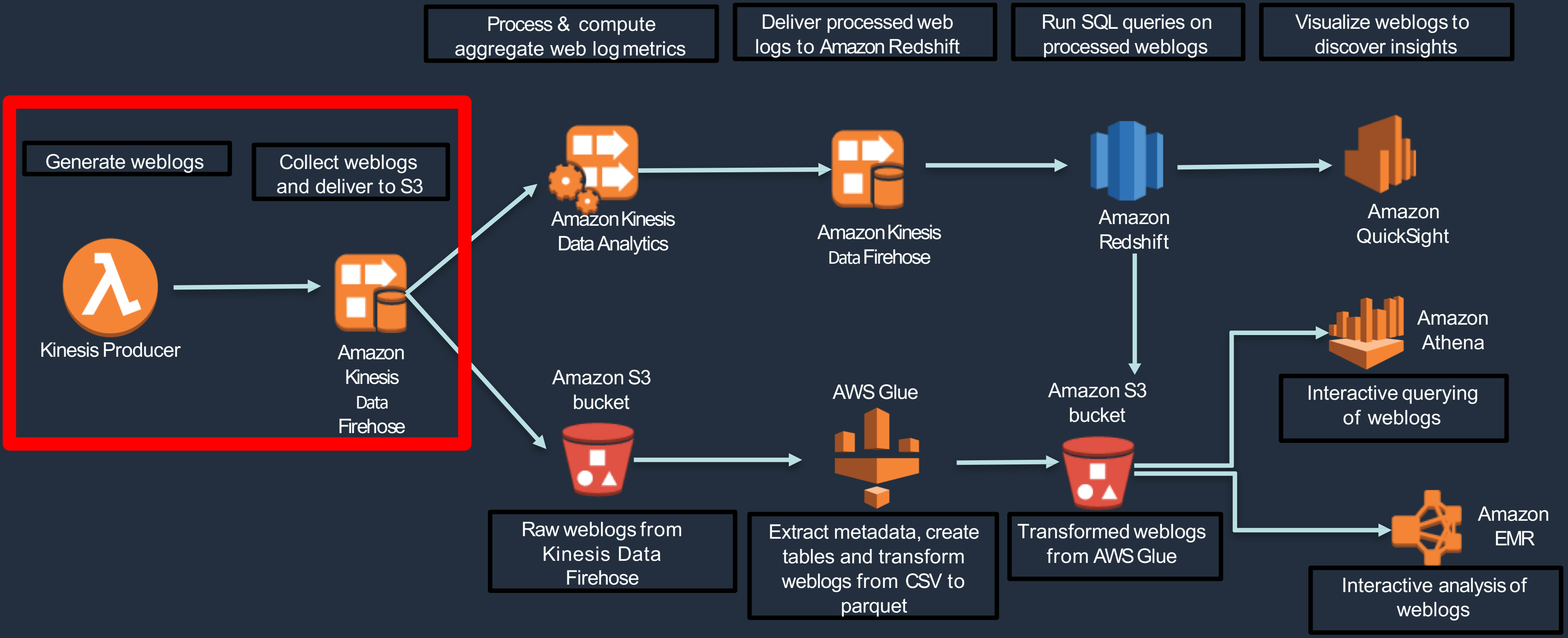
- First big data app workshop activities:
    - 1: Collect logs with Kinesis Data Firehose delivery stream—5 mins
    - 2: Real-time streaming queries using Kinesis Data Analytics—20 mins
    - 3: Deliver streaming results to Amazon Redshift—5 mins
    - 4: Transform weblogs using AWS Glue—30 mins
    - 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



© 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

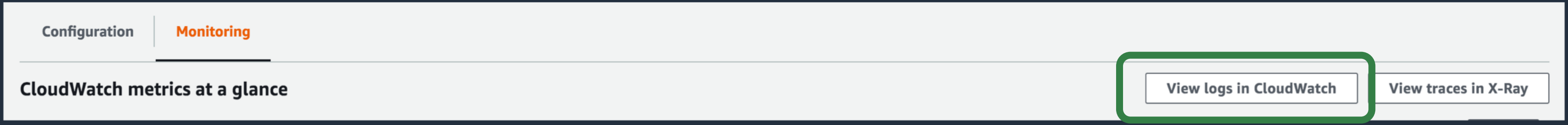
- So that we don’t have to install or set up software on your machine, we are going to use a Lambda function to simulate using the Amazon Kinesis Agent. The Lambda function can populate a Data Firehose delivery stream using a template and is simple to setup.

- **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

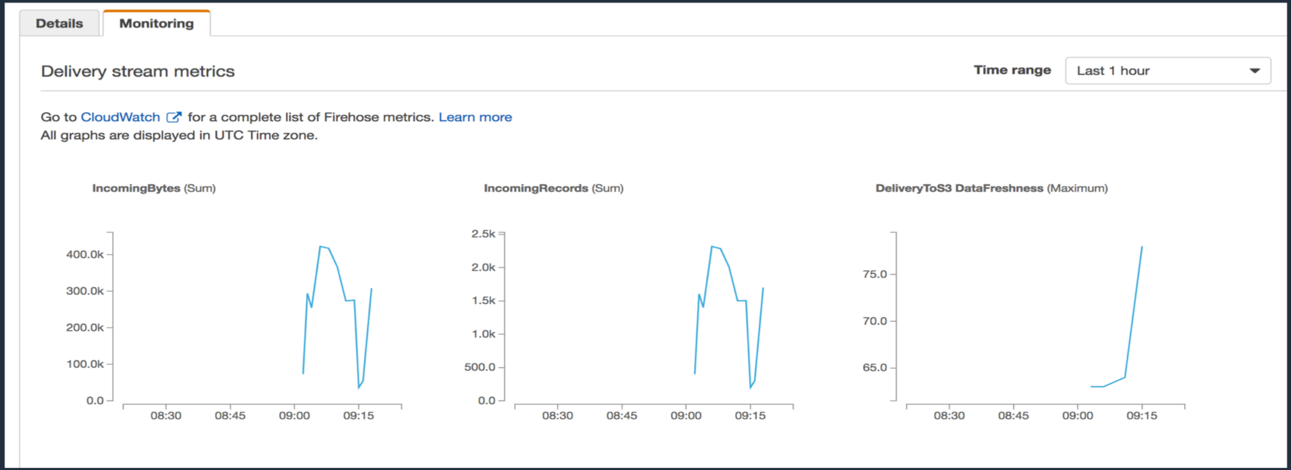


- View the top log stream and you should see Apache log entries like below:

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

Review: Monitoring Kinesis Data Firehose delivery to Amazon S3

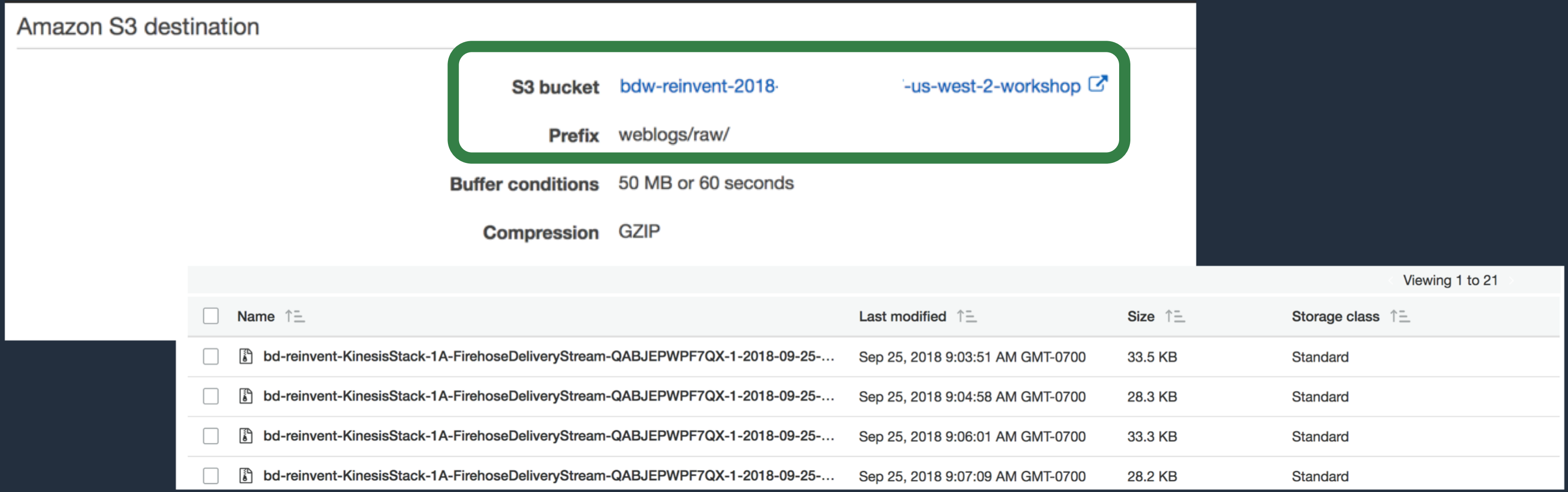
- Go to the Kinesis console, click on “Data Firehose” and find data stream named like:
    - <StackName>-KinesisStack-xxx-FirehoseDeliveryStream-xxxxxxx
- Go to the Monitoring tab and there should be metrics for delivery to Amazon S3. This might take some time to show up.



© 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:

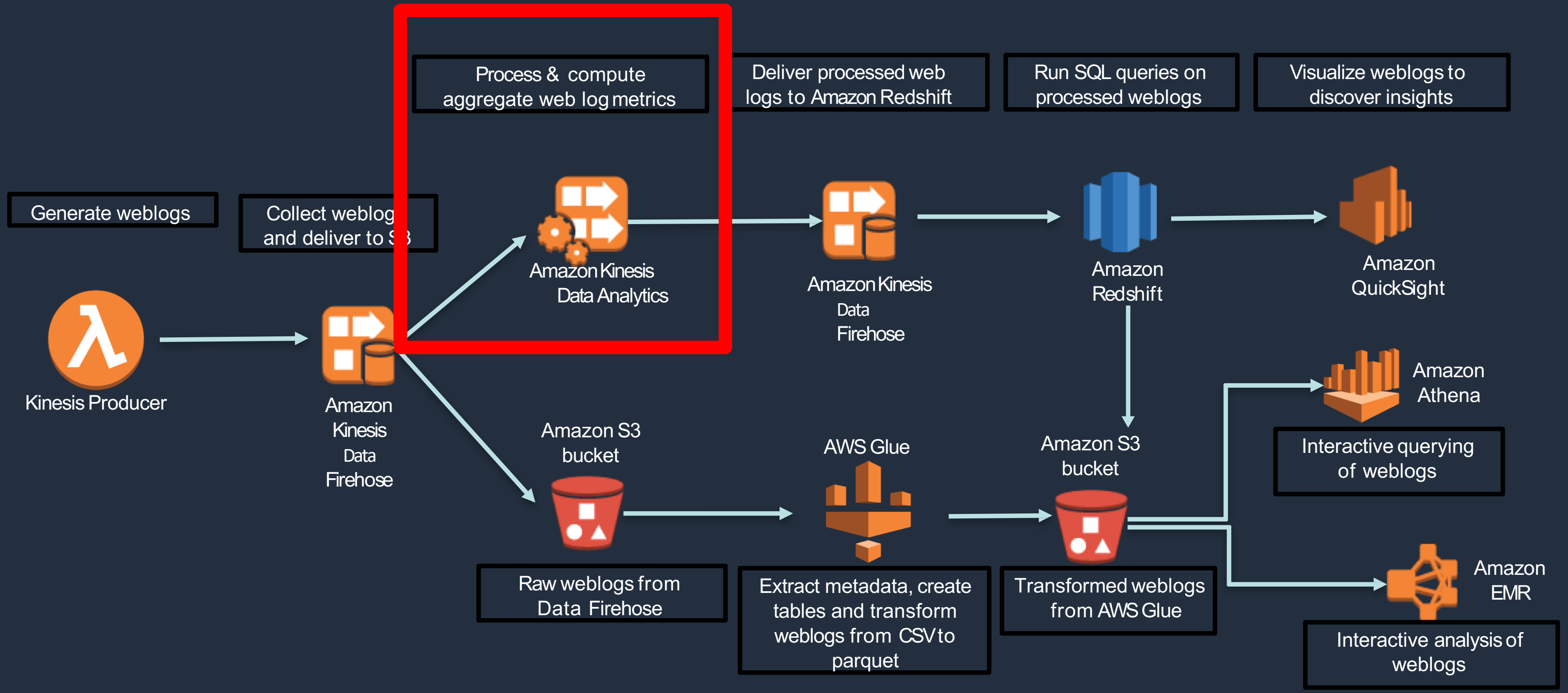


© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

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



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

Process data using Kinesis Data Analytics

Time: 20 minutes

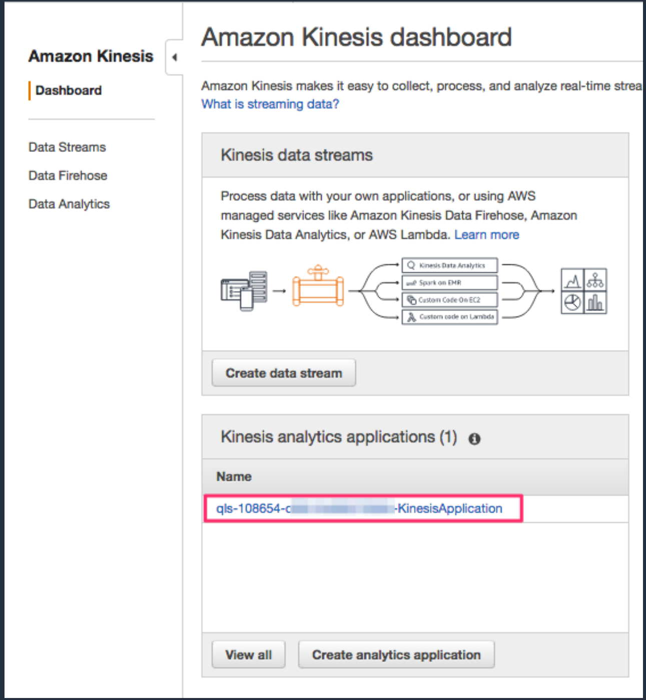
We are going to:

- 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

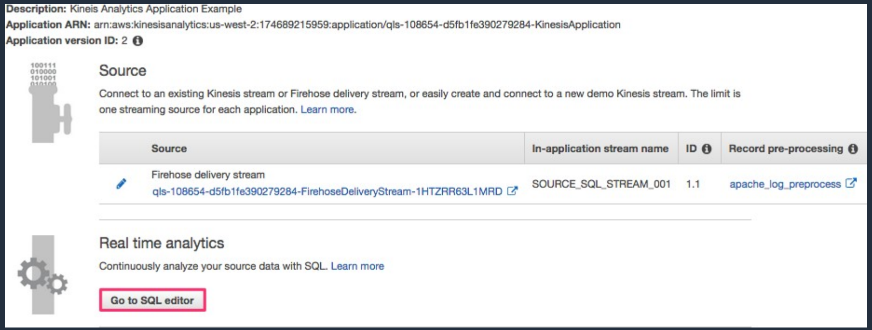
- Navigate to the Kinesis dashboard
- Click on the Kinesis Data Analytics application

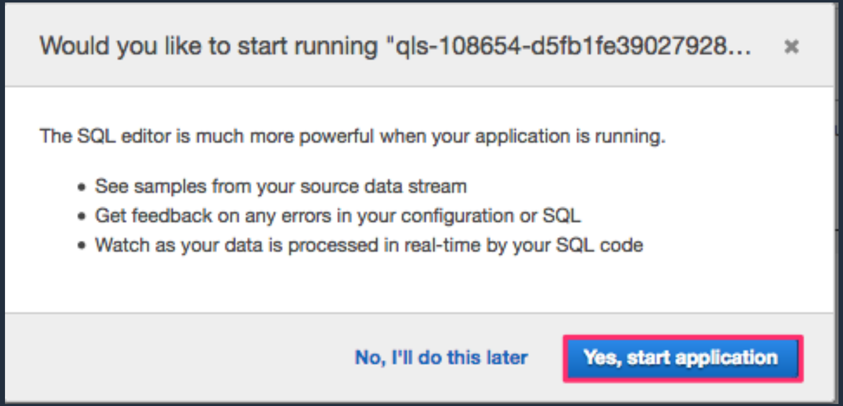


© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Activity 2A: Start Kinesis app

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





© 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)

