

## Whizlabs - ML Specialty Exam Course - Data Analysis - part 2

<https://www.whizlabs.com/learn/course/aws-mls-practice-tests/video/3494>

### Kinesis Video Stream

#### AWS Machine Learning Exploratory Data Analysis

##### Kinesis Video Streams

- ❑ Automatically provisions and scales infrastructure to read streaming media
- ❑ Producers such as web cams, security cameras, audio feeds, images
  - ❑ Securely stream video using the Kinesis Video Streams SDK
- ❑ Storage
  - ❑ Kinesis Video Streams ingests the stream data, stores, encrypts, and indexes the stream for either real-time or batch analytics
- ❑ Consumers
  - ❑ Real-time or batch machine learning applications
  - ❑ Video processing or playback services



### Use Cases

# Kinesis Video Streams in Machine Learning

- ❑ Facial recognition
  - ❑ Facial analytics using machine learning services and algorithms such as the SageMaker built-in Image Classification algorithm or the Rekognition service
- ❑ Object detection
  - ❑ Object identification algorithm to detect when certain objects appear in the stream
- ❑ Computer vision and video analytics using machine learning frameworks such as Apache MxNet, TensorFlow, and OpenCV
- ❑ Smart home/city applications
  - ❑ Home surveillance, Red light cams
- ❑ Industrial automation
  - ❑ Predictive maintenance using MxNet, TensorFlow, and OpenCV

## Quick Tour:

The screenshot shows the 'Create a new video stream' wizard in the AWS Management Console. The top navigation bar includes 'Services' and 'Resource Groups'. The breadcrumb path is 'Kinesis Video Streams > Video streams > Create video stream'. The main title is 'Create a new video stream' with an 'Info' link. A note below states: 'Create a new video stream and then use the Kinesis Video Streams API to put data into or read data from your video stream. Kinesis Video Streams resources are not covered under the AWS Free Tier [?], and usage-based charges apply. For more information, see Kinesis Video Streams pricing [?].'

**Setup**

**Video stream name**  
Your stream name must be unique for the current account and region.  
Input field: machinelearningvideostream

Maximum length: 128 characters. May include numbers, letters, underscores (\_), and hyphens (-).

**Default configuration**  
Use the default retention period and the AWS managed customer master key (CMK).

**Custom configuration**  
Specify your own retention period and a customer managed CMK.

**Data retention** [Info](#)  
1 day(s)

**KMS customer master key (CMK)** [Info](#)  
(Default) aws/kinesisvideo

**Info** The KMS customer master key (CMK) used for encryption cannot be changed later. [Info](#)  
Kinesis Video Streams use the AWS managed CMK (aws/kinesisvideo) to encrypt your data by default.

**Tags** [Info](#)  
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Cancel **Create video stream**

To note we can specify our own Customer Managed Key (CMK)

The screenshot shows a configuration dialog with two main options:

- Default configuration**: Use the default retention period and the AWS managed customer master key (CMK).
- Custom configuration**: Specify your own retention period and a customer managed CMK.

A cursor is visible near the top center of the dialog.

Retention period can be in hours, days and years

#### Data retention [Info](#)

Data retention specifies a period of time during which a video stream retains data.

The screenshot shows a dropdown menu for selecting a retention period unit:

- 1
- day(s) (selected)
- hour(s)
- day(s)
- year(s)

Below the dropdown, there is a note: "Minimum: 0 hours, maximum: 10 years".

Default is 1 day. Max is 10 years.

#### Data retention [Info](#)

Data retention specifies a period of time during which a video stream retains data.

The screenshot shows a dropdown menu for selecting a retention period unit:

- 1
- day(s) (selected)

Below the dropdown, there is a note: "Minimum: 0 hours, maximum: 10 years".

As for KMS - data is encrypted before it is stored to video stream storage layer (S3)

Data is always encrypted at REST

#### KMS customer master key (CMK) [Info](#)

Kinesis Video Stream uses AWS Key Management Service (KMS) to encrypt your data at rest. You can choose the AWS managed customer master key (CMK) (`aws/kinesisvideo`) to encrypt your data or specify a customer managed CMK.



**The KMS customer master key (CMK) used for encryption cannot be changed later. [Info](#)**

Kinesis Video Streams use the AWS managed CMK (`aws/kinesisvideo`) to encrypt your data by default.

#### AWS managed CMK

The AWS managed CMK is a CMK in your account that is created, managed, and used on your behalf by Kinesis Video Streams

#### Customer managed CMK

Customer managed CMKs are CMKs that you can create, own, and manage.

We can also set up our own key:

**AWS managed CMK**

The AWS managed CMK is a CMK in your account that is created, managed, and used on your behalf by Kinesis Video Streams

**Customer managed CMK**

Customer managed CMKs are CMKs that you can create, own, and manage.

Customer managed CMK in KMS [Info](#)

Enter CMK ID, alias name, or ARN

Browse KMS

Create key

The CMK can be specified using the ID, alias name, or as an ARN.

We can tag the video stream

▼ Tags [Info](#)

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key

Value - optional

Enter key

Enter value

Remove

Add new tag

Now the video stream is created

Success! machinelearningvideostream has been created.

Next, download the Kinesis Video Streams Producer SDK to set up your connected device as a producer for this stream.

Kinesis Video Streams > Video streams > machinelearningvideostream

machinelearningvideostream [Info](#)

Use this page to view and configure your Kinesis video stream.

▼ Connect video stream

Set up your producer [Info](#)

Use the Kinesis Video Streams Producer SDK to set up your producer.

Download SDK

► Media playback

Video stream info

Monitoring

Encryption

Data retention

Tags

Video stream info

Video stream name

machinelearningvideostream

Video stream ARN

arn:aws:kinesisvideo:us-east-1:001178231653:stream/machinelearningvideostream/1578869078204

Status

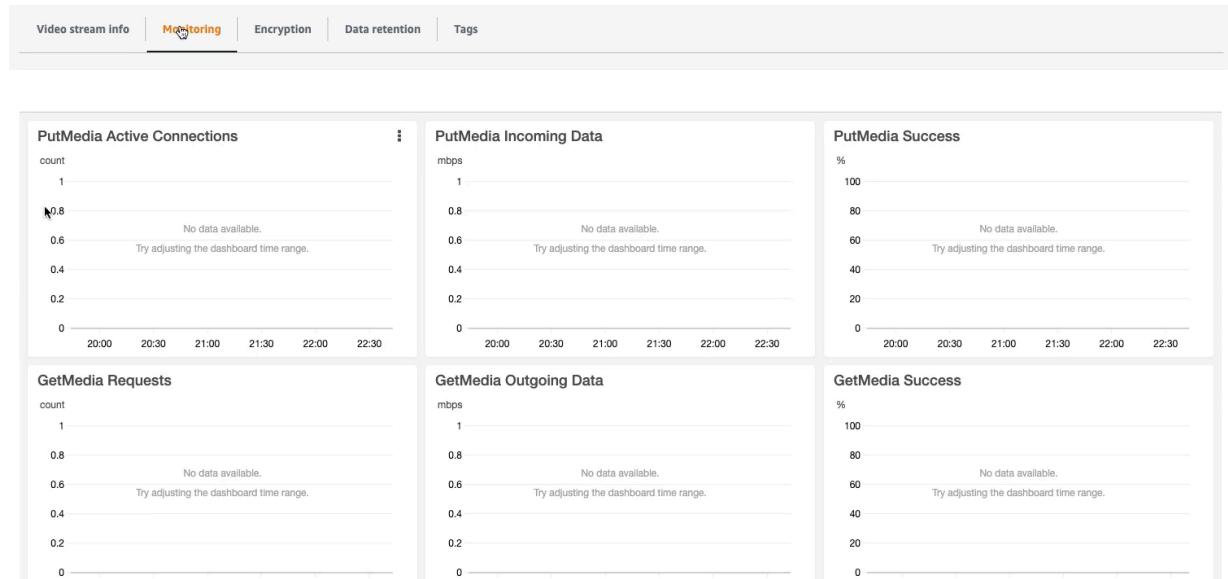
Active

Creation time

January 12, 2020 5:44:38 PM

Version

aeRTVCVIQ3YvwdxARJp



**Encryption** Info

Kinesis Video Streams use AWS Key Management Service (KMS) to encrypt your data at rest.