Filter by column name

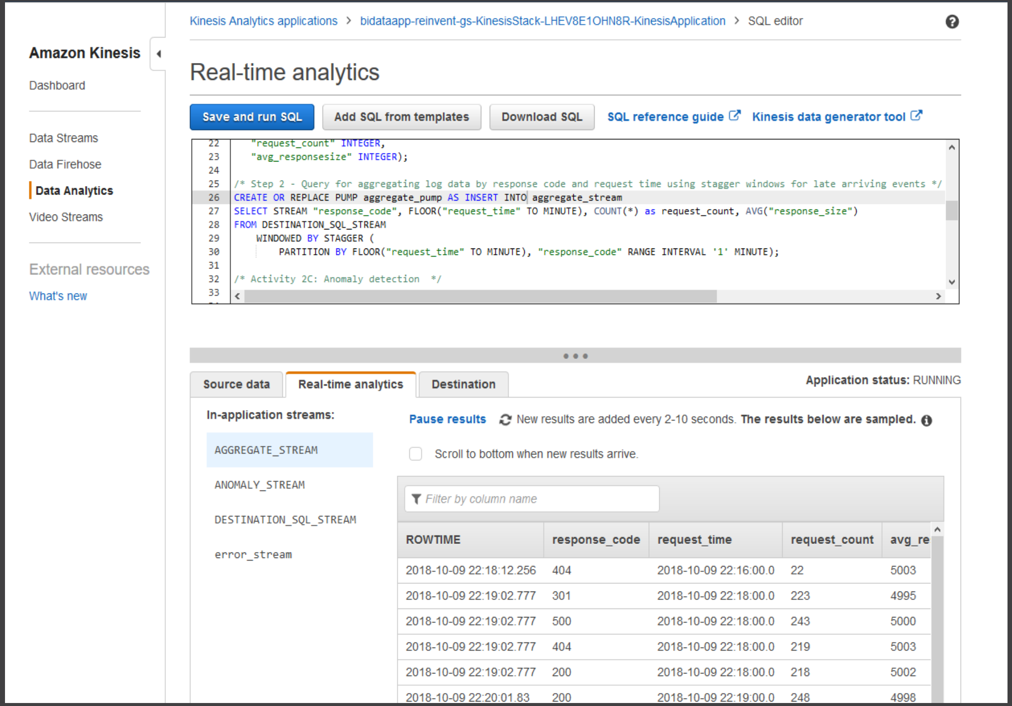
Edit schema

ROWTIME TIMESTAMP	request_method VARCHAR(8)	request_time TIMESTAMP	response_size INTEGER	user_agent VARCHAR(256)	response_code INTEGER	host_address 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	17.199.76.178
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	139.184.6.39
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**"

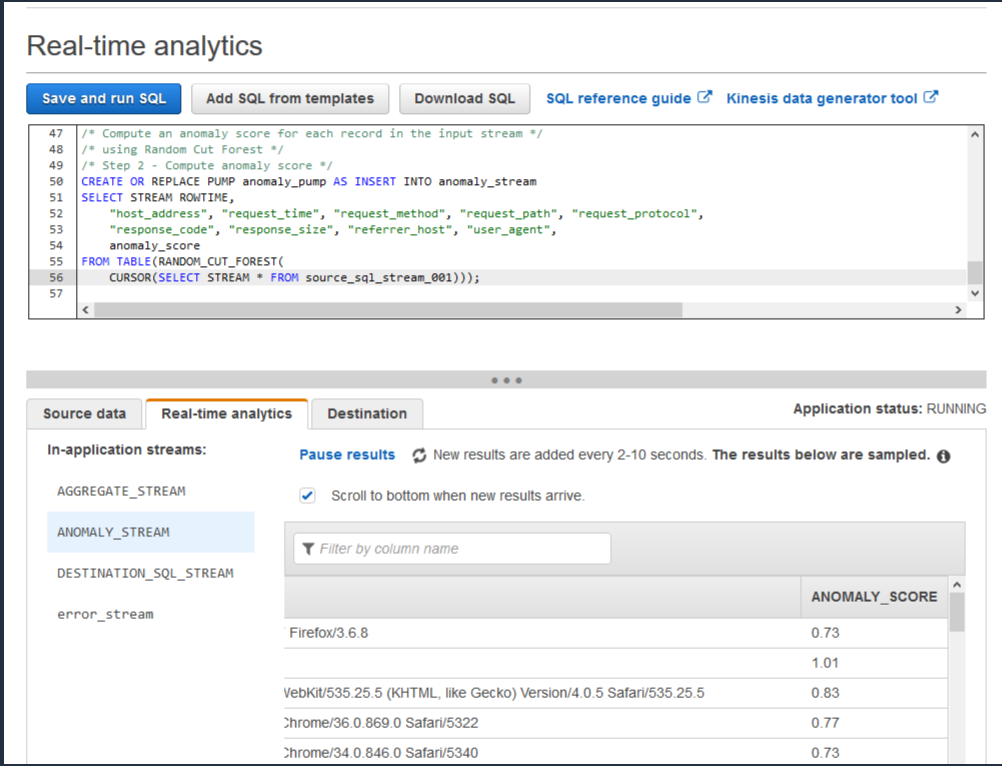


© 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.

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

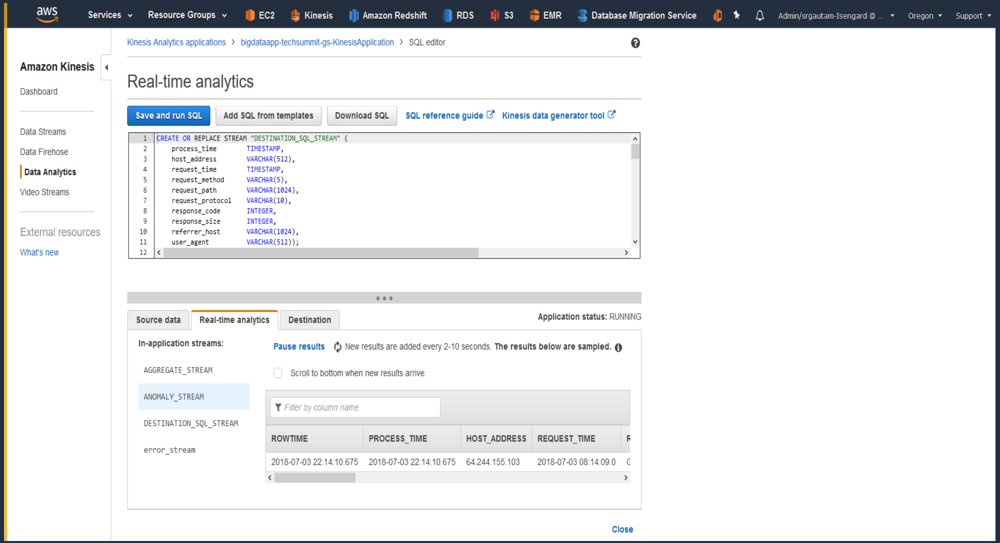


© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Kinesis Data Analytics in-application streams

- In-application streams:

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

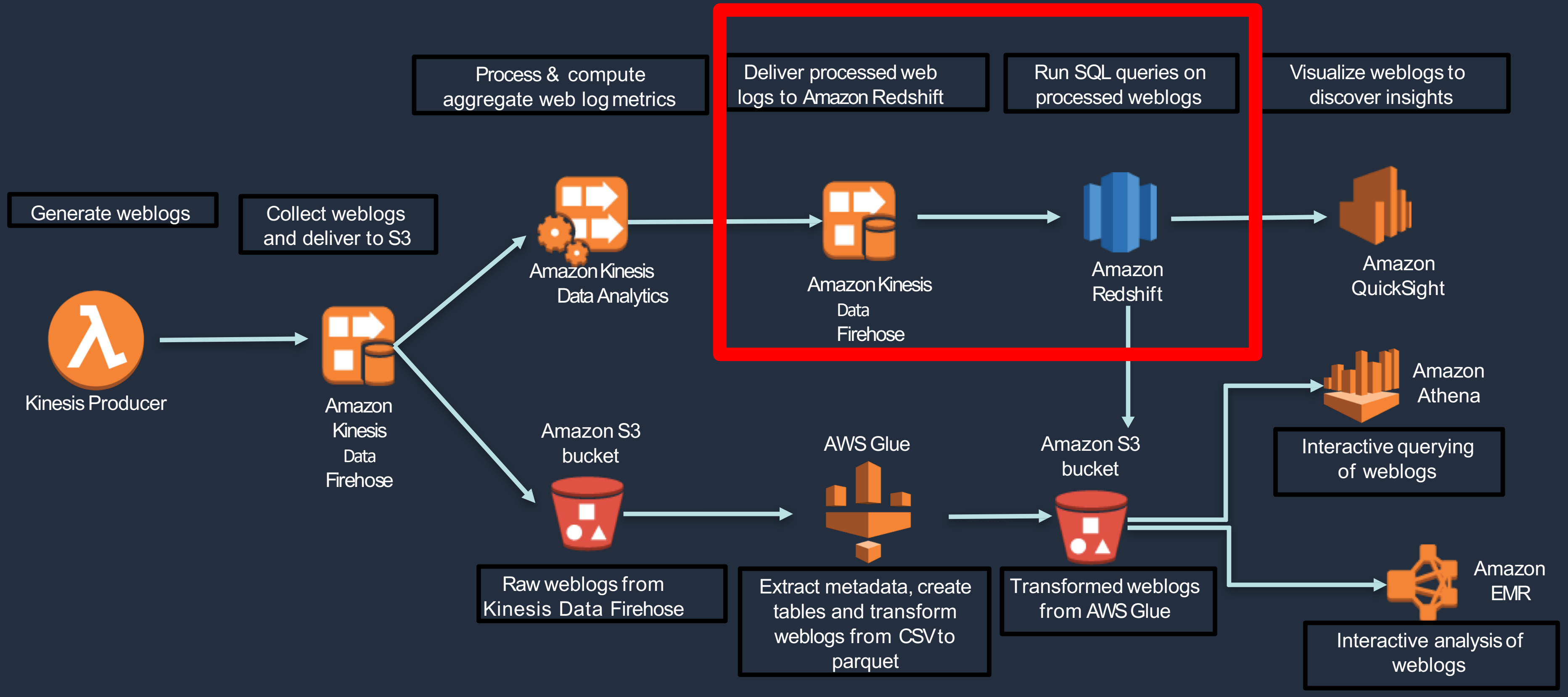


© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Activity 3
Deliver streaming results to Amazon Redshift

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

Your application architecture



© 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

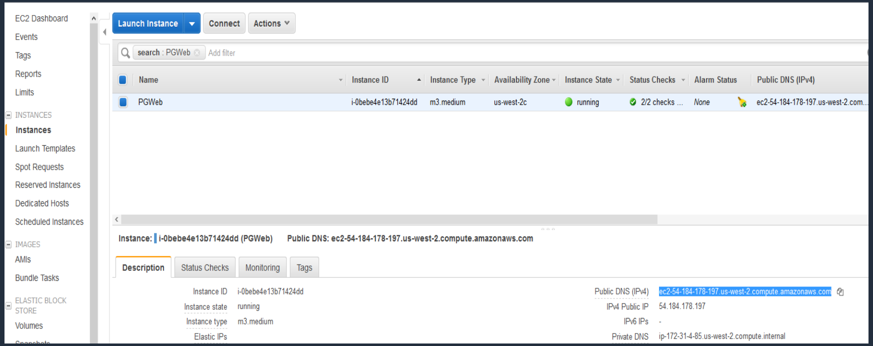
We are going to:

- 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

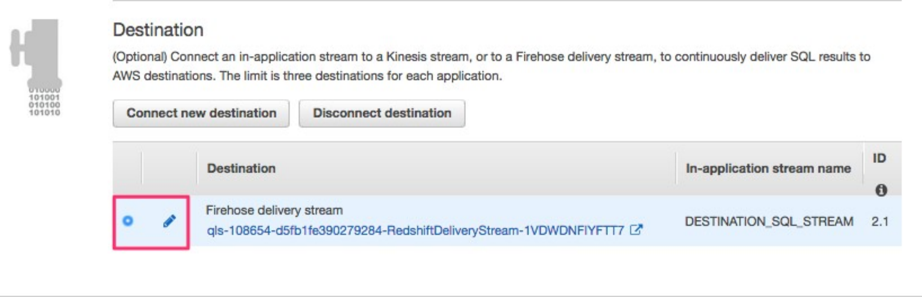
- Make sure you are in the logs database in pgweb
- Create table weblogs to capture incoming data from Kinesis Data Firehose delivery stream
- Use RedshiftSQL.txt from Workshop Scripts for the CREATE TABLE statement
- Paste in the CREATE TABLE SQL statement and click “Run Query” to create the Redshift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

```sql
--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).



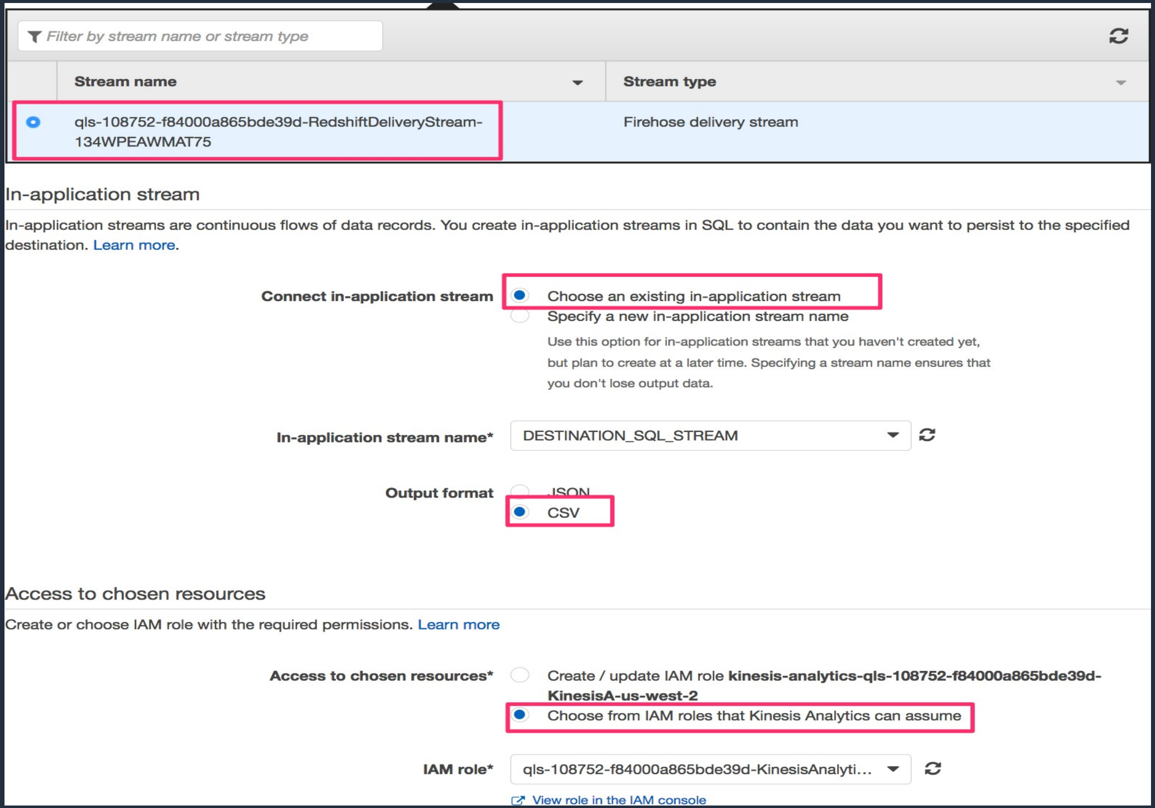
© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

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



© 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

```sql
-- 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

```sql
-- 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"

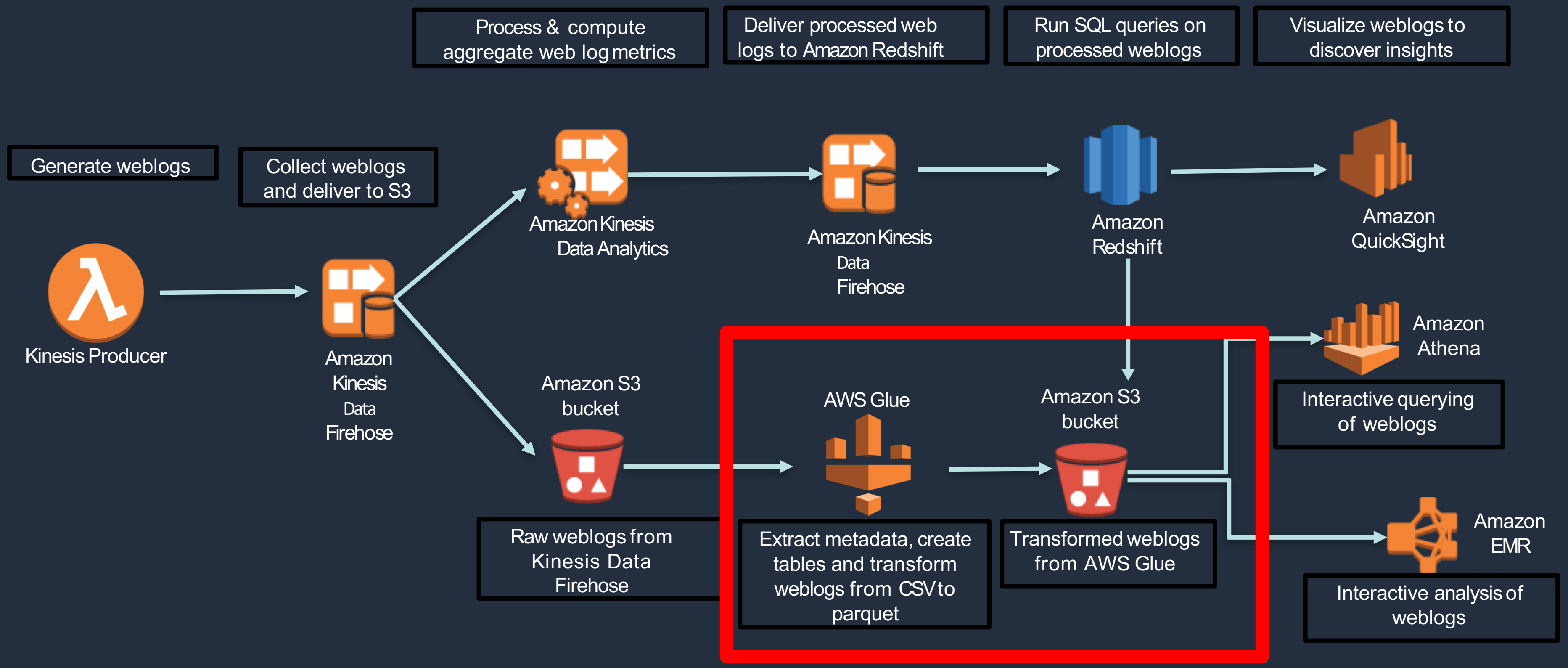
```sql
-- show all requests for status as PAGE NOT FOUND
SELECT TOP 1 request_path, COUNT(1)
FROM weblogs
WHERE response_code = 404
GROUP BY 1
ORDER BY 2 DESC;
```

© 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



© 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:

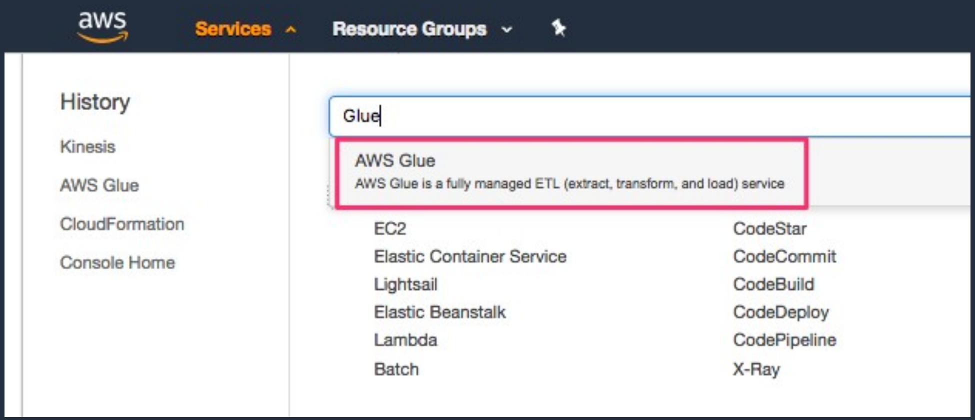
A. Discover and catalog the weblog data deposited into the S3 bucket using AWS Glue crawler

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

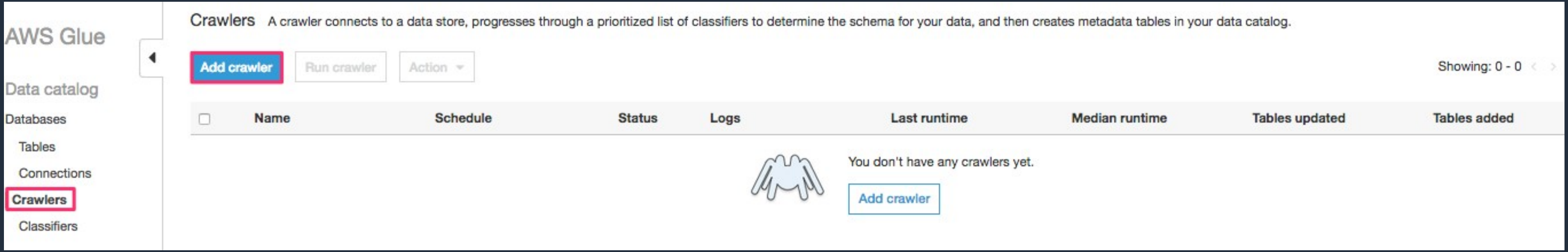
- We use AWS GLUE’s crawler to extract data and metadata. From the AWS Management Console, select AWS Glue. Click on “**Get Started”** on the next screen.



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

Activity 4A: Add crawler using AWS Glue

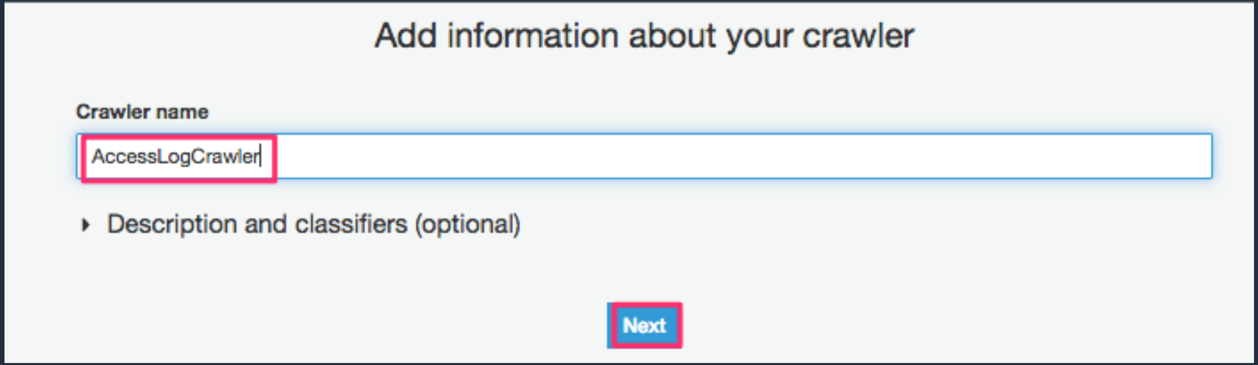
- Select "**Crawlers"** section on the left and click on **"Add crawler"**



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

Activity 4A: ETL with AWS Glue

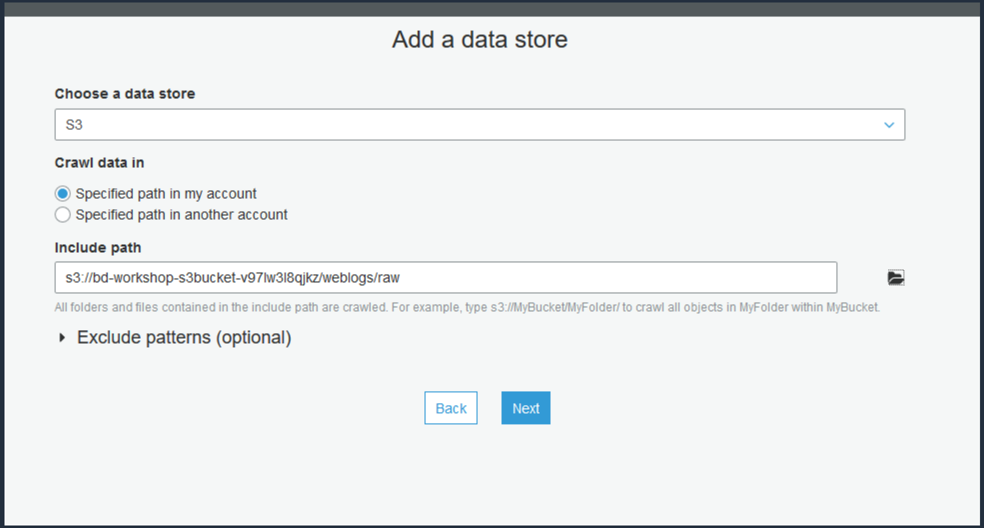
- Specify a name for the crawler. Click "**Next**"

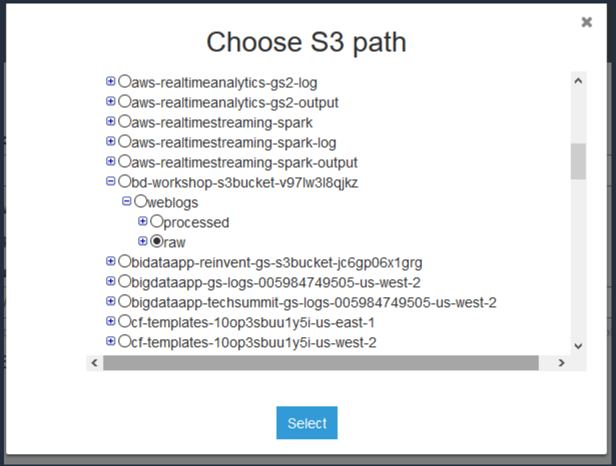


© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

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**"

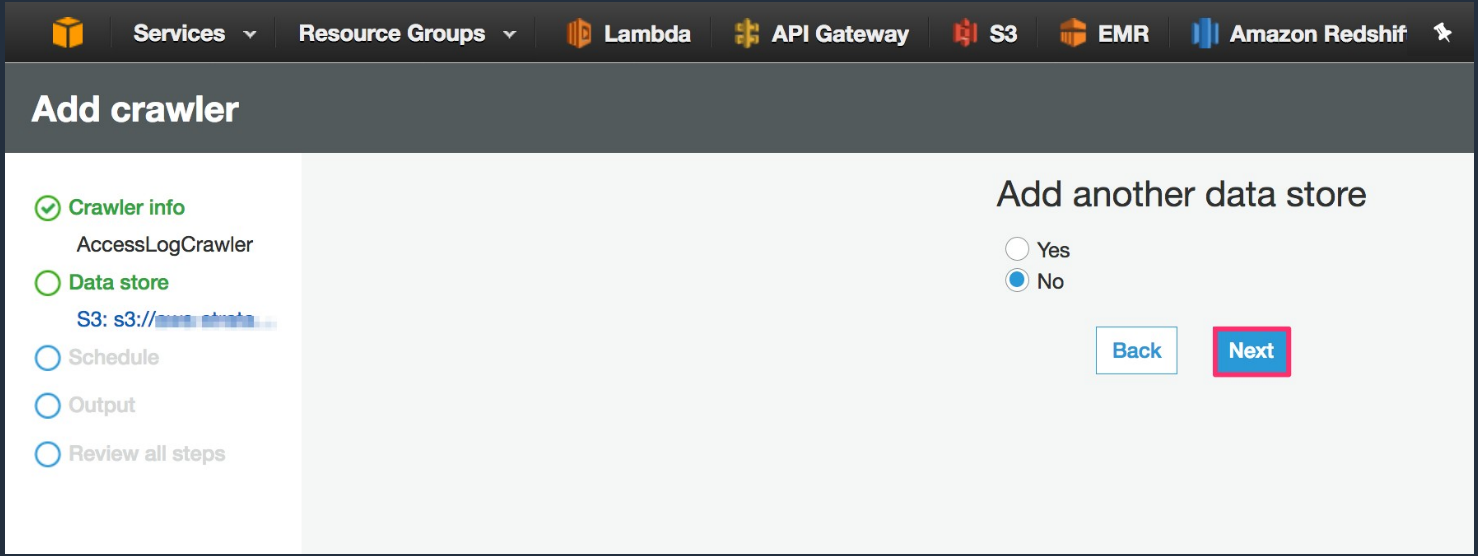




© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Activity 4A: ETL with AWS Glue

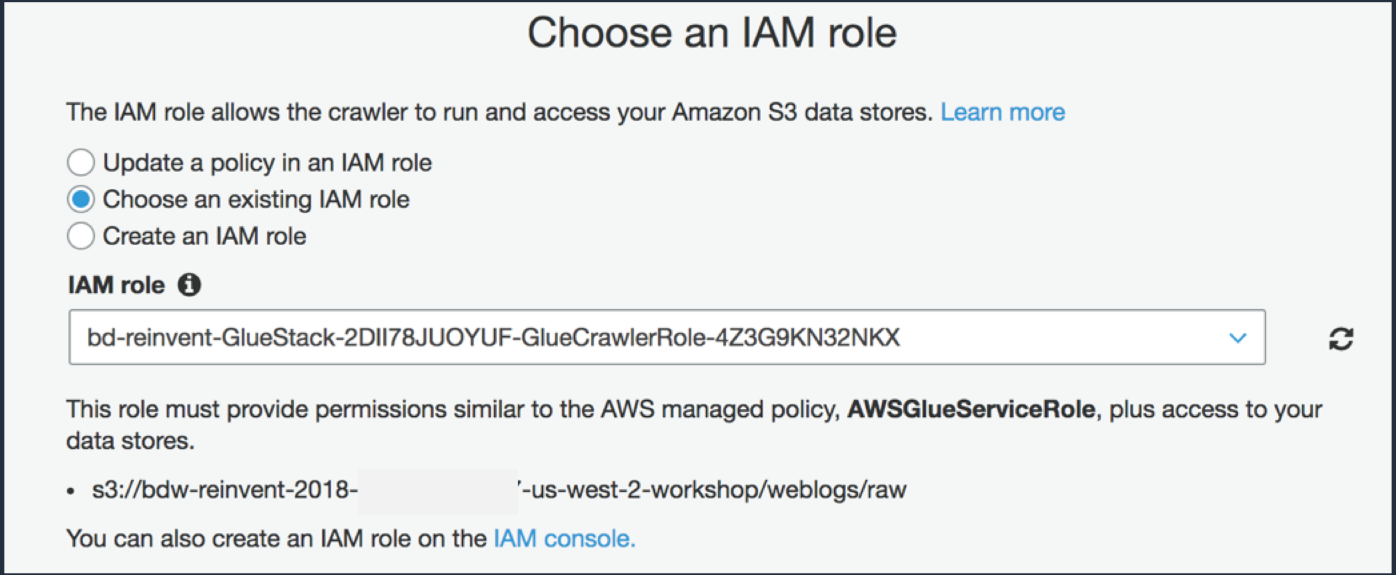
- Click "**Next**" on the next screen to not add another data store



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

Activity 4A: ETL with AWS Glue

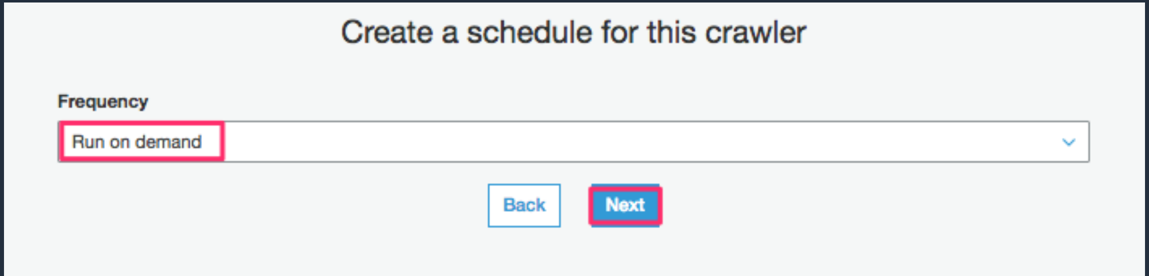
- In the IAM role section, select “Choose an existing IAM role”
- Select role <StackName>-GlueStack-xxx-GlueCrawlerRole-xxxxxxx as the IAM role and click “Next”



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

Activity 4A: Add crawler with AWS Glue

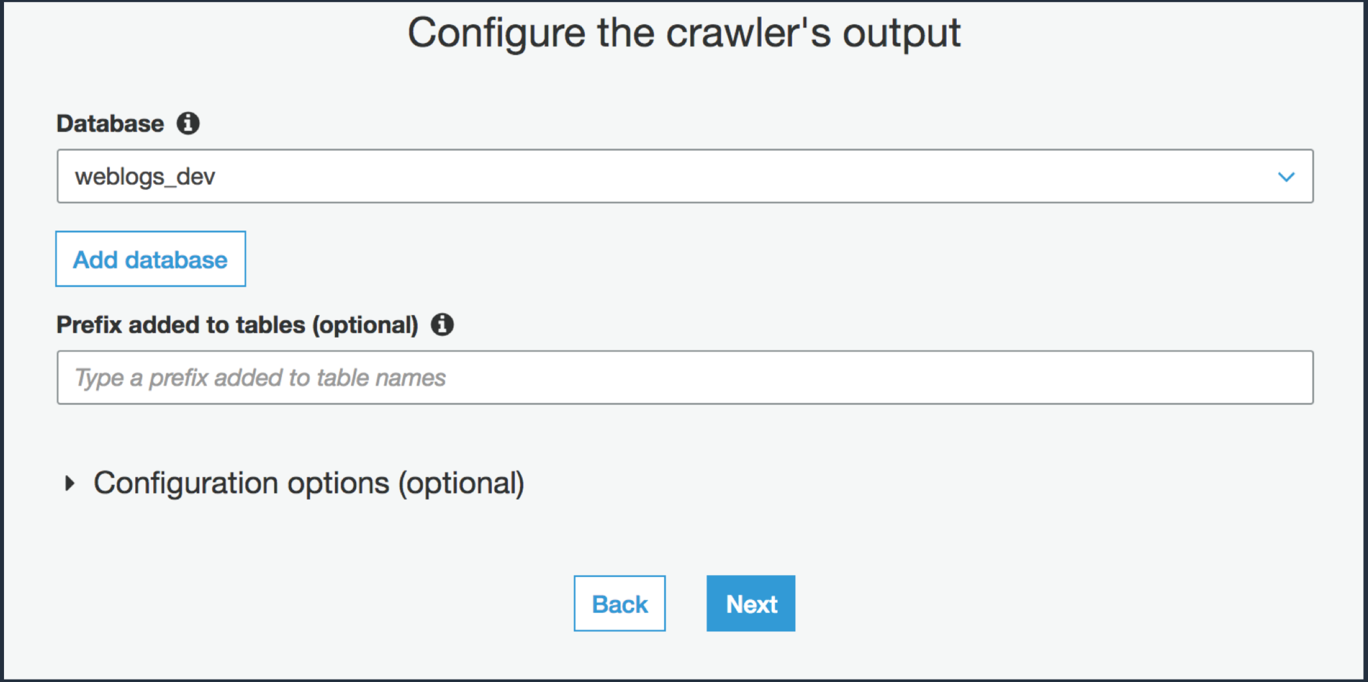
- Choose "**Run on demand**" to run the crawler now, and click "**Next**"



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

Activity 4A: Add crawler with AWS Glue

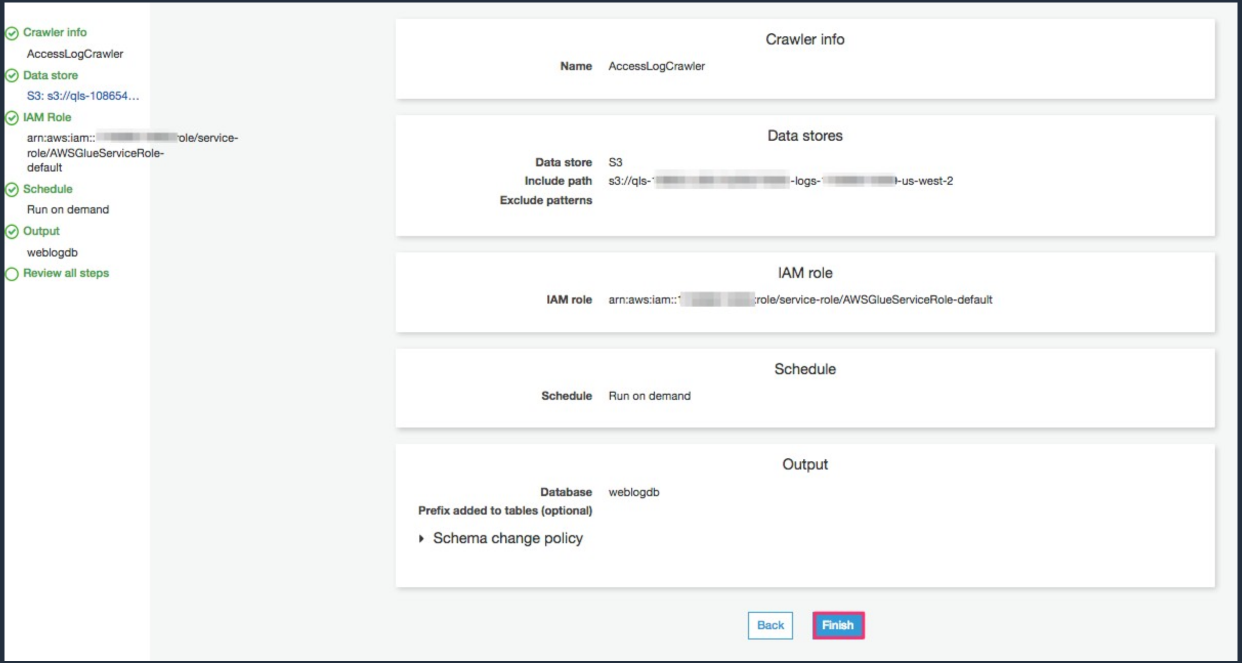
- On the next screen, drop down Database and select "weblogs_dev"
- Click "Next"



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

Activity 4A: Add crawler with AWS Glue

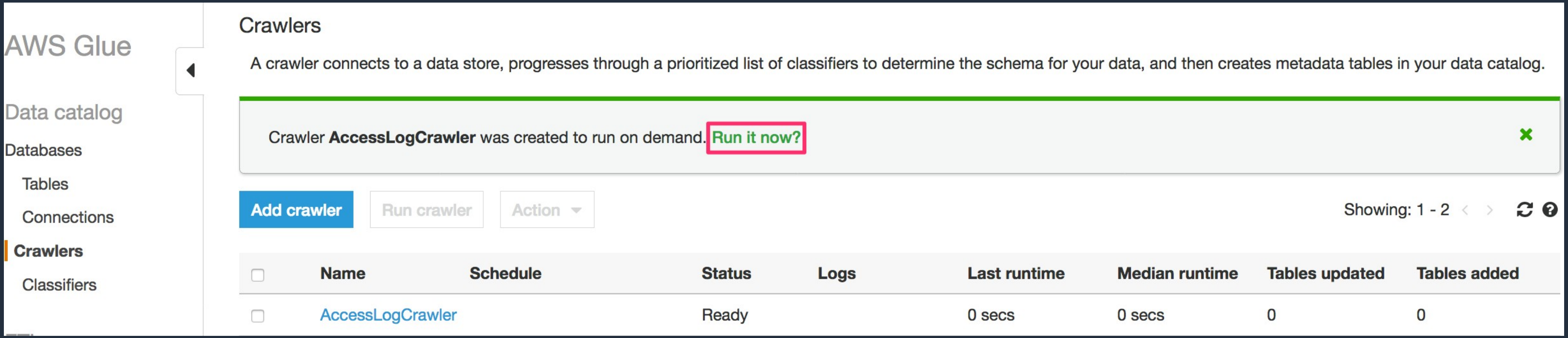
- Review and click "**Finish**" to create a crawler



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