KMS key alias name <code>aws/kinesisvideo (default)</code>	KMS key ID <code>870fe195-2658-44ff-9e4a-4e50cbf2fb68</code>
KMS key alias ARN <code>arn:aws:kms:us-east-1:001178231653:alias/aws/kinesisvideo</code>	KMS key ARN <code>arn:aws:kms:us-east-1:001178231653:key/870fe195-2658-44ff-9e4a-4e50cbf2fb68</code>
Description Default master key that protects my Kinesis Video Streams data when no other key is defined	Account <code>001178231653</code>

**Data retention** Info

Data retention specifies a period of time during which a video stream retains data.

Data retention <code>1 day(s)</code>
---

We can also look at Signaling channels

**Kinesis Video Streams**

**Signaling channels**

**Signaling channels** Info

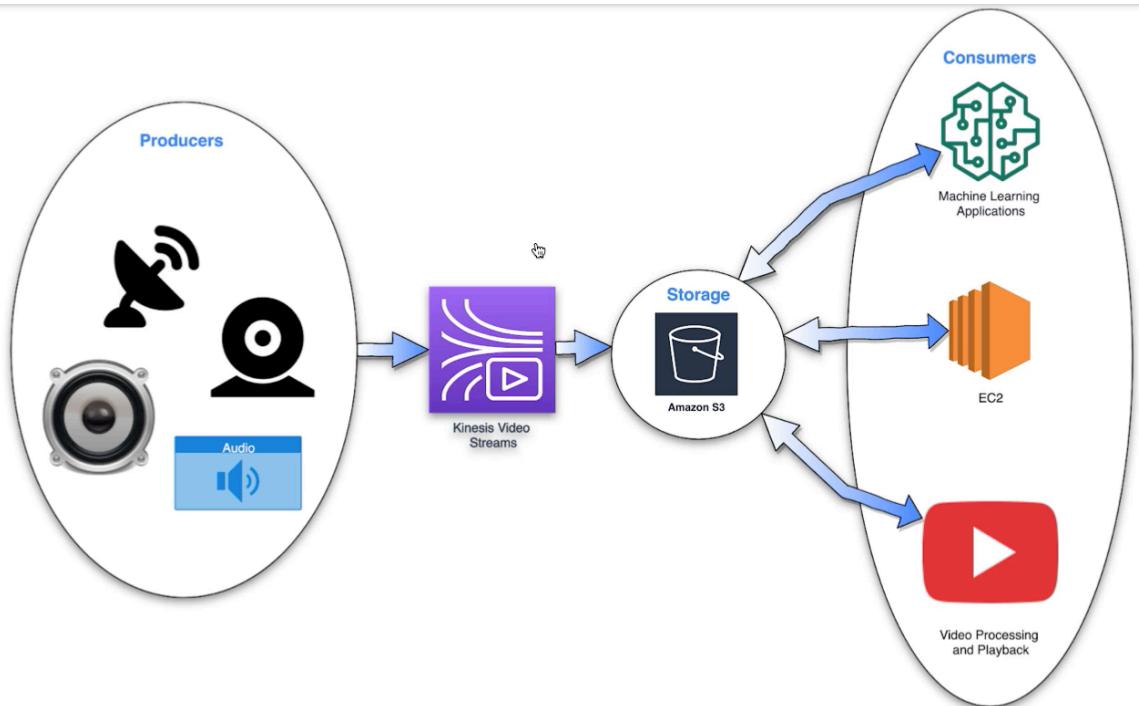
Find signaling channels by name prefix

Signal	Signaling channel name	Status	TTL (seconds)	Creation time
No signaling channels No signaling channels to display.				

**Create signaling channel**

For example; doorbell camera - we can see it and talk to them

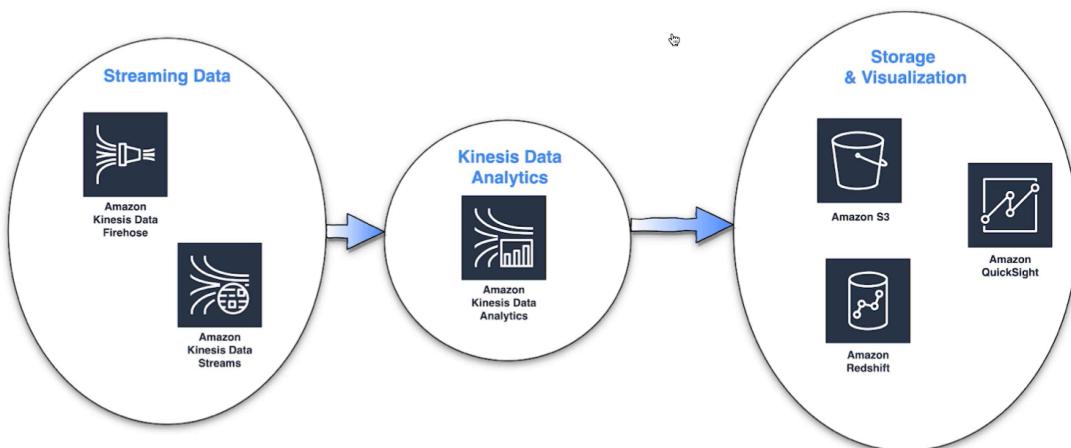
## Typical architecture for Kinesis Video Streams



## Kinesis Data Analytics

# AWS Machine Learning Exploratory Data Analysis

## Kinesis Data Analytics



- ❑ Use SQL to process streaming data
- ❑ Sources: Kinesis Data Streams and Kinesis Data Firehose
- ❑ SQL queries put to S3, Redshift, or visualization and Business Intelligence tools
- ❑ A Kinesis Data Analytics streaming application consists of three parts
  - ❑ Streaming data sources
  - ❑ Analytics written in SQL
  - ❑ Destinations for the results
- ❑ The streaming application continuously reads data from a streaming source, generates analytics using SQL code, and emits results to 1 to 4 destinations

## Kinesis Data Analytics Streaming Application

- ❑ Streaming application is the primary resource in Kinesis Data Analytics
- ❑ Create/manage streaming applications via the console or the Kinesis Data Analytics API
- ❑ The API has operations to manage your streaming applications
- ❑ A Kinesis Data Analytics streaming application continually read and process streaming data in real-time
- ❑ Application code in SQL processes the incoming streaming data and produces output
- ❑ Kinesis Data Analytics writes the output to configured destinations
  
- ❑ Streaming data input from Kinesis Data Streams or Kinesis Data Firehose
- ❑ Reference data from S3
- ❑ Data read continuously from the streaming data sources but one time from the reference source
- ❑ Reference data sources are used for joining against the incoming stream to enrich the data

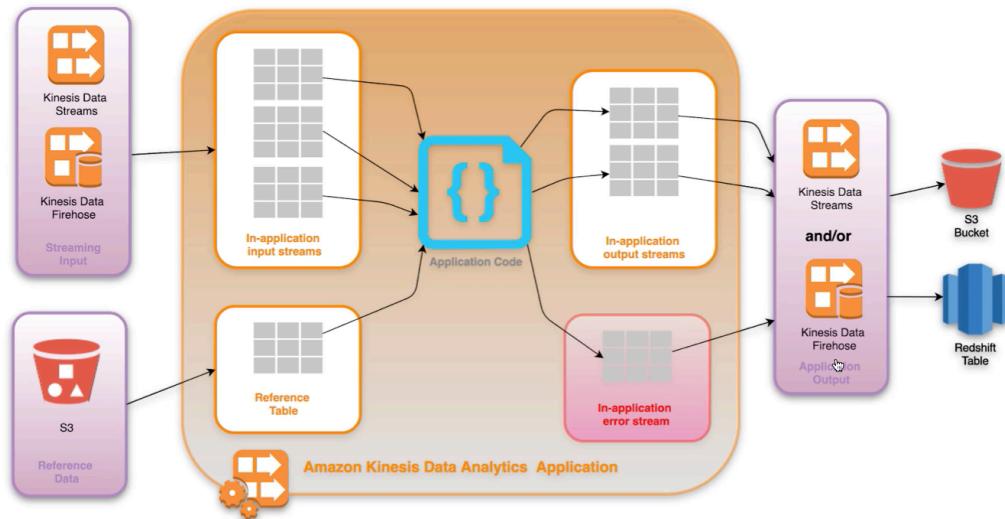
Example:

In application input stream read continuously from Firehose or Stream.  
Reference table is read just once

Application writes to: in-application output stream and errors to in-application error stream

The in-application output streams then writes the output to 1-4 destinations, that can be reached via a Kinesis Data Streams and/or Firehose => 1-4 S3/Redshift destinations

## Kinesis Data Analytics Streaming Application



This is typically used to engineer data from sources on-the-fly, in real time.

## Kinesis Data Analytics - Lab

**Amazon Kinesis Analytics**

Run continuous analysis on streaming data in real-time from Amazon Kinesis Streams and Firehose.

**Create application**

**Getting started guide**

**Generate time-series analytics**

Calculate performance metrics over time windows, and stream values to Amazon S3 or Amazon Redshift through an Amazon Kinesis Firehose delivery system.

**Feed real-time dashboards**

Send aggregated and processed streaming results downstream to feed real-time dashboards.

**Create real-time metrics**

Create custom metrics and triggers for use in real-time monitoring, notifications, and alarms.

### Create Application

We will use the stock streaming data stream

We can run a **SQL** or **Apache Flink** to transform the data

**Kinesis Analytics - Create application**

Kinesis Analytics applications continuously read and analyze data from a connected streaming source in real-time. To enable interactivity with your data during configuration you will be prompted to run your application. Kinesis Analytics resources are not covered under the [AWS Free Tier](#), and [usage-based charges apply](#). For more information, see [Kinesis Analytics pricing](#).

Application name  
stocksstream  
Acceptable characters are uppercase and lowercase letters, numbers, underscores, hyphens, and periods.

Description - optional

Runtime

**SQL**  
Process data in real-time using SQL, which provides an easy way to quickly query large volumes of streaming data without learning new frameworks or languages. [Learn more](#)

**Apache Flink**  
Apache Flink is an open-source framework and distributed processing engine for stateful computations over unbounded and bounded data streams. [Learn more](#)

**!** After you create the application, you can't change the type or version of the runtime environment.

[Cancel](#) [Create application](#)

## Connect streaming data:

**Kinesis Analytics applications** > stocksstream

**stocksstream**

Application ARN: arn:aws:kinesisanalytics:us-east-1:001178231653:application/stocksstream  
Application version ID: 1 [Edit](#)

**Successfully created Application stocksstream** [X](#)  
Next, choose **Connect streaming data**.

**Source**

**Streaming data**  
Connect to an existing Kinesis stream or Firehose delivery stream, or easily create and connect to a new demo Kinesis stream. Each application can connect to one streaming data source. [Learn more](#)

**Connect streaming data**

Reference data (optional)

Amazon Kinesis

Dashboard

Data Streams

Data Firehose

**Data Analytics**

Video Streams

External resources

What's new

## Connect streaming data source

Choose from your Kinesis data streams and Firehose delivery streams, or quickly configure a demo Kinesis stream that can be used to explore Kinesis Analytics.

Choose source       Configure a new stream

Source

Kinesis data stream  
Kinesis data stream is an ordered sequence of data records used for rapid and continuous data intake and aggregation.

Kinesis Firehose delivery stream  
Kinesis Firehose delivery streams send source records to the destinations that you specify, automatically and continuously.

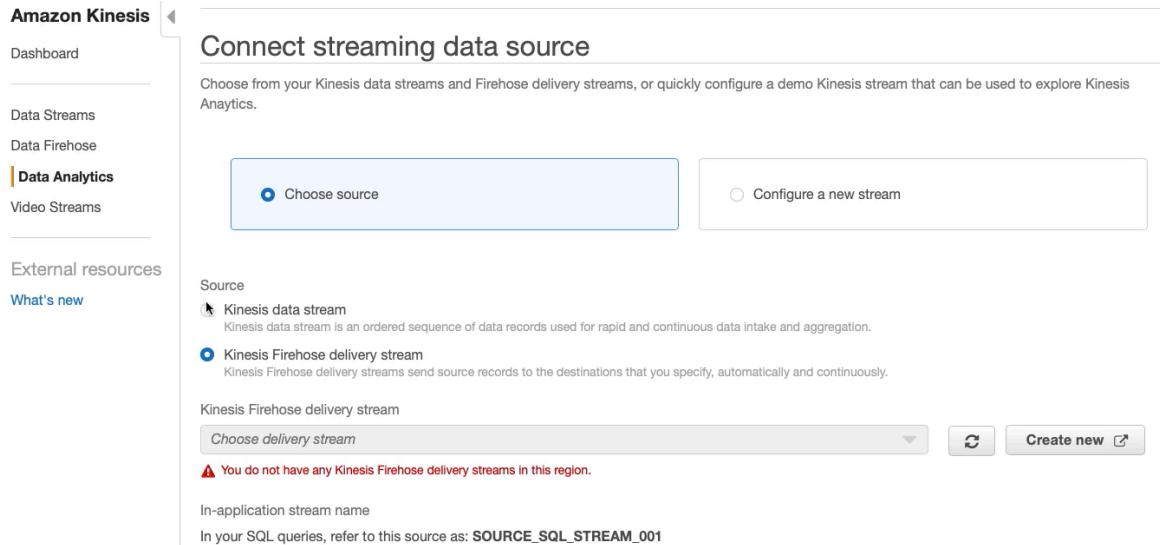
Kinesis Firehose delivery stream

Choose delivery stream

⚠ You do not have any Kinesis Firehose delivery streams in this region.

In-application stream name

In your SQL queries, refer to this source as: **SOURCE\_SQL\_STREAM\_001**



To note, We can do record pre-processing with Lambda

### Record pre-processing with AWS Lambda

Kinesis Analytics can invoke your Lambda function to pre-process records before they are used in this application. To pre-process records, your Lambda function must be compliant with the required record transformation output model. [Learn more](#)

#### Record pre-processing

- Disabled  
 Enabled

We are going to configure a new stream

## Connect streaming data source

Choose from your Kinesis data streams and Firehose delivery streams, or quickly configure a demo Kinesis stream that can be used to explore Kinesis Analytics.

Choose source       Configure a new stream

### Create your first stream

Create a demo stream that you can use to explore Kinesis Analytics. This stream will be populated with sample stock ticker data. See [Kinesis Streams pricing](#)

**Create a demo stream**  
(recommended)

Create a Firehose delivery stream to continuously deliver to source data (to Amazon S3, Redshift, or Elasticsearch) and make source data available to applications.

[Go to Kinesis Firehose](#)

Configure a Kinesis stream to continuously collect and temporarily store source data, which can be consumed by an application.