Activity 4A: Add crawler with AWS Glue

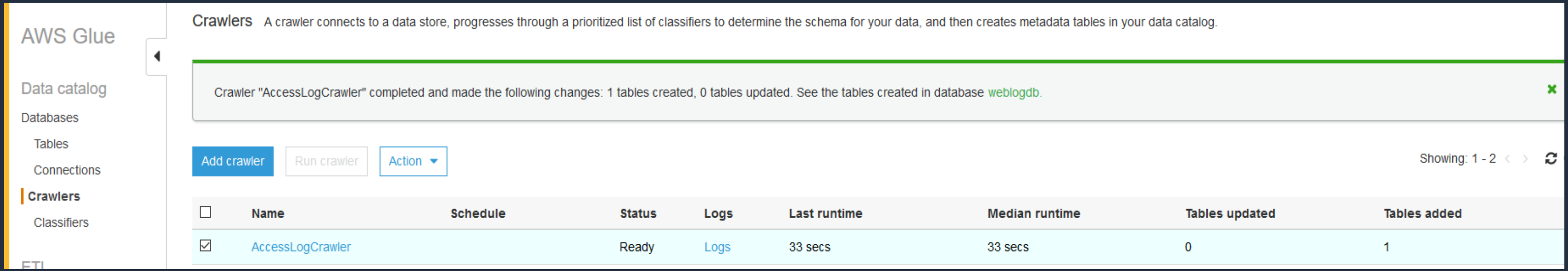
- Click on "**Run it now?**" link to run the crawler



© 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



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

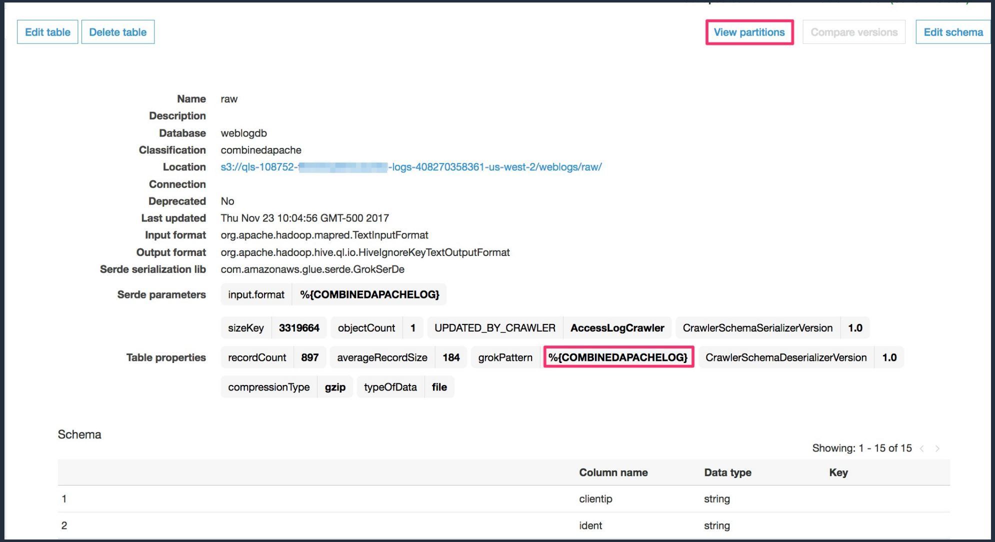
Activity 4A: Table creation in AWS Glue

- 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

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

Activity 4A: Table creation in AWS Glue

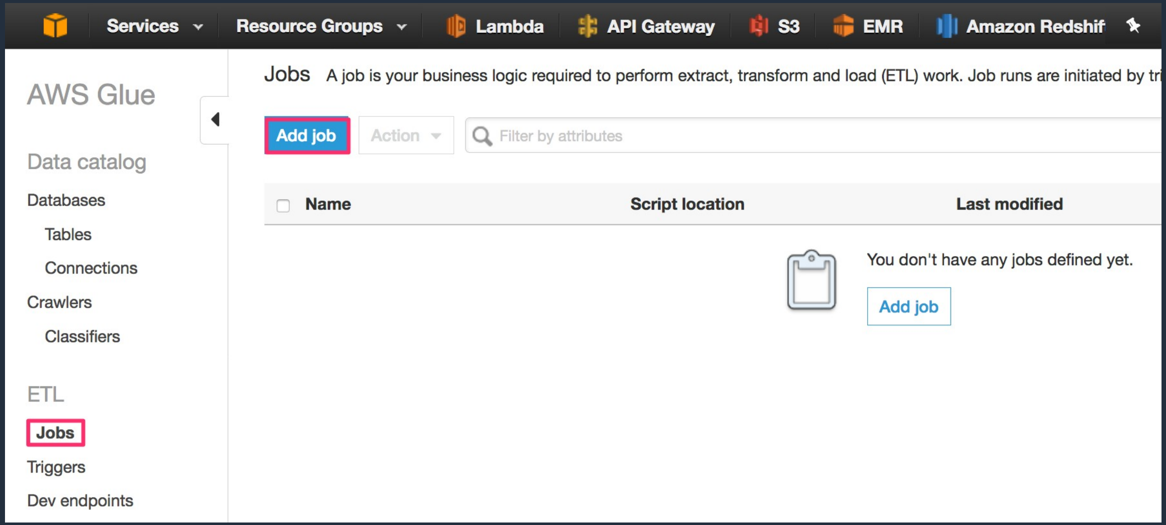
- AWS Glue used the **GrokSerDe** (Serializer/Deserializer) to correctly interpret the weblogs
- You can click on the "**View partitions**" link to look at the partitions in the dataset



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

Activity 4A: Create ETL job in AWS Glue

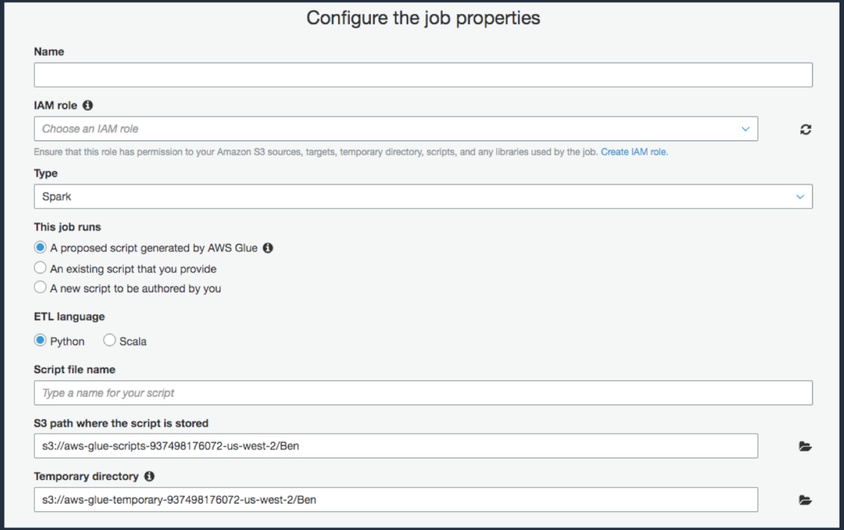
- With the dataset cataloged and table created, we are now ready to convert the weblogs–based Apache combined log format to a more optimal parquet format for querying.
- Click on "**Add job**" to begin creating the ETL job



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

Activity 4B: ETL job in AWS Glue

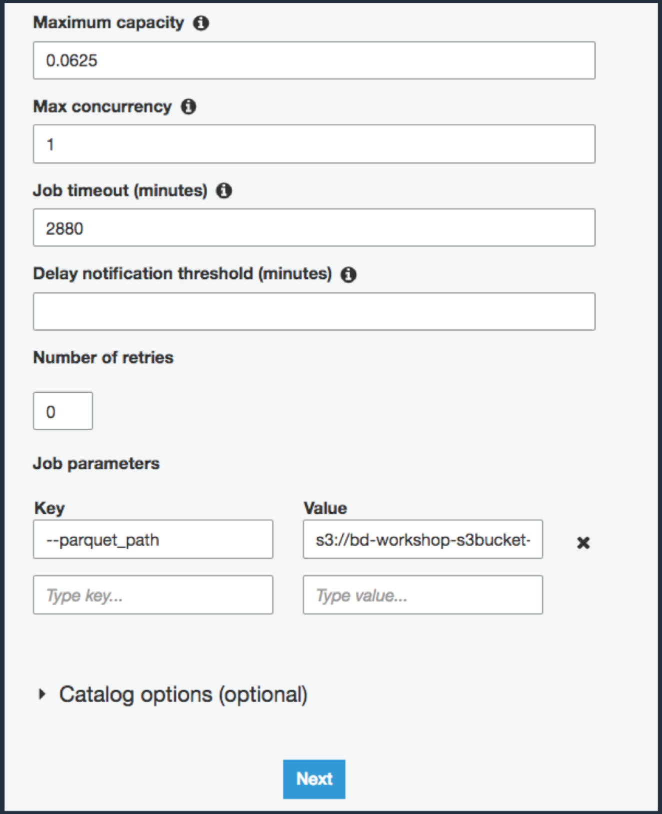
- Job name: **accesslogetl1**
- Select the IAM role <StackName>-GlueStack-xxx-GlueJobRole-xxxxxxx
- Select Type as Spark
- Select "a new script to be authored by you"
  Script file name: **glue-workshop-etl.py**
- For the S3 path where the script will be stored, use path **s3://<S3 bucket from template>/script**
- For the Temporary Directory, use path **s3://<S3 bucket from template>/temp**
- **DO NOT CLICK NEXT JUST YET**



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

Activity 4B: ETL job in AWS Glue

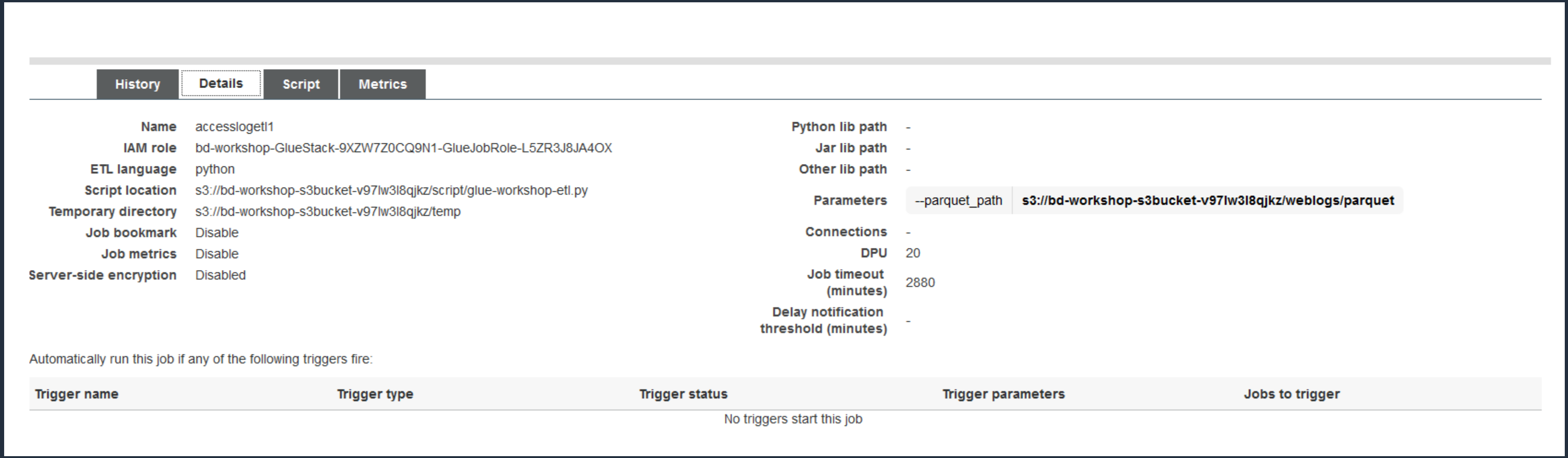
- Expand **Script libraries and job parameters** section, and set the Maximum capacity (DPUs) to 10.
- Let's pass a job parameter to send the S3 path where parquet files will be deposited.
- Specify the following values for Key and Value
- **Key: --parquet_path** (notice the 2 hyphens at the beginning 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



© 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



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

Activity 4B: ETL job in AWS Glue

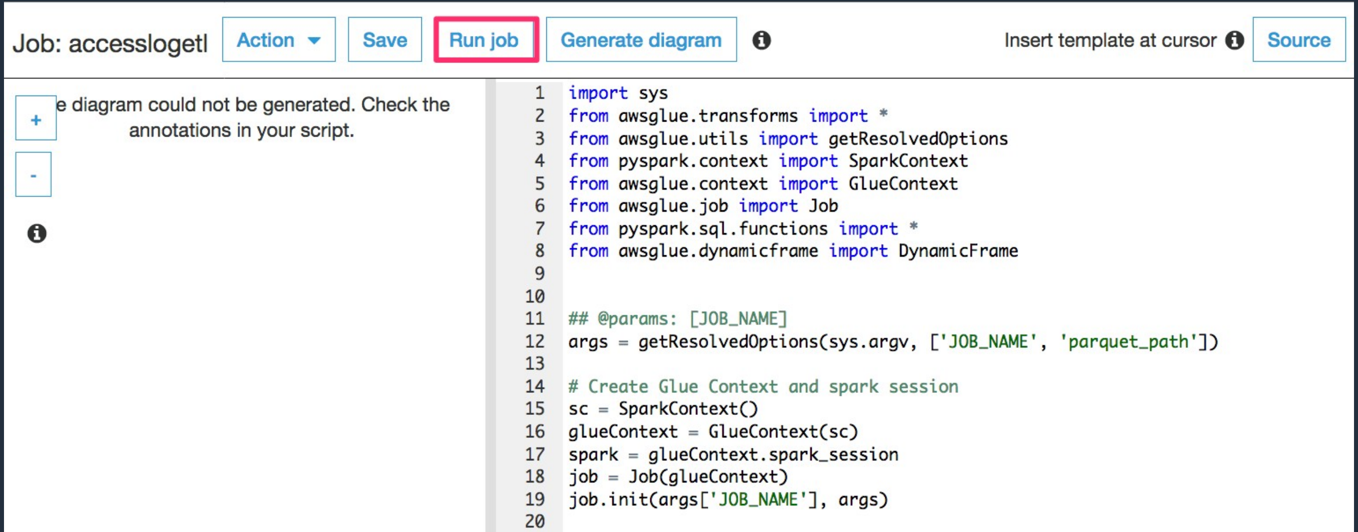
```
import sys
from awsglue.transforms import *
from awsglue.utils import getResolvedOptions
from pyspark.context import SparkContext
from awsglue.context import GlueContext
from awsglue.job import Job
from pyspark.sql.functions import *
from awsglue.dynamicframe import DynamicFrame