[Go to Kinesis Streams](#)

[Cancel](#)      **Save and continue**

Create a demo stream that you can use to explore Kinesis Analytics. This stream will be populated with sample stock ticker data. See [Kinesis Streams pricing](#)

**Create a demo stream**

- ✓ Create/update IAM role **kinesis-analytics-stocksstream-us-east-1**
- ✓ Create Kinesis stream **kinesis-analytics-demo-stream** (takes on average 30-40 seconds)
- ✓ Begin populating stream **kinesis-analytics-demo-stream** with sample stock ticker data
  - Discover schema: capture a stream sample, identify data format, apply schema
- ⓘ Select stream **kinesis-analytics-demo-stream** from your streams

We can now select the newly created stock stream

Kinesis data stream

kinesis-analytics-demo-stream

[View kinesis-analytics-demo-stream in Kinesis data streams](#)

In-application stream name

In your SQL queries, refer to this source as: **SOURCE\_SQL\_STREAM\_001**

It has just 1 shard

Amazon Kinesis

Dashboard

**Data Streams**

- Data Firehose
- Data Analytics
- Video Streams

External resources

What's new

kinesis-analytics-demo-stream

Configure producers [\(2\)](#) to put data records into a data stream. Configure consumers [\(2\)](#) to continuously process data stream records.

**Details** Monitoring Tags Enhanced fan-out

Stream ARN: arn:aws:kinesis:us-east-1:001178231653:stream/kinesis-analytics-demo-stream

Status: ACTIVE

Sending data to the stream: Configure producers to send data using the Streams PUT API operation or the Amazon Kinesis Producer Library (KPL). [Learn more](#)

Receiving and processing data: **Kinesis Streams:** Build a custom application to process or analyze streaming data using the Kinesis Client Library. [Learn more](#)  
**Kinesis Firehose:** Connect your Kinesis stream to a Firehose delivery stream. [Go to Kinesis Firehose](#)  
**Kinesis Analytics:** Analyze streaming data from Kinesis Streams in real time. [Go to Kinesis Analytics](#)

Shards

In order to adapt to changes in the rate of data flow through the stream, Amazon Kinesis Streams supports scaling, which enables you to adjust the number of shards in your stream. [Learn more](#)

Open shards: 1 [Edit](#)

Closed shards: 0 [Edit](#)

Amazon Kinesis

Dashboard

**Data Streams**

- Data Firehose
- Data Analytics**
- Video Streams

External resources

What's new

Kinesis Analytics applications

Kinesis Analytics applications continuously read and process data from streaming sources in real-time. [Learn more](#)

**Create application** Actions

Filter or search by application name

	Application name	Runtime	State
stocksstream	SQL	Ready	

Created: Jan 15, 2020 9:19:29 PM  
Last updated: Jan 15, 2020 9:19:29 PM

**Application details**

**Input**

- Source ARN: No source ARN specified
- Role ARN: No Role ARN specified
- Format: No record format specified

**Output**

- Destination ARN: No destination ARN specified
- Role ARN: No Role ARN specified
- Format: No record format specified

We now discover the schema

### Schema

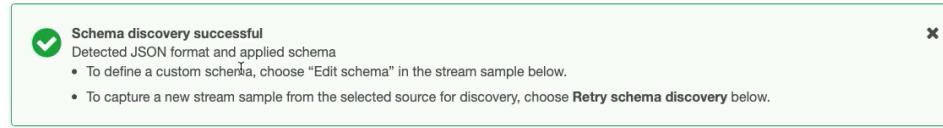
Schema discovery can generate a schema using recent records from the source. Schema column names are the same as in the source, unless they contain special characters, repeated column names, or reserved keywords. [Learn more](#)

**Discover schema**

Discovering schema for stream **kinesis-analytics-demo-stream** with starting point 'NOW'

## Schema

Schema discovery can generate a schema using recent records from the source. Schema column names are the same as in the source, unless they contain special characters, repeated column names, or reserved keywords. [Learn more](#)



A screenshot showing the "Formatted" tab selected. Below it is a table with columns: TICKER\_SYMBOL, SECTOR, CHANGE, and PRICE. The data rows are: QXZ (RETAIL, -0.2, 51.02), QAZ (FINANCIAL, 3.68, 205.77), NFS (ENERGY, -0.64, 86.78), PPL (HEALTHCARE, -0.16, 30.52), BAC (FINANCIAL, -0.22, 14.56), NGC (HEALTHCARE, -0.16, 5.28), TGT (RETAIL, 2.98, 69.93), ABC (RETAIL, -0.67, 22.68), ASD (FINANCIAL, 0.58, 64.09), and MJN (RETAIL, 4.5200000000000005, 154.09).

A screenshot showing the "Formatted" tab selected. Below it is a table with columns: TICKER\_SYMBOL, SECTOR, CHANGE, and PRICE. The data rows are: QXZ (RETAIL, -0.2, 51.02), QAZ (FINANCIAL, 3.68, 205.77), NFS (ENERGY, -0.64, 86.78), PPL (HEALTHCARE, -0.16, 30.52), BAC (FINANCIAL, -0.22, 14.56), NGC (HEALTHCARE, -0.16, 5.28), TGT (RETAIL, 2.98, 69.93), ABC (RETAIL, -0.67, 22.68), ASD (FINANCIAL, 0.58, 64.09), and MJN (RETAIL, 4.5200000000000005, 154.09).

To note, we could also add a lambda to update that ticket symbol.

Now we go too the SQL Editor

Real time analytics

Author your own SQL queries or add SQL from templates to easily analyze your source data. [Learn more](#)

[Go to SQL editor](#)

Would you like to start running "stocksstream"?

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

No, I'll do this later      Yes, start application

Kinesis Analytics applications > stocksstream > SQL Editor

## Real-time analytics

Save and run SQL   Add SQL from templates   Download SQL   [SQL reference guide](#)   [Kinesis data generator tool](#)

```

1 /**
2  * Welcome to the SQL editor
3  * =====
4  *
5  * The SQL code you write here will continuously transform your streaming data
6  * when your application is running.
7  *
8  * Get started by clicking "Add SQL from templates" or pull up the
9  * documentation and start writing your own custom queries.
10 */

```

We are starting your application, which usually takes 30-90 seconds.

Application has now started

Application status: RUNNING

Source

Real-time analytics

Destination

Streaming data

SOURCE\_SQL\_STREAM\_001

Reference data (optional) 

Connect reference data

Actions 

Filter by column name

TICKER_SYMBOL VARCHAR(4)	SECTOR VARCHAR(16)	CHANGE REAL	PRICE REAL
-----------------------------	-----------------------	----------------	---------------



No rows in source stream

We were unable to detect any rows streaming into your source stream. You will not be able to process this stream unless there is data present.