## @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 = "weblog_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, transformation_ctx = "datasource0")

# 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, df.bytes, df.referrer, df.agent)

# Cast the datetime column to timestamp
#ts_df = new_df.withColumn("dt", unix_timestamp(df.timestamp, "yyyy-MM-dd HH:mm:ss").cast('timestamp').cast("double").cast("timestamp")).drop(df.timestamp
ts_df = new_df.withColumn("dt", unix_timestamp(df.timestamp, "yyyy-MM-dd HH:mm:ss").cast("timestamp")).drop(df.timestamp)
```

- 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

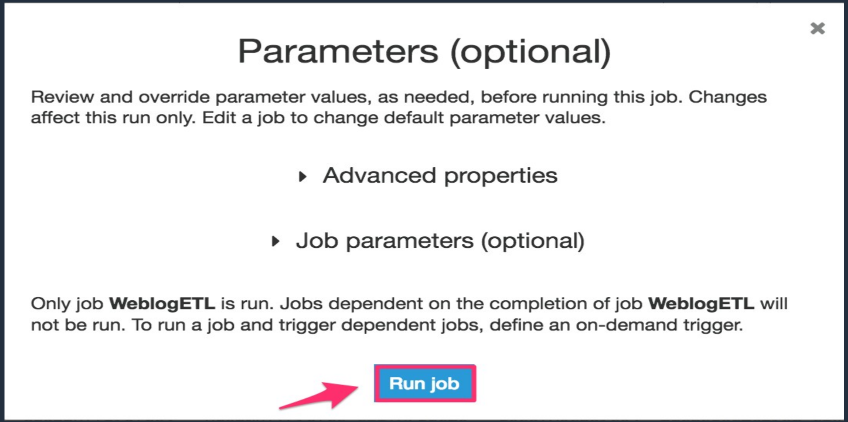
- Click "**Save**" and then "**Run job**" button to execute your ETL



© 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 script name is 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.

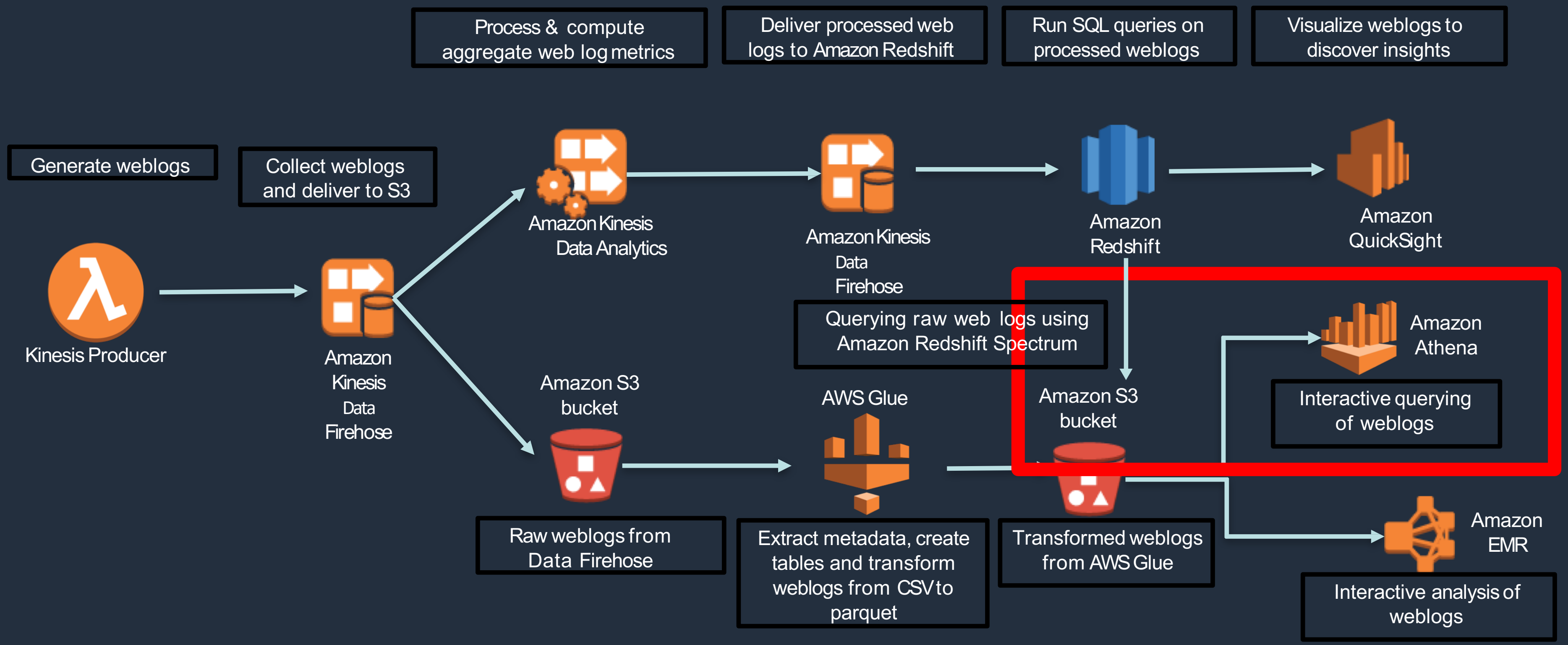


© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

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



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

Activity: Querying data in Amazon S3 using Redshift Spectrum

Time: 30 minutes

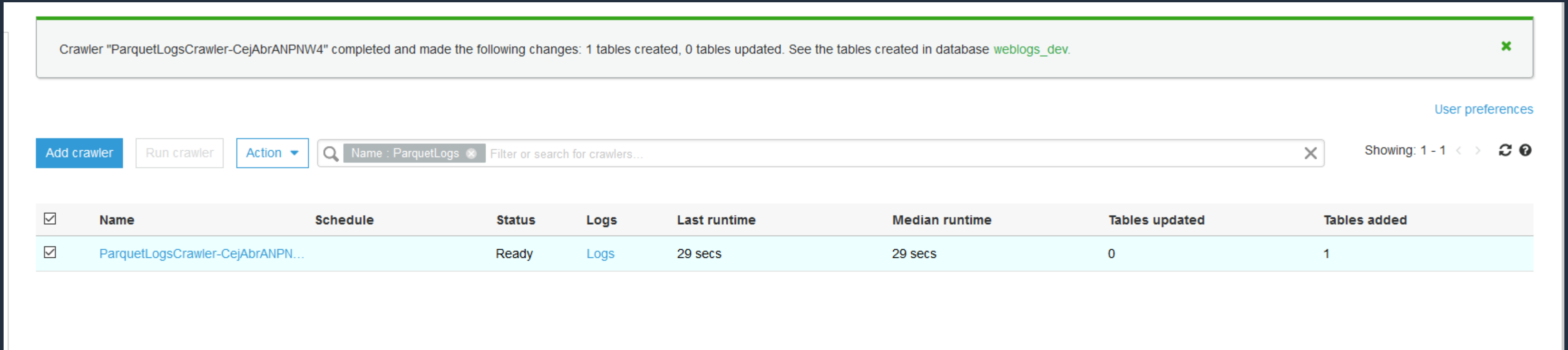
We are going to:

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.



© 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.

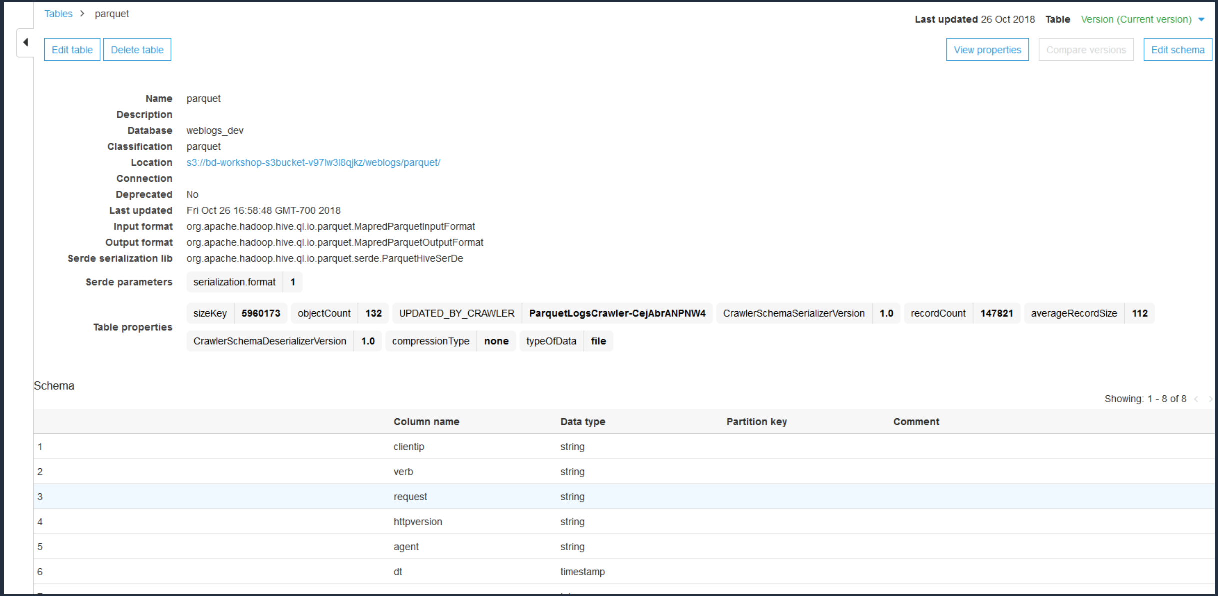
Add tables | Action | search : weblogs_dev | Filter or search for tables... | Save view | Showing: 1 - 2

Name	Database	Location	Classification	Last updated	Deprecated
parquet	weblogs_dev	s3://bd-workshop-s3bucket-v97lw3l8qjkz/weblogs/parquet/	parquet	26 October 2018 4:58 PM UTC-7	
raw	weblogs_dev	s3://bd-workshop-s3bucket-v97lw3l8qjkz/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

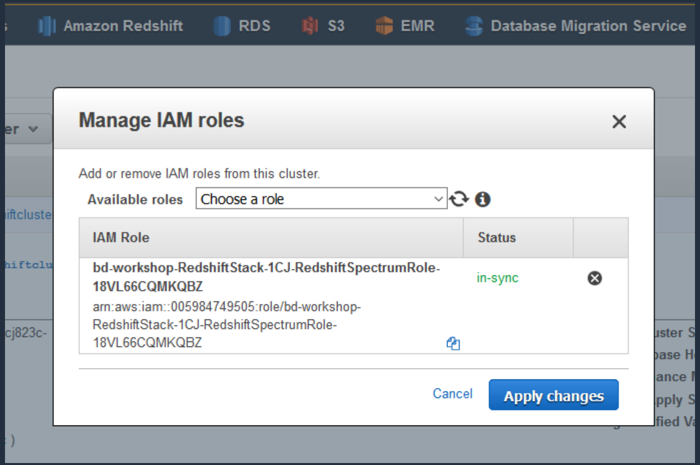
- Take a look at the schema of the parquet table
- Should have eight columns
- Classification should be parquet



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

Activity 5B: Query using Amazon Redshift Spectrum

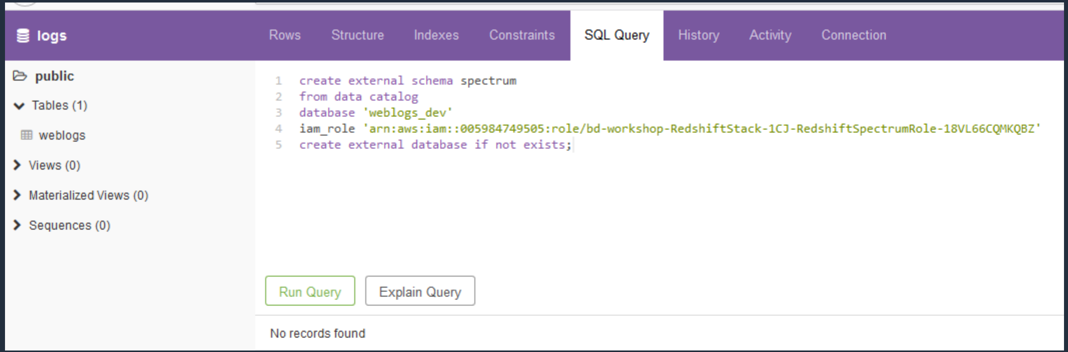
- Navigate to Amazon Redshift-Clusters. Select the cluster with the <stack-name>
- Click on “Manage IAM roles”.
- Validate that IAM role shows bd-workshop-RedshiftStack-xxx-RedshiftSpectrumRole-xxxxxxxxxxx.



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

Activity 5B: Query using Amazon Redshift Spectrum

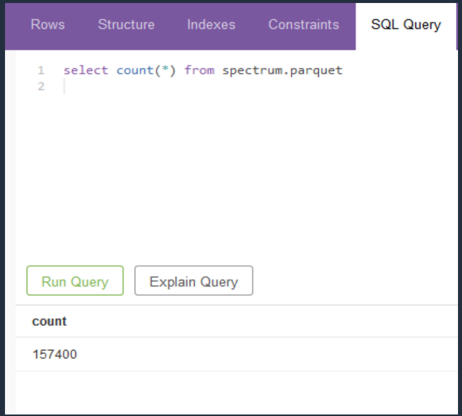
- Navigate to pgWeb. If you need the URL, check public IP of the pgweb EC2.
- Get the RedshiftSpectrumSetup.sql from the BigDataWorkshop.zip.
    - Make sure you replace the role ARN with your SpectrumRole ARN.
    - Make sure you are pointing to the weblogs_dev database.
    - This will create an external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

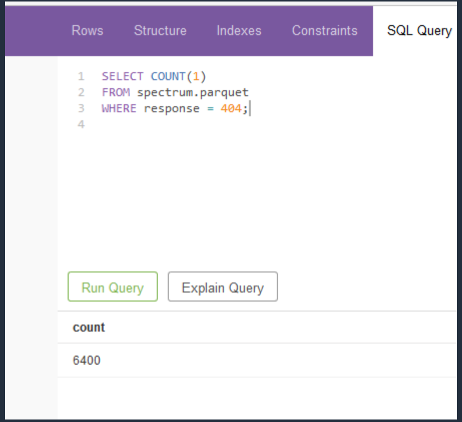


© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

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;

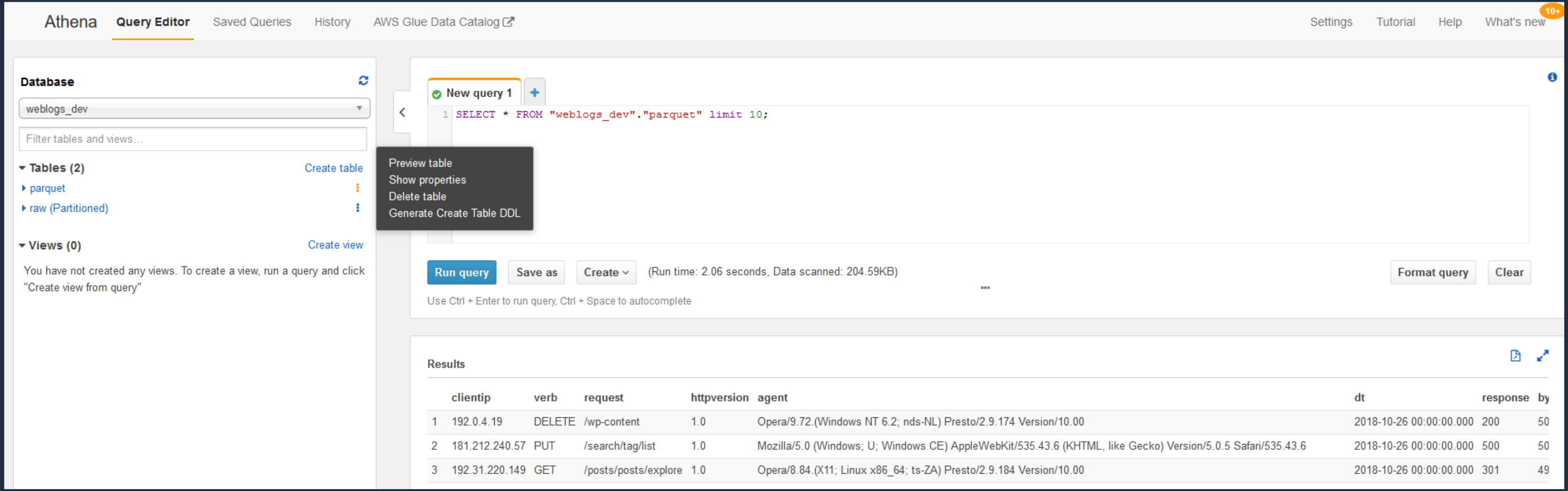




© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Activity 5C: Querying using Amazon Athena

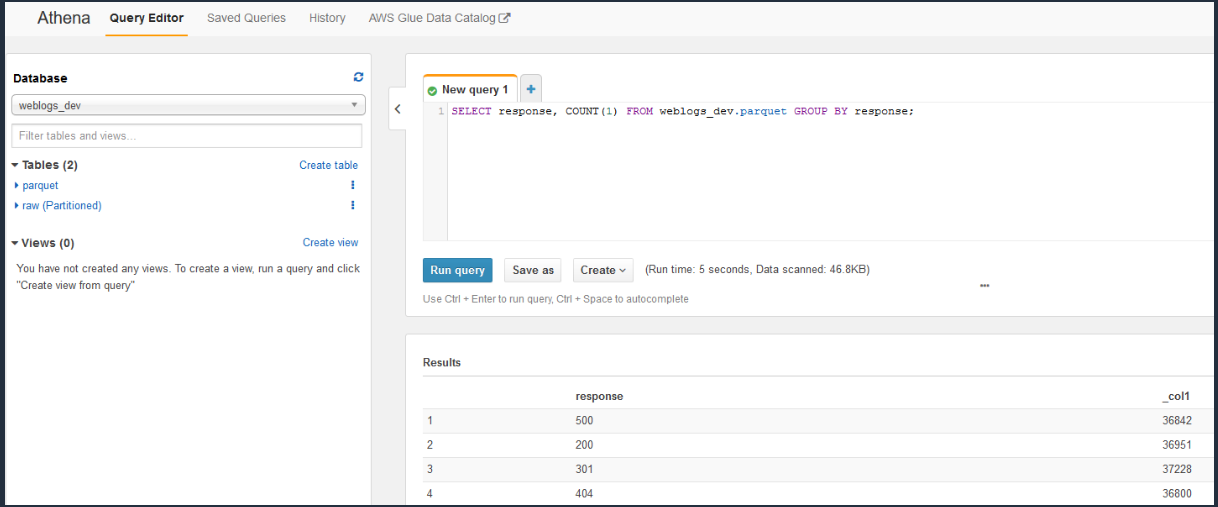
- Go to the Athena console. Close the tutorial that comes up.
- In the Athena console, choose the "**weblogs_<environment>**" on the database dropdown.
- Select "**parquet**“ from the tables section and click on the **three stacked dots** icon to preview/sample a few rows of the S3 data.



© 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



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

Congratulations!

and

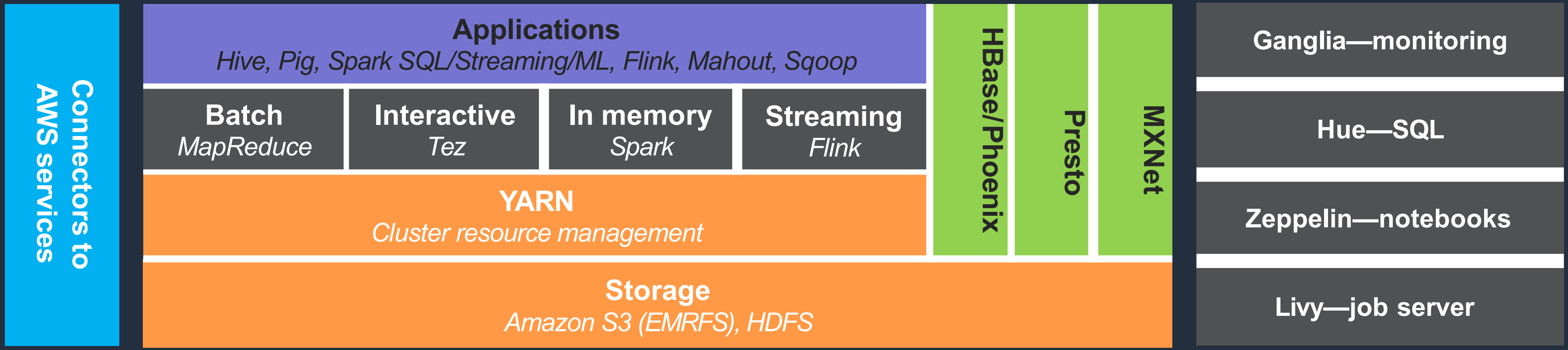
Thank you for attending !

© 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



Amazon EMR service

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

On-cluster UIs

SQL editor, workflow designer, metastore browser

Notebooks

Design and execute queries and workloads

Manage applications

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

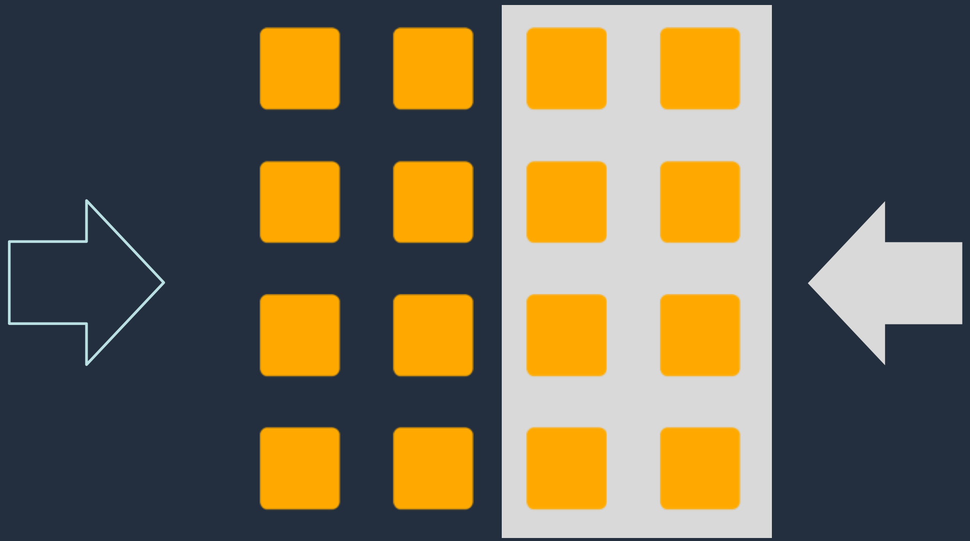
mahout

© 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



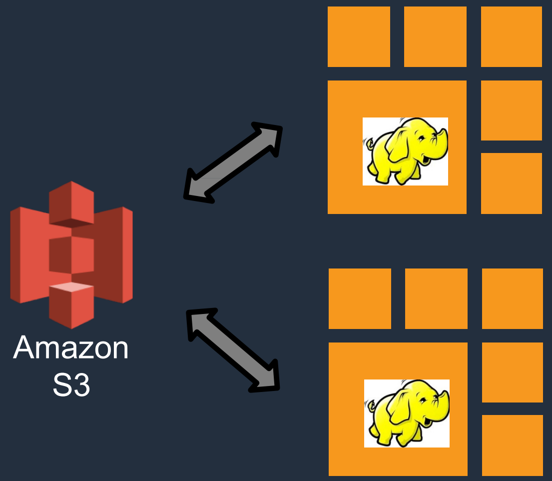
Spot Instances for task nodes

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

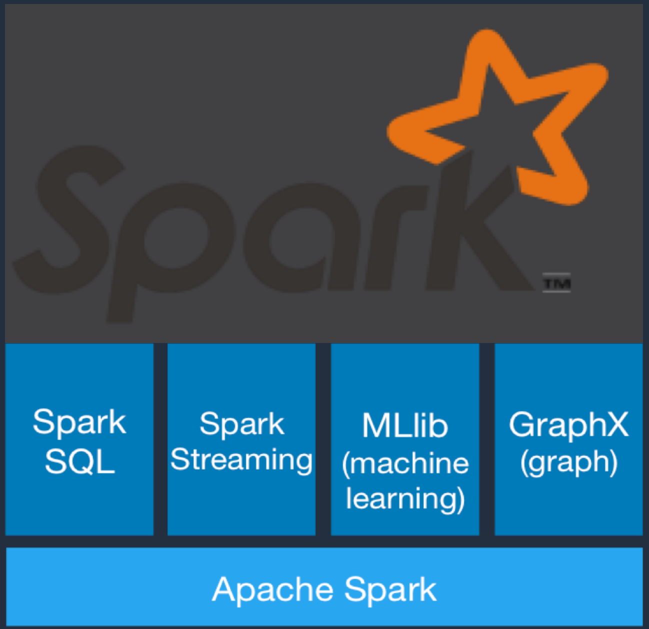
EMRFS makes it easier to leverage Amazon S3

- 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
```