[Begin populating stream with sample stock ticker data](#)

## Populate it

Application status: RUNNING

Source

Real-time analytics

Destination

Streaming data

SOURCE\_SQL\_STREAM\_001

Reference data (optional) 

Connect reference data

Actions 

Filter by column name

ROWTIME TIMESTAMP	TICKER_SYMBOL VARCHAR(4)	SECTOR VARCHAR(16)	CHANGE REAL	PRICE REAL	PARTITION_KEY VARCHAR(512)	S V.
2020-01-16 02:24:43.051	VVY	HEALTHCARE	0.3	35.15	PartitionKey	4f
2020-01-16 02:24:43.051	UHN	FINANCIAL	0.55	544.52	PartitionKey	4f
2020-01-16 02:24:43.051	WFC	FINANCIAL	0.96	47.75	PartitionKey	4f

## Add SQL from template

Amazon Kinesis

- Dashboard
- Data Streams
- Data Firehose
- Data Analytics**
- Video Streams
- External resources
- What's new

Select a template to preview the SQL

To customize the SQL, add it to the editor

Continuous filter

Aggregate function in a tumbling time window

Aggregate function in a sliding time window

Aggregate function in a sliding row window

Multi-step application

Anomaly detection

Approximate top-K items

Approximate distinct count

Data enrichment (join)

Aggregate using two time windows

[Cancel \(return to the editor\)](#) [Add this SQL to the editor](#)

<input type="radio"/> Continuous filter	
<input checked="" type="radio"/> Aggregate function in a tumbling time window	<pre>-- ** Aggregate (COUNT, AVG, etc.) + Tumbling Time Window ** -- Performs function on the aggregate rows over a 10 second tumbling window for a specified --    ----- ----- -----  --     SOURCE     INSERT     DESTIN.   -- Source--&gt;  STREAM  --&gt;  &amp; SELECT  --&gt;  STREAM  --&gt;Destination --               (PUMP)             --   '-----' -----' -----' -- STREAM (in-application): a continuously updated entity that you can SELECT from and INSERT -- PUMP: an entity used to continuously 'SELECT ... FROM' a source STREAM, and INSERT SQL re -- Create output stream, which can be used to send to a destination CREATE OR REPLACE STREAM "DESTINATION_SQL_STREAM" (ticker_symbol VARCHAR(4), ticker_symbol_c -- Create a pump which continuously selects from a source stream (SOURCE_SQL_STREAM_001) -- performs an aggregate count that is grouped by columns ticker over a 10-second tumbling w -- and inserts into output stream (DESTINATION_SQL_STREAM) CREATE OR REPLACE PUMP "STREAM_PUMP" AS INSERT INTO "DESTINATION_SQL_STREAM" -- Aggregate function COUNT AVG MAX MIN SUM STDDEV_POP STDDEV Samp VAR_POP VAR_SAMP SELECT STREAM ticker_symbol, COUNT(*) AS ticker_symbol_count FROM "SOURCE_SQL_STREAM_001" -- Uses a 10-second tumbling time window GROUP BY ticker_symbol, FLOOR((SOURCE_SQL_STREAM_001).ROWTIME - TIMESTAMP '1970-01-01 00:00 </pre>
<input type="radio"/> Aggregate function in a sliding time window	
<input type="radio"/> Aggregate function in a sliding row window	
<input type="radio"/> Multi-step application	
<input type="radio"/> Anomaly detection	
<input type="radio"/> Approximate top-K items	

=> see how many ticket symbols come in a tumbling time window (10 seconds)

We can now run and save the SQL

Amazon Kinesis

- Dashboard
- Data Streams
- Data Firehose
- Data Analytics**
- Video Streams
- External resources
- What's new

## Real-time analytics

Save and run SQL Add SQL from templates Download SQL SQL reference guide Kinesis data generator tool

```

1
2
3  ** Aggregate (COUNT, AVG, etc.) + Tumbling Time Window **
4  -- Performs function on the aggregate rows over a 10 second tumbling window for a specified column.
5  --
6  |-----|-----|-----|
7  | SOURCE | INSERT | DESTIN |
8  |-----|-----|-----|
9  |          | (PUMP) |          |
10 -- STREAM (in-application): a continuously updated entity that you can SELECT from and INSERT into like a TABLE
11  -- PUMP: an entity used to continuously 'SELECT FROM' a source STREAM and 'INSERT SQL' results into an output STREAM

```

Saving SQL  
Running SQL

We can now see real-time analytics

Source	<b>Real-time analytics</b>	Destination																																							
In-application streams:	<b>Pause results</b> New results are added every 2-10 seconds. The results below are sampled. ⓘ <input checked="" type="checkbox"/> DESTINATION_SQL_STREAM <input type="checkbox"/> error_stream	<input checked="" type="checkbox"/> Scroll to bottom when new results arrive.																																							
<table border="1"> <thead> <tr> <th colspan="3">Filter by column name</th> </tr> </thead> <tbody> <tr> <td>2020-01-16 02:27:30.0</td> <td>TGT</td> <td>1</td> </tr> <tr> <td>2020-01-16 02:27:30.0</td> <td>DFT</td> <td>5</td> </tr> <tr> <td>2020-01-16 02:27:30.0</td> <td>SLW</td> <td>2</td> </tr> <tr> <td>2020-01-16 02:27:30.0</td> <td>WSB</td> <td>4</td> </tr> <tr> <td>2020-01-16 02:27:30.0</td> <td>QXZ</td> <td>6</td> </tr> <tr> <td>2020-01-16 02:27:30.0</td> <td>MMB</td> <td>3</td> </tr> <tr> <td>2020-01-16 02:27:30.0</td> <td>RFV</td> <td>3</td> </tr> <tr> <td>2020-01-16 02:27:30.0</td> <td>BNM</td> <td>2</td> </tr> <tr> <td>2020-01-16 02:27:30.0</td> <td>CRM</td> <td>3</td> </tr> <tr> <td>2020-01-16 02:27:30.0</td> <td>WAS</td> <td>1</td> </tr> <tr> <td>2020-01-16 02:27:30.0</td> <td>UHN</td> <td>1</td> </tr> <tr> <td>2020-01-16 02:27:30.0</td> <td>XTC</td> <td>1</td> </tr> </tbody> </table>			Filter by column name			2020-01-16 02:27:30.0	TGT	1	2020-01-16 02:27:30.0	DFT	5	2020-01-16 02:27:30.0	SLW	2	2020-01-16 02:27:30.0	WSB	4	2020-01-16 02:27:30.0	QXZ	6	2020-01-16 02:27:30.0	MMB	3	2020-01-16 02:27:30.0	RFV	3	2020-01-16 02:27:30.0	BNM	2	2020-01-16 02:27:30.0	CRM	3	2020-01-16 02:27:30.0	WAS	1	2020-01-16 02:27:30.0	UHN	1	2020-01-16 02:27:30.0	XTC	1
Filter by column name																																									
2020-01-16 02:27:30.0	TGT	1																																							
2020-01-16 02:27:30.0	DFT	5																																							
2020-01-16 02:27:30.0	SLW	2																																							
2020-01-16 02:27:30.0	WSB	4																																							
2020-01-16 02:27:30.0	QXZ	6																																							
2020-01-16 02:27:30.0	MMB	3																																							
2020-01-16 02:27:30.0	RFV	3																																							
2020-01-16 02:27:30.0	BNM	2																																							
2020-01-16 02:27:30.0	CRM	3																																							
2020-01-16 02:27:30.0	WAS	1																																							
2020-01-16 02:27:30.0	UHN	1																																							
2020-01-16 02:27:30.0	XTC	1																																							

Finally, add a Destination to store this aggregated streaming data  
=> Use Firehose to persist to S3 so SageMaker can consume

## AWS Glue

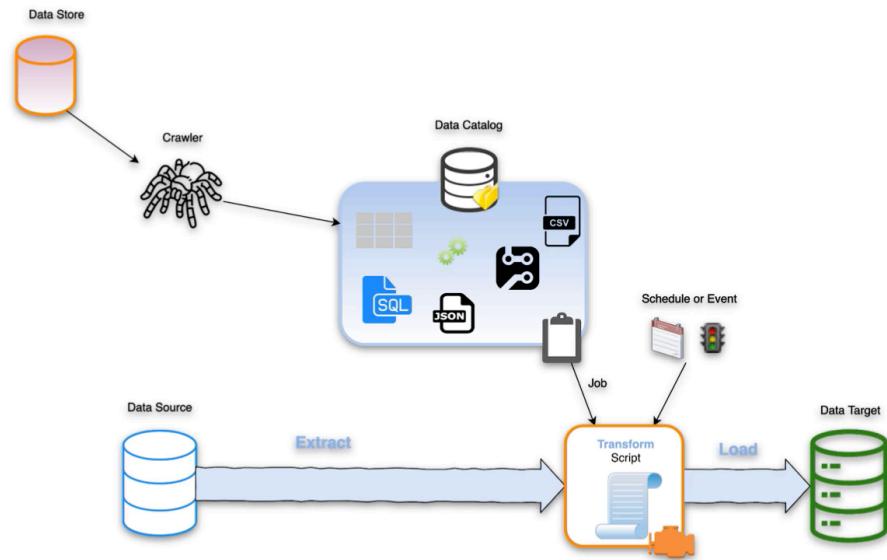
To perform ETL work from a data source to a data target

- points crawler to data store
- crawler populates Glue data catalog with metadata table definition

- AWS Glue can generate a script to transform data or we can use the console.
- we can run the job on demand or on trigger (time based, schedule, trigger event)
- the script runs in an Apache Spark environment

## AWS Machine Learning Exploratory Data Analysis

### AWS Glue Concepts



### AWS Glue Key Facts

- A fully managed ETL service for categorizing, cleaning, enriching, and moving your data
- Glue components
  - Central metadata repository: Glue Catalog
  - ETL engine that automatically generates python or scala code
  - Flexible scheduler for dependency resolution, job monitoring, and retries
- Serverless
- Can convert semi-structured schemas to relational-schemas on the fly

## AWS Glue Terminology

- ❑ Data Catalog: persistent metadata store
- ❑ Classifier: determines the schema of your data
- ❑ Connection: the properties required to connect to data store
- ❑ Crawler: connects to a data store and steps through prioritized list of classifiers to determine schema
- ❑ Database: set of associated data catalog table definitions
- ❑ Data store: repository for persistently storing data
- ❑ Data source: data store used as input to transformation
- ❑ Data target: data store that a transformation writes to
- ❑ Job: ETL logic
- ❑ Table: metadata definition that represents your data
- ❑ Transform: code logic to change your data into a different format



## AWS Glue Components

- ❑ Console: define and orchestrate ETL workflows
- ❑ Data Catalog: persistent metadata store
- ❑ Crawlers and Classifiers: crawlers scan data and classify it
- ❑ ETL Operations: using metadata in the data catalog, autogenerated python or scala code
- ❑ Jobs System: managed infrastructure to orchestrate your ETL workflow

### AWS Glue - Lab

Take data from 1 format (parquet) and transform it to CSV

Using adult data set from census data

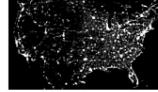


**Machine Learning Repository**  
Center for Machine Learning and Intelligent Systems

## Adult Data Set

Download: [Data Folder](#), [Data Set Description](#)

**Abstract:** Predict whether income exceeds \$50K/yr based on census data. Also known as "Census Income" dataset.



Data Set Characteristics:	Multivariate	Number of Instances:	48842	Area:	Social
Attribute Characteristics:	Categorical, Integer	Number of Attributes:	14	Date Donated	1996-05-01
Associated Tasks:	Classification	Missing Values?	Yes	Number of Web Hits:	1703343

```
hi_avail_cdk_stack_notepad.py • GlueTransforms.py | adult-census.data | adult.names | app_notepad.py
1 age,workclass,fnlwgt,education,education-num,marital-status,occupation,relationship,race,sex,capital-gain,capital-lo:arnings
2 39, State-gov, 77516, Bachelors, 13, Never-married, Adm-clerical, Not-in-family, White, Male, 2174, 0, 40, United-States, <=50K
3 50, Self-emp-not-inc, 83311, Bachelors, 13, Married-civ-spouse, Exec-managerial, Husband, White, Male, 0, 0, 13, United-States, <=50K
4 38, Private, 215646, HS-grad, 9, Divorced, Handlers-cleaners, Not-in-family, White, Male, 0, 0, 40, United-States, <=50K
5 53, Private, 234721, 11th, 7, Married-civ-spouse, Handlers-cleaners, Husband, Black, Male, 0, 0, 40, United-States, <=50K
6 28, Private, 338409, Bachelors, 13, Married-civ-spouse, Prof-specialty, Wife, Black, Female, 0, 0, 40, Cuba, <=50K
7 37, Private, 284582, Masters, 14, Married-civ-spouse, Exec-managerial, Wife, White, Female, 0, 0, 40, United-States, <=50K
8 49, Private, 160187, 9th, 5, Married-spouse-absent, Other-service, Not-in-family, Black, Female, 0, 0, 16, Jamaica, <=50K
```

Data is available in S3 as parquet file

Name	Last modified	Size	Storage class
glue-output	--	--	--
glue-scripts	--	--	--
glue-temp	--	--	--
adult-census.data	Jan 18, 2020 2:32:55 PM GMT-0500	3.8 MB	Standard
part-00000-314eaf64-e415-4a31-be3c-01adcd1352fd-c000.snappy.parquet	Jan 18, 2020 3:16:57 PM GMT-0500	128.0 KB	Standard

First create an IAM role or Glue

**Identity and Access Management (IAM)**

- [Dashboard](#)
- [Access management](#)
  - [Groups](#)
  - [Users](#)
  - Roles**
  - [Policies](#)
  - [Identity providers](#)
  - [Account settings](#)
- [Access reports](#)
  - [Access analyzer](#)
  - [Archive rules](#)
  - [Analyzer details](#)
- [Credential report](#)
- [Organization activity](#)

**Roles**

**What are IAM roles?**

IAM roles are a secure way to grant permissions to entities that you trust. Examples of entities include the following:

- IAM user in another account
- Application code running on an EC2 instance that needs to perform actions on AWS resources
- An AWS service that needs to act on resources in your account to provide its features
- Users from a corporate directory who use identity federation with SAML

IAM roles issue keys that are valid for short durations, making them a more secure way to grant access.

**Additional resources:**

- [IAM Roles FAQ](#)
- [IAM Roles Documentation](#)
- [Tutorial: Setting Up Cross Account Access](#)
- [Common Scenarios for Roles](#)

[Create role](#)
[Delete role](#)

Use Glue Service

## Choose the service that will use this role

### EC2

Allows EC2 instances to call AWS services on your behalf.

### Lambda

Allows Lambda functions to call AWS services on your behalf.

API Gateway	CodeBuild	EKS	KMS	RoboMaker
AWS Backup	CodeDeploy	EMR	Kinesis	S3
AWS Chatbot	CodeStar Notifications	ElastiCache	Lambda	SMS
AWS Support	Comprehend	Elastic Beanstalk	Lex	SNS
Amplify	Config	Elastic Container Service	License Manager	SWF
AppStream 2.0	Connect	Elastic Transcoder	Machine Learning	SageMaker
AppSync	DMS	Elastic Load Balancing	Macie	Security Hub
Application Auto Scaling	Data Lifecycle Manager	Forecast	MediaConvert	Service Catalog
Application Discovery Service	Data Pipeline	Global Accelerator	Migration Hub	Step Functions
Batch	DataSync	Glue	OpsWorks	Storage Gateway
...	DeepLens	Greengrass	Personalize	Textract

Then go to permissions

[Cancel](#) [Next: Permissions](#)

Attach 2 managed policies to the role:

- AWSGlueServiceRole
- AmazonS3FullAccess

## Create role

1 2 3 4

### ▼ Attach permissions policies

Choose one or more policies to attach to your new role.

Create policy		
Filter policies	Search	Showing 673 results
	Policy name	Used as
<input type="checkbox"/>	AccessAnalyzerServiceRolePolicy	None
<input type="checkbox"/>	addQuickSight	Permissions policy (1)
<input type="checkbox"/>	AdministratorAccess	Permissions policy (3)
<input type="checkbox"/>	AlexaForBusinessDeviceSetup	None
<input type="checkbox"/>	AlexaForBusinessFullAccess	None
<input type="checkbox"/>	AlexaForBusinessGatewayExecution	None
<input type="checkbox"/>	AlexaForBusinessNetworkProfileServicePolicy	None
<input type="checkbox"/>	AlexaForBusinessPolyDelegatedAccessPolicy	None

Let's give it a name

## Review

Provide the required information below and review this role before you create it.