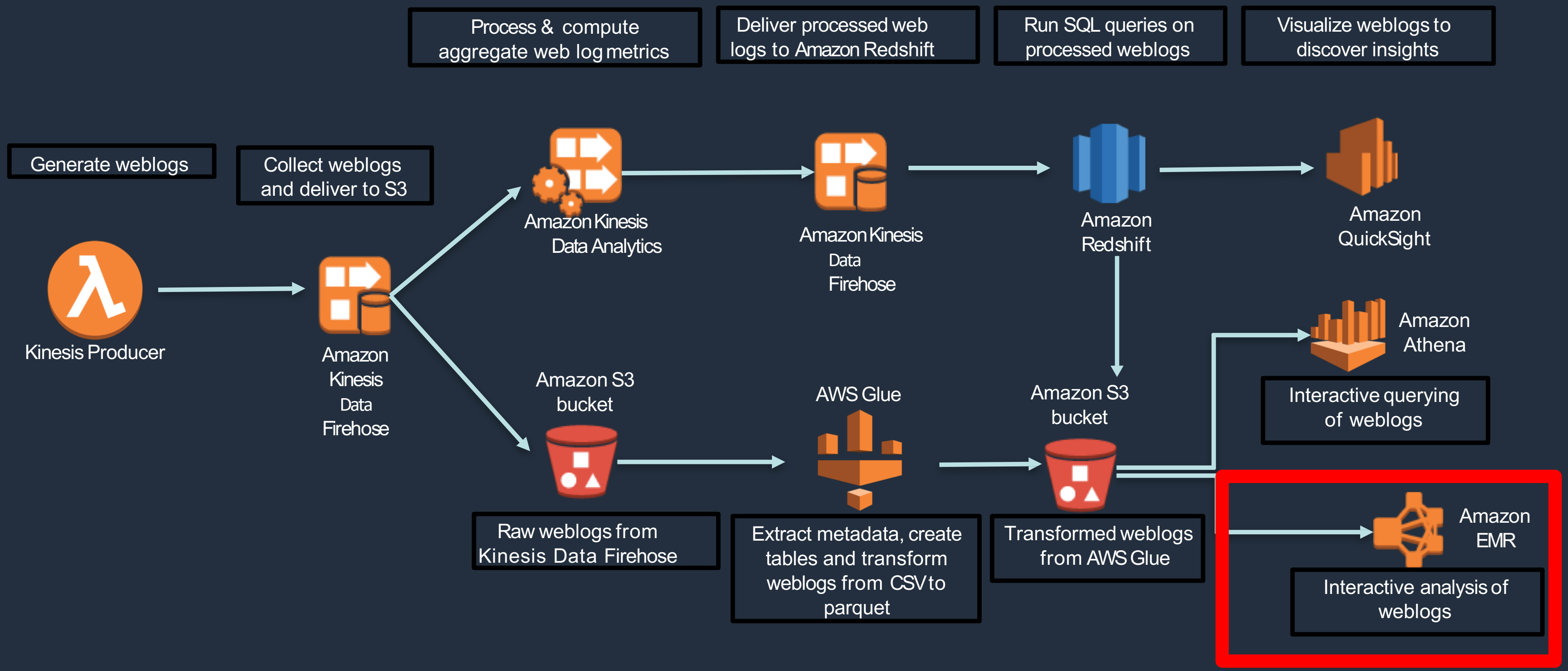
```
%sql -- built-in SparkSQL Support
select * from RDD
```

© 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



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

Activity 6: Process and query data with Amazon EMR

Time: 20 minutes

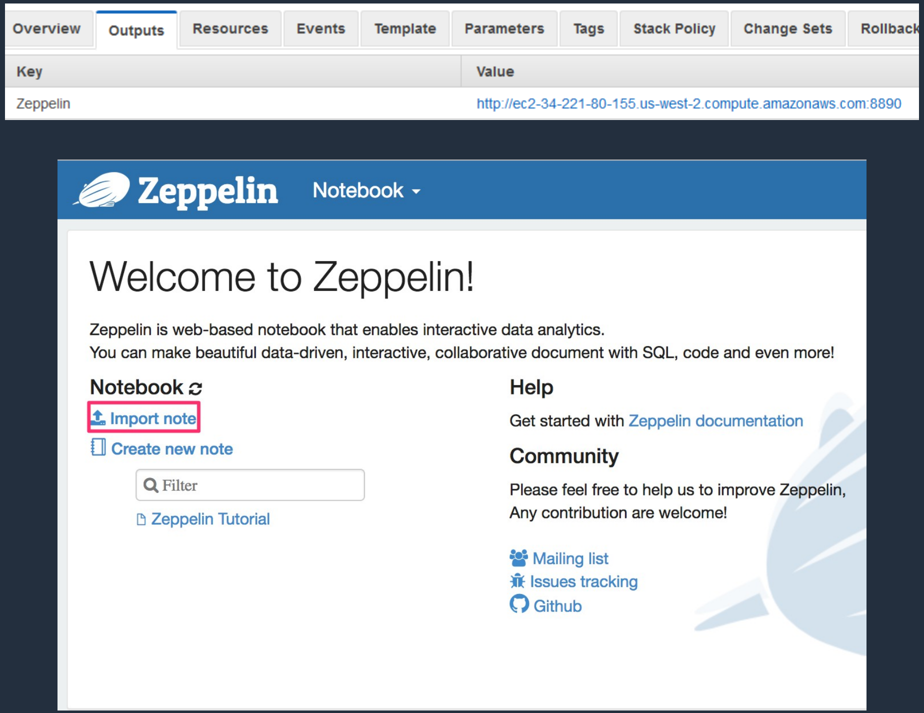
We are going to:

- 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

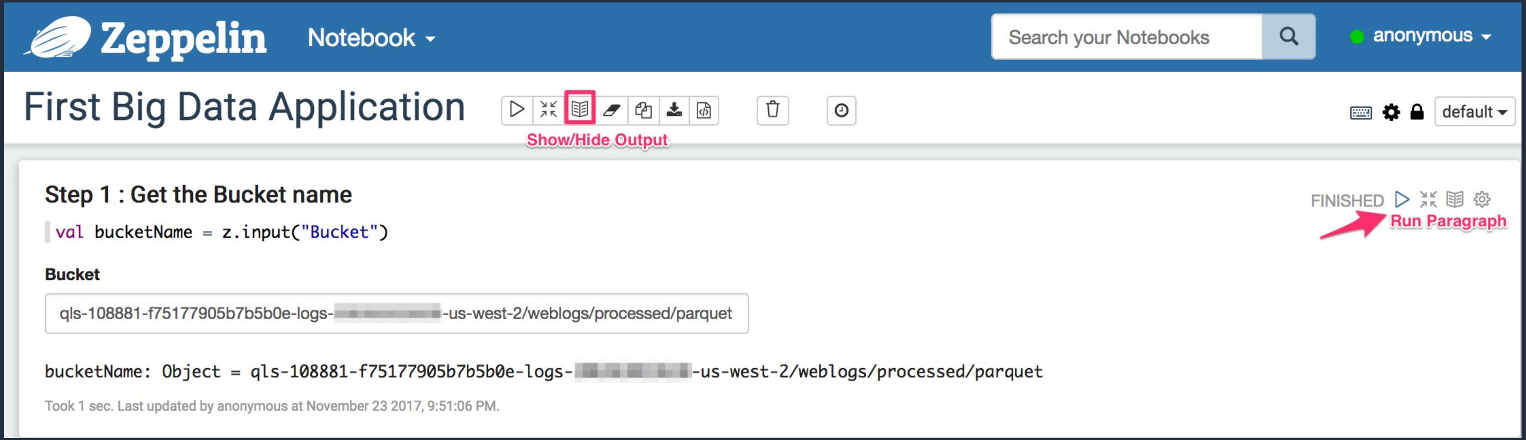
- Copy the Zeppelin end point in **the AWS CloudFormation** output section
- Open the Zeppelin link in a new browser window
- Download the First Big Data Application json file from the BigDataWorkshop.zip
- Import the notebook using the "Import Note" link on Zeppelin interface
- **Note: Disconnect from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



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

Activity 6A: Open the Zeppelin interface

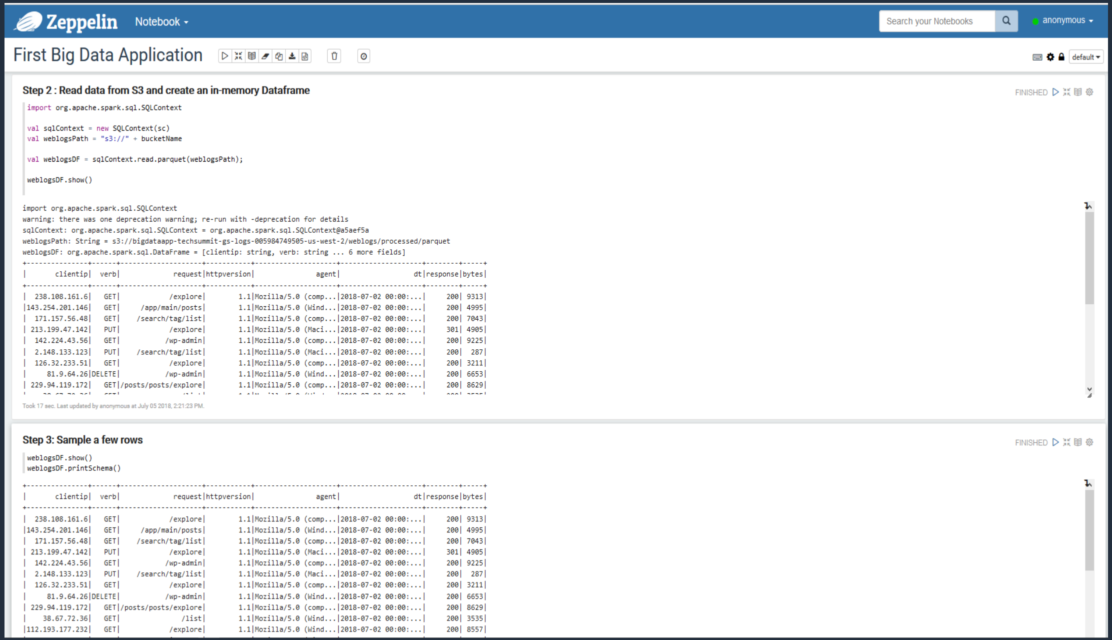
- Step 1: Run the first paragraph
- Enter **<stackname>-logs-<account>-us-west-2/processed/parquet** in the Bucket field as input
- This is where the processed parquet files were stored earlier in your S3 bucket



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

Activity 6B: Run the notebook

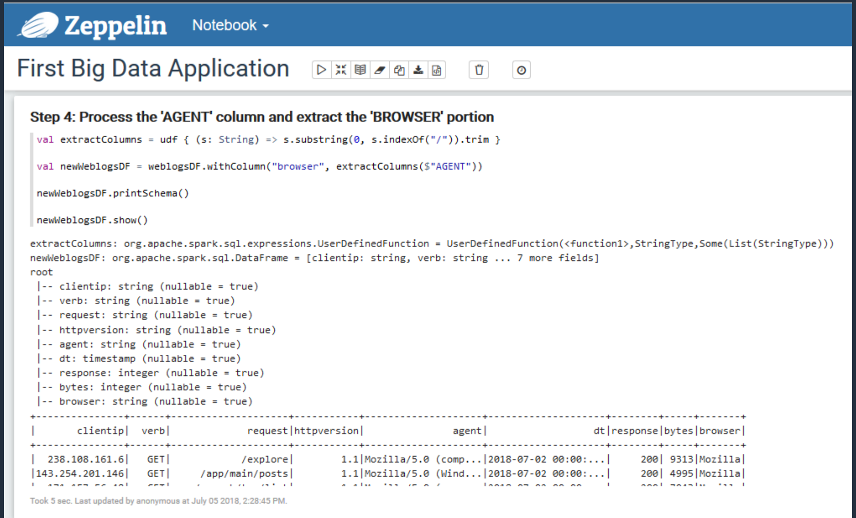
- Execute Step 2
  - Create a data frame with the parquet files from the AWS Glue ETL job
- Execute Step 3
  - Sample a few rows



© 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



© 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

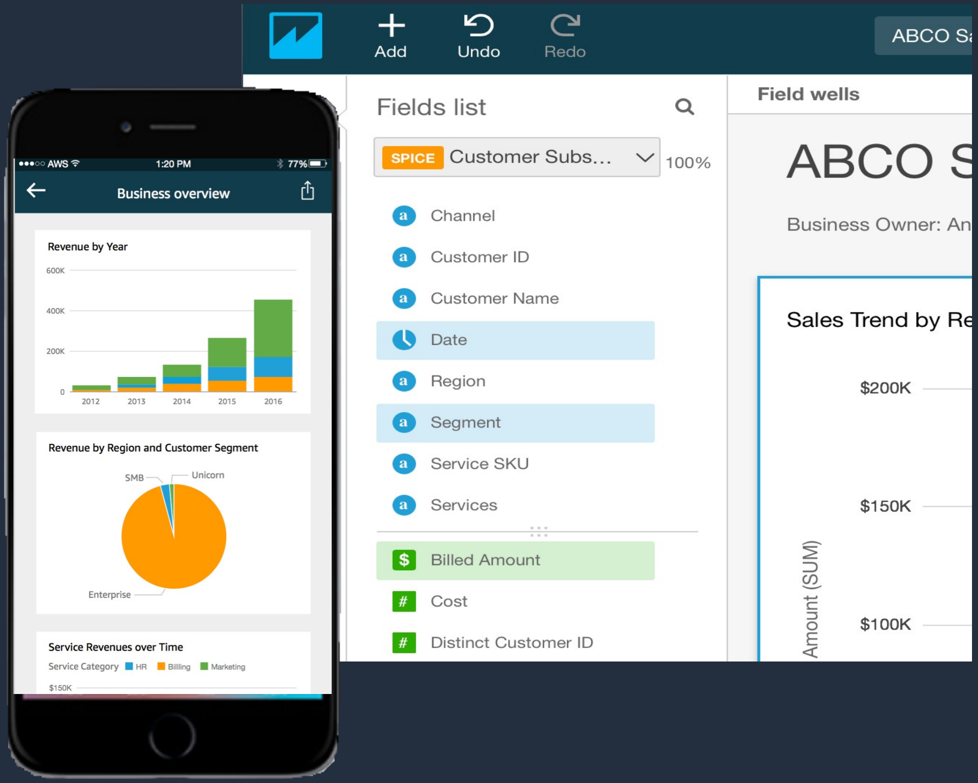
© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Optional exercise: Data visualization with Amazon QuickSight

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

Amazon QuickSight

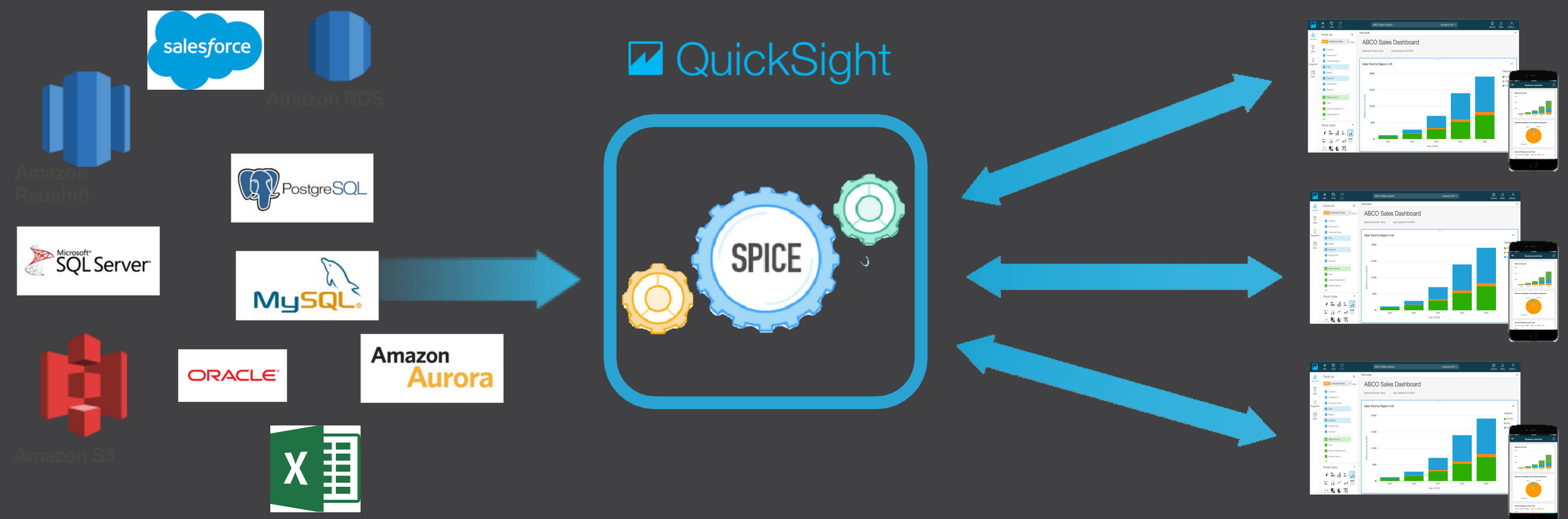
- **Fast, easy interactive analytics for anyone, everywhere**
- Ease of use targeted at business users
- Blazing-fast performance powered by SPICE
- Broad connectivity with AWS data services, on-premises data, files, and business applications
- Cloud-native solution that scales automatically
- 1/10th 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

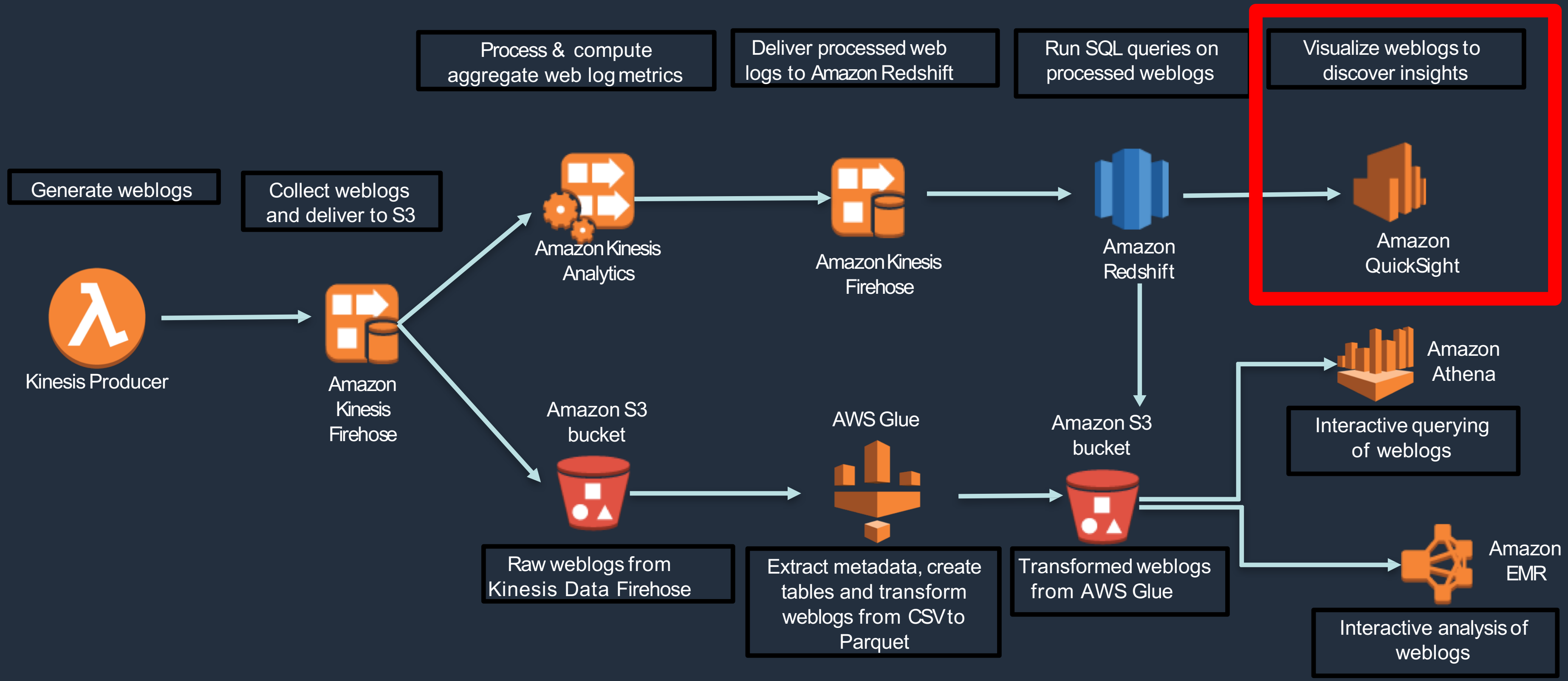


© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

Activity 7
Visualize results in Amazon QuickSight

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

Your application architecture



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

Activity 7: Visualization with Amazon QuickSight

Time: 10 mins

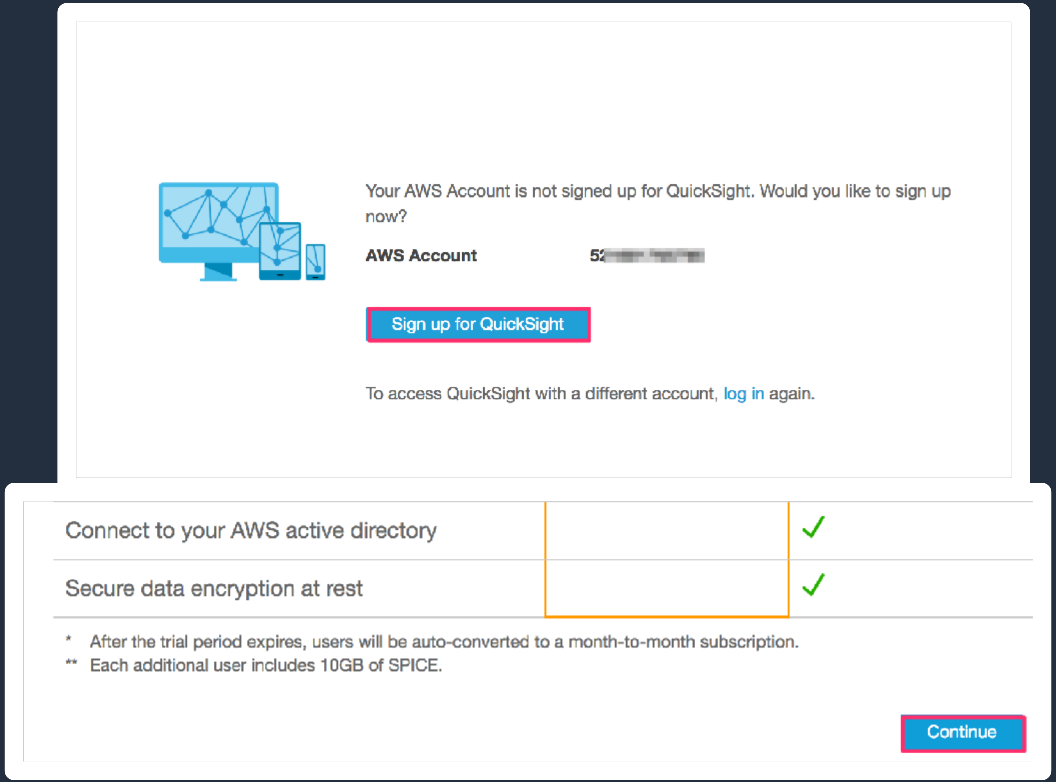
We are going to:

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

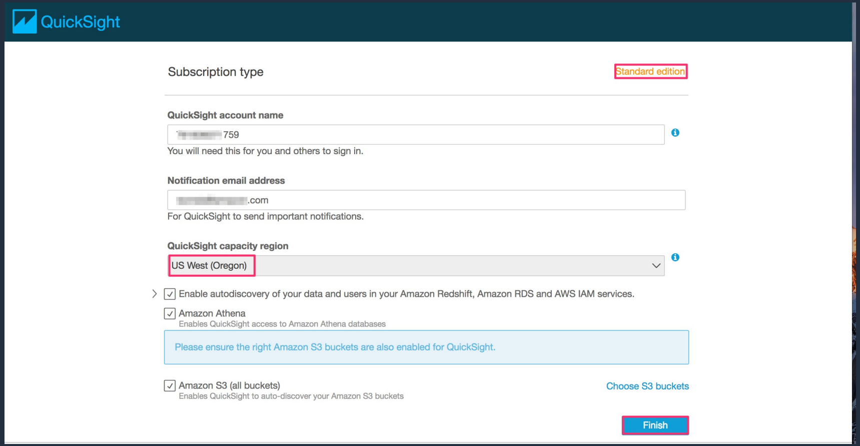
- Go to AWS console, click on Amazon QuickSight from the analytics section
- Click on "Sign up" in the next window
- Make sure the subscription type is standard and click "Continue" on the next screen



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

Activity 7A: Amazon QuickSight registration

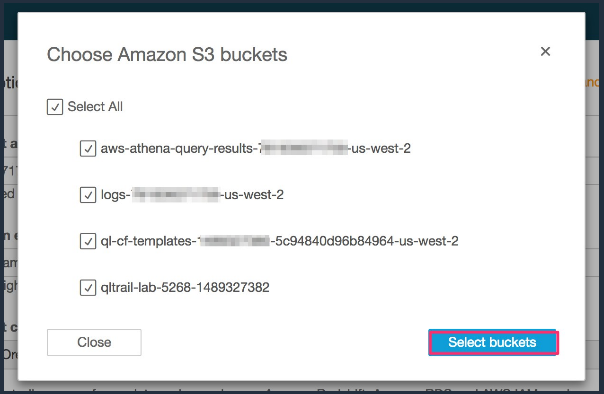
- On the "Subscription type" page, enter the account name (see **note** below)
- Enter your **email address**
- Select **US West region**
- Check the "**Amazon S3 (all buckets)**" box
- **Note: Amazon QuickSight account name is your AWS account number**

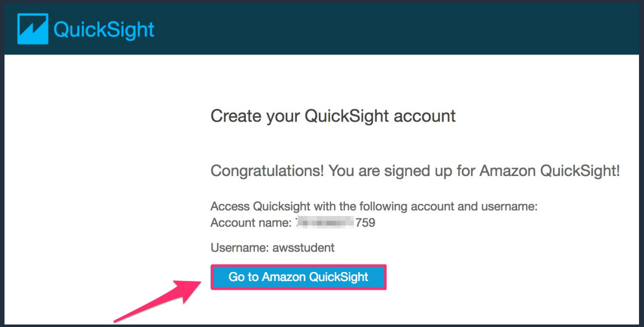


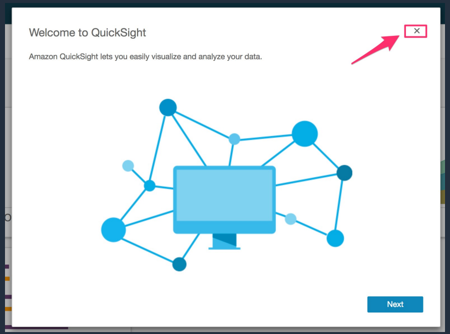
© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

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)



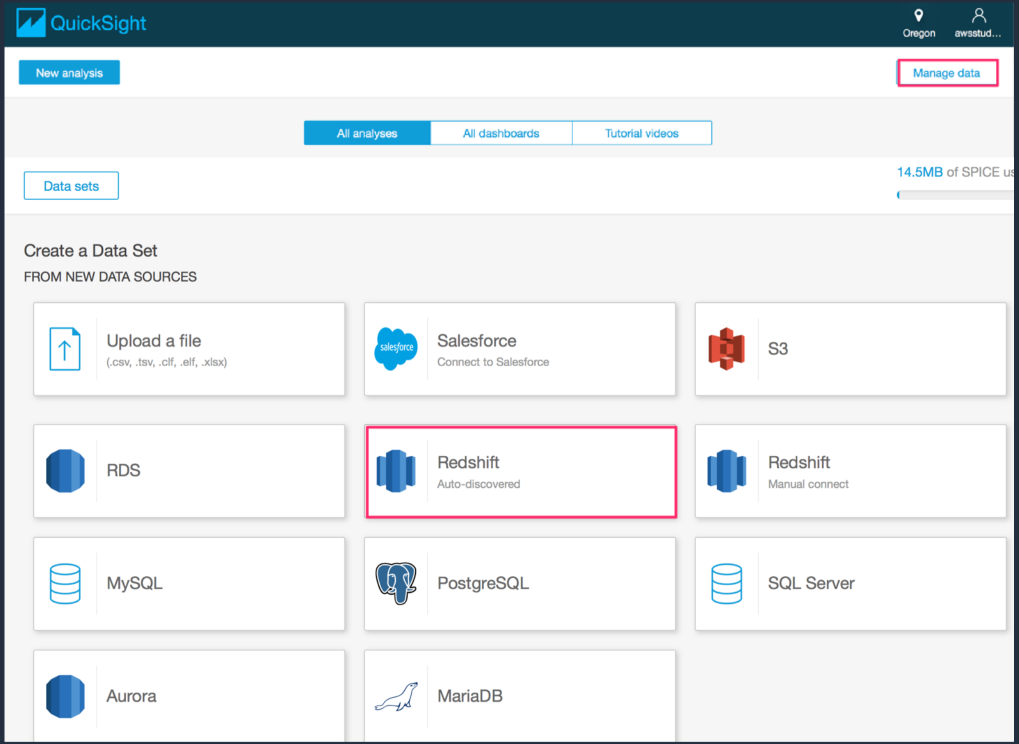




© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved

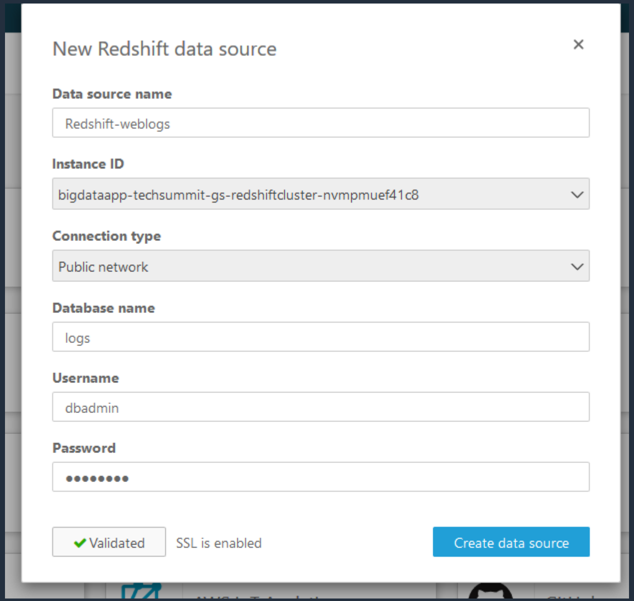
Activity 7B: Connect to data source

- 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

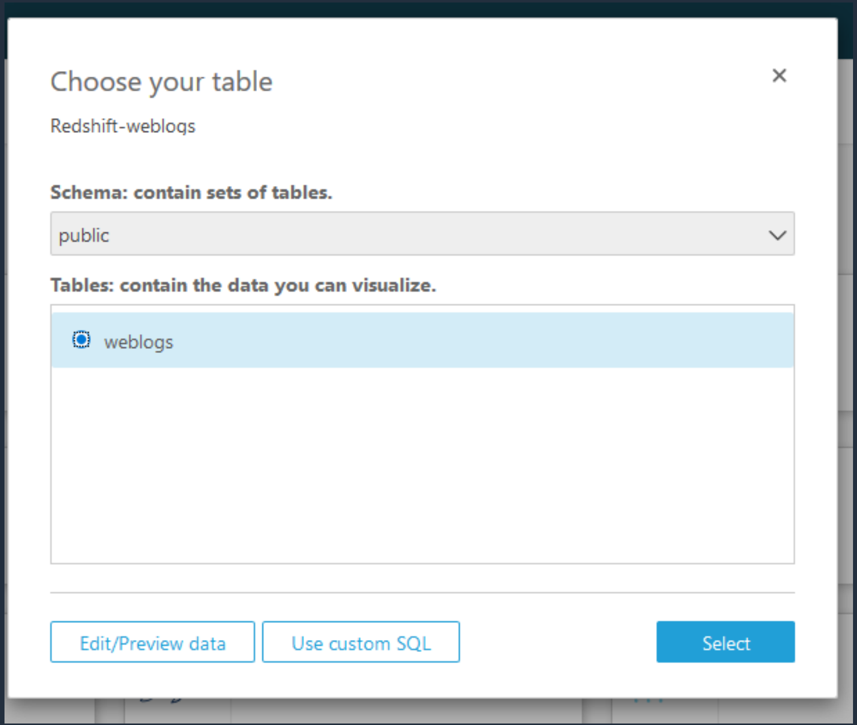


Note: Use ”dbadmin” as the username. You can get the Amazon Redshift database password by navigating to the AWS CloudFormation outputs section.

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

Activity 7C: Choose your weblogs Amazon Redshift table

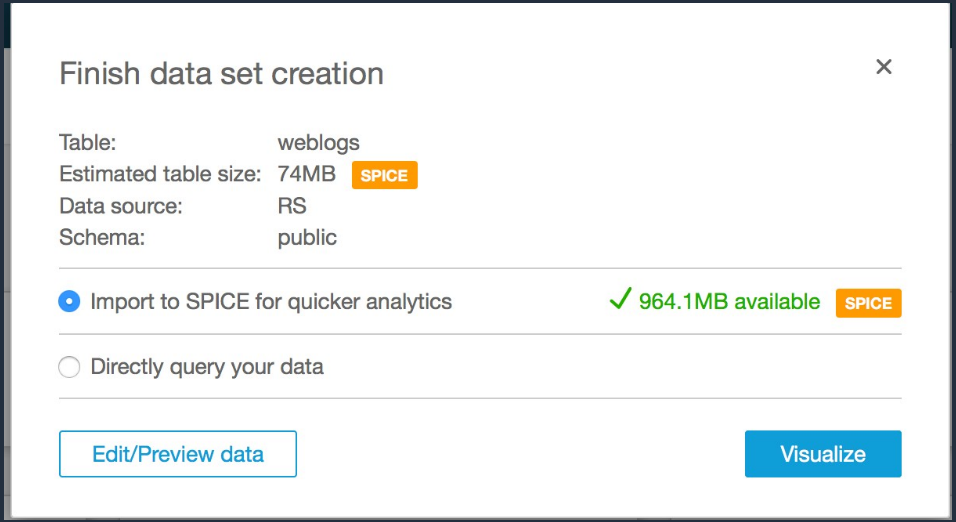
- Select "public" schema
- Select "weblogs" in tables



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

Activity 7D: Ingest data into SPICE

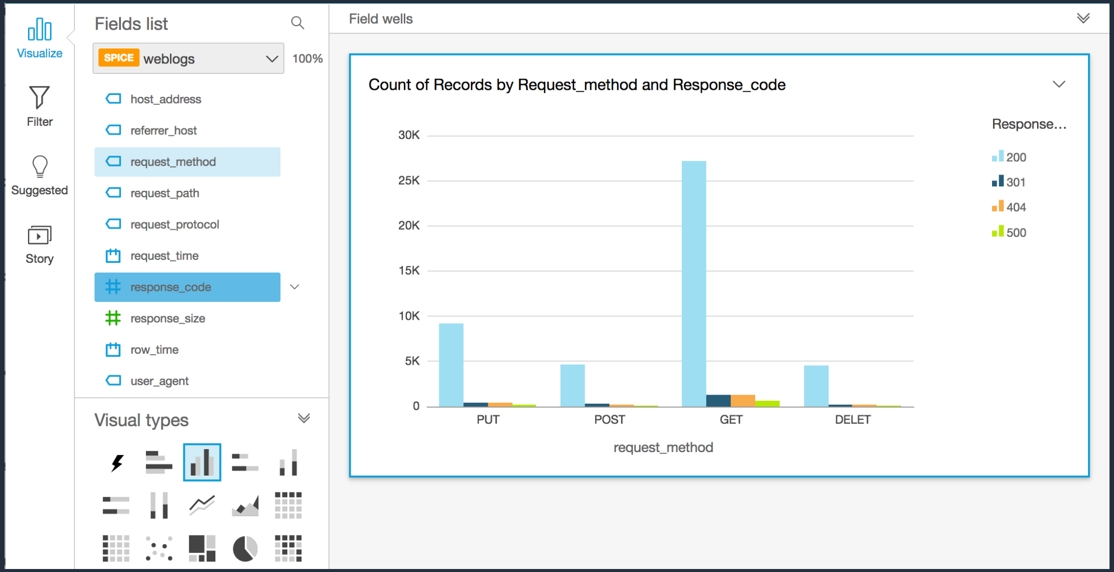
- SPICE is Amazon QuickSight's in-memory optimized calculation engine, designed specifically for fast, interactive data visualization
- You can improve the performance of database data sets by importing the data into SPICE instead of using a direct query to the database



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

Activity 7E: Creating your first analysis

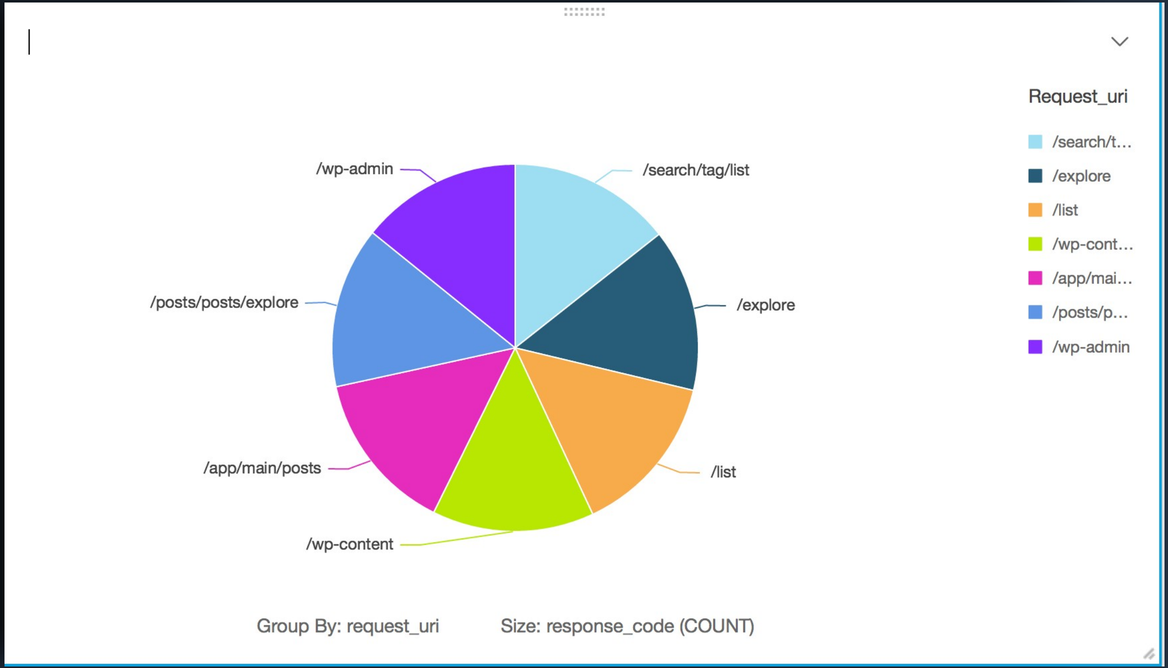
- What are the most requested http request types and their corresponding response codes for this site?
- Simply select request, response, and let AUTOGRAPH create the optimal visualization



© 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