Role name\*

AWSGlue-census|

Use alphanumeric and '+,-,@-' characters. Maximum 64 characters.

Role description

Allows Glue to call AWS services on your behalf.

Maximum 1000 characters. Use alphanumeric and '+,-,@-' characters.

Trusted entities AWS service: glue.amazonaws.com

Policies

 AWSGlueServiceRole

 AmazonS3FullAccess

## Identity and Access Management (IAM)

Dashboard

▼ Access management

Groups

Users

**Roles**

Policies

Identity providers

Account settings

The role **AWSGlue-census** has been created.

Create role

Delete role

Q Search

Role name	Trusted entities	Last activity
<input type="checkbox"/> AmazonSageMaker-ExecutionRole-20191007T222545	AWS service: sagemaker	37 days
<input type="checkbox"/> AmazonSageMaker-ExecutionRole-20191211T174650	AWS service: sagemaker	18 days
<input type="checkbox"/> AmazonSageMaker-ExecutionRole-20191217T163878	AWS service: sagemaker	31 days

Now add a database - where all the metadata about the megastore will be housed

## AWS Glue

Data catalog

### Databases

Tables

Connections

Crawlers

Classifiers

Settings

Databases A database is a set of associated table definitions, organized into a logical group.

Add database

View tables

Action ▾

Name

Description



You don't have any databases defined in your data catalog.

Add database

The screenshot shows the AWS Glue Data catalog interface. On the left, there's a sidebar with options like Data catalog, Databases, Tables, Connections, Crawlers, Classifiers, Settings, and ETL. Under Databases, 'Add database' is highlighted. A modal window titled 'Add database' is open, showing a 'Database name' input field containing 'census-data'. Below it is a 'Description and location (optional)' section. At the bottom right of the modal is a blue 'Create' button.

The census-data database has no table

The screenshot shows the AWS Glue Data catalog interface. The path 'Databases > census-data' is selected. The main area displays the database details: Name (census-data), Description, and Location. Below this, a section titled 'Tables in census-data' is shown, which is currently empty.

Let's create a table using a crawler

The screenshot shows the AWS Glue Data catalog interface. The path 'Tables > Add tables' is selected. The main area shows a table header with columns: Name, Database, Location, Classification, and Last updated. A message says 'You don't have any tables defined in your data catalog.' Below it is a blue 'Add tables using a crawler' button.

The screenshot shows the 'Add crawler' wizard. The first step, 'Crawler info', is selected. It has a title 'Add information about your crawler'. The 'Crawler name' field contains 'census-crawler'. Below it is a 'Tags, description, security configuration, and classifiers (optional)' section. At the bottom right is a blue 'Next' button. To the left, there's a sidebar with tabs: Crawler info (selected), Crawler source type, Data store, IAM Role, Schedule, Output, and Review all steps.

## Add crawler

Crawler info  
census-crawler

Crawler source type

Data store

IAM Role

Schedule

Output

Review all steps

### Specify crawler source type

Choose Existing catalog tables to specify catalog tables as the crawler source. The selected tables specify the data stores to crawl. This option doesn't support JDBC data stores.

Crawler source type

Data stores

Existing catalog tables

[Back](#) [Next](#)

## Add crawler

Crawler info  
census-crawler

Crawler source type

Data stores

Data store

IAM Role

Schedule

Output

Review all steps

### Add a data store

Choose a data store

S3

S3

JDBC

DynamoDB

Include path

s3://bucket/prefix/object

Exclude patterns (optional)

[Back](#) [Next](#)

## Add crawler

Crawler info  
census-crawler

Crawler source type

Data stores

Data store

IAM Role

Schedule

Output

Review all steps

### Choose S3 path

- machine-learning-exam
  - census-data-csv
  - census-data
  - glue-output
  - glue-scripts
  - glue-temp
  - heart-data
    - FAO-database.csv
    - adult.data
    - heart.csv
    - part-00000-314eaf64-e415-4a31-be3c-01ad1d1352fd-c000.snap
    - s3ManifestFile.json
- marinbloise.com
- mbloise-resume

[Select](#)

We now have the source set up

Crawler info  
census-crawler

Crawler source type  
Data stores

Data store  
S3

IAM Role

Schedule

Output

Review all steps

### Add a data store

Choose a data store

Crawl data in

Specified path in my account  
 Specified path in another account

Include path

Exclude patterns (optional)

Select the role we created earlier to access S3 and other glue services

Crawler info  
census-crawler

Crawler source type  
Data stores

Data store  
S3: s3://machine-le...

IAM Role

Schedule

Output

Review all steps

### Choose an IAM role

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

IAM role i

s3://machine-learning-exam/part-00000-314eaf64-e415-4a31-be3c-01ad1d1352fd-c000.snappy.parquet

Schedule: run it on demand

Crawler info  
census-crawler

Crawler source type  
Data stores

Data store  
S3: s3://machine-le...

IAM Role  
arn:aws:iam::001178  
231653:role/AWSGlue-census

Schedule

Output

Review all steps

### Create a schedule for this crawler

Frequency

Hourly  
Daily  
Choose days  
Weekly  
Monthly  
Custom

Crawler info  
census-crawler

Crawler source type  
Data stores

Data store  
S3: s3://machine-le...

IAM Role

### Create a schedule for this crawler

Frequency

Select the database where to store the metadata

Configure the crawler's output

**Database** census-data

**Add database**

**Prefix added to tables (optional)** Type a prefix added to table names

▶ Grouping behavior for S3 data (optional)

▶ Configuration options (optional)

**Back** **Next**

## Finish

**Add crawler**

**Crawler info** census-crawler

**Crawler source type** Data stores

**Data store** S3: s3://machine-learning-exam/part-00000-314eaf64-e415-4a31-be3c-01ad1d1352fd-c000.snappy.parquet

**IAM Role** arn:aws:iam::001178231653:role/AWSGlue-census

**Schedule** Run on demand

**Output** census-data

**Review all steps**

**Data store** S3  
Include path s3://machine-learning-exam/part-00000-314eaf64-e415-4a31-be3c-01ad1d1352fd-c000.snappy.parquet  
Exclude patterns

**IAM role** IAM role arn:aws:iam::001178231653:role/AWSGlue-census

**Schedule** Schedule Run on demand

**Output** Database census-data  
Prefix added to tables (optional) Create a single schema for each S3 path false  
▼ Configuration options  
Schema updates in the data store Update the table definition in the data catalog.  
Object deletion in the data store Mark the table as deprecated in the data catalog.

**Back** **Finish**

## Now run the crawler to infer the schema

AWS Glue

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 **census-crawler** was created to run on demand. [Run it now?](#)

**Add crawler** **Run crawler** **Action**  Filter by tags and attributes User preferences Showing: 1 - 1

Name	Schedule	Status	Logs	Last runtime	Median runtime	Tables updated	Tables added
census-crawler		Ready		0 secs	0 secs	0	0

Create a crawler to crawl that  
 Best practices for S3 to store temporary files and output the results of glue job:

The screenshot shows the AWS S3 console with the path `Amazon S3 > machine-learning-exam > census-data-csv`. The bucket name is `machine-learning-exam`. The `Overview` tab is selected. A search bar at the top says "Type a prefix and press Enter to search. Press ESC to clear." Below it are buttons for `Upload`, `+ Create folder`, `Download`, and `Actions`. The location is listed as `US East (N. Virginia)`. The table below shows three objects:

Name	Last modified	Size	Storage class
<code>glue-output</code>	--	--	--
<code>glue-scripts</code>	--	--	--
<code>glue-temp</code>	--	--	--

A table was created by our crawler

The screenshot shows the AWS Glue console with the sidebar navigation: `AWS Glue`, `Data catalog`, `Databases`, **Tables**, `Connections`. The main area shows the `Tables` section with the following details:

**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**, **Database : census-data**, **Save view**, **Showing: 1 - 1**.

Name	Database	Location	Classification	Last updated	Deprecate
<code>part_00000_314ef64_e415_4a31_be3c_01ad1d1352fd_c000...</code>	census-data	s3://machine-learning-exa...	parquet	18 January 2020 4:09 PM U...	
<code>part_00000_314ef64_e415_4a31_be3c_01ad1d1352fd_c000...</code>					

**Table properties**: `averageRecordSize`: 4, `CrawlerSchemaDeserializerVersion`: 1.0, `compressionType`: none, `typeOfData`: file.

**Schema**: `Showing: 1 - 15 of 15`.

Column name	Data type	Partition key	Comment
1 age	bigint		
2 workclass	string		
3 fnlwgt	bigint		
4 education	string		
5 education-num	bigint		
6 marital-status	string		
7 occupation	string		
8 relationship	string		
9 race	string		
10 sex	string		
11 capital-gain	bigint		
12 capital-loss	bigint		
13 hours-per-week	bigint		
14 native-country	string		
15 earnings	string		

Classifiers:

AWS Glue

Data catalog

Databases

Tables

Connections

Crawlers

| Classifiers

Settings

ETL

Workflows

Jobs

ML Transforms

Triggers

Classifiers A classifier determines the schema of your data. You can use the AWS Glue built-in classifiers or write your own.

Add classifier Action ▾

Classifier

Add classifier

Classifier name

Classifier type  Grok  XML  JSON  CSV

JSON path

The JSON path expression defines a JSON structure and is used to define a table schema.

Create

This screenshot shows the AWS Glue console with the 'Classifiers' section selected. A modal dialog box titled 'Add classifier' is open, prompting the user to enter a classifier name and select its type (JSON is chosen). It also includes a field for defining a JSON path to define a table schema.

Now create a job to transform our data from parquet to CSV

AWS Glue

Data catalog

Databases

Tables

Connections

Crawlers

Classifiers

Settings

ETL

Workflows

| Jobs

ML Transforms

Jobs A job is your business logic required to perform extract, transform and load (ETL) work. Job runs are initiated by triggers which can be scheduled or driven by events.

User preferences

Add job Action ▾ Filter by tags and attributes Showing: 0 - 0 < > ⌂ ⌂

Name	Type	ETL language	Script location	Last modified	Job bookmark

You don't have any jobs defined yet.

Add job

This screenshot shows the AWS Glue console with the 'Jobs' section selected. A message states that no jobs are defined yet, and an 'Add job' button is available to start creating one.

## Add job

- Job properties
- Data source
- Transform type
- Data target
- Schema

### Configure the job properties

#### Name

#### IAM role i



#### Type

#### Glue version

#### This job runs

- A proposed script generated by AWS Glue i
- An existing script that you provide
- A new script to be authored by you

#### Script file name

#### S3 path where the script is stored



#### Temporary directory i



## Update the 2 target directories

#### Script file name

#### S3 path where the script is stored



#### Temporary directory i



## Pick data source: parquet file

- Job properties
- census-csv
- Data source
- Transform type
- Data target
- Schema

### Choose a data source

Showing: 1 - 1 < >

Name	Database	Location	Classification
part_00000_314eaf64_e415_4a31_be3c....	census-data	s3://machine-learning-exam/part-00000-...	parquet

## Transform

**Choose a transform type**

- Change schema  
Change schema of your source data and create a new target dataset
- Find matching records  
Use machine learning to find matching records within your source data
- Remove duplicate records  
When records match, the record with the lowest primary key value survives.

To note Glue can help find matching records using ML

**Choose a transform type**

- Change schema  
Change schema of your source data and create a new target dataset
- Find matching records  
Use machine learning to find matching records within your source data
- Remove duplicate records  
When records match, the record with the lowest primary key value survives.

**Worker type** ⓘ

Standard

Jobs containing ML transforms work best with newer instance types. We recommend the G.2X worker type for ML transforms.

**Maximum capacity** ⓘ

10

Target - we will use S3

**Choose a data target**

- Create tables in your data target
- Use tables in the data catalog and update your data target

**Data store**

Amazon S3

**Format**

- JSON
- JSON
- CSV
- Avro
- Parquet
- ORC

Back      Next

**Add job**

Job properties  
census-csv

Data source  
part\_00000\_314eaf64\_e415\_4a31\_be3c\_01ad1d1352fd\_c000\_snappy\_parquet

Transform type  
Change schema

Data target  
s3://machine-learning-exam/census-data-csv/glue-output

Schema

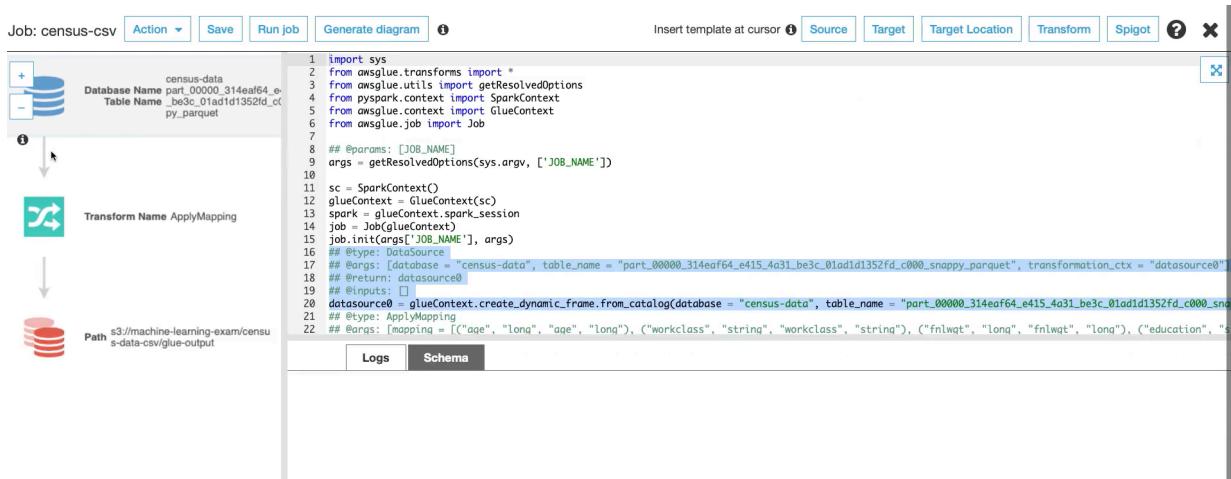
Map the source columns to target columns.

Verify the mappings created by AWS Glue. Change mappings by choosing other columns with **Map to target**. You can **Clear** all mappings and **Reset** to default AWS Glue mappings. AWS Glue generates your script with the defined mappings.

Source	Target			
Column name	Data type	Map to target	Column name	Data type
age	bigint	age	age	long
workclass	string	workclass	workclass	string
fnlwgt	bigint	fnlwgt	fnlwgt	long
education	string	education	education	string
education-num	bigint	education-num	education-num	long
marital-status	string	marital-status	marital-status	string
occupation	string	occupation	occupation	string
relationship	string	relationship	relationship	string
race	string	race	race	string
sex	string	sex	sex	string
capital-gain	bigint	capital-gain	capital-gain	long
capital-loss	bigint	capital-loss	capital-loss	long
hours-per-week	bigint	hours-per-week	hours-per-week	long

**Add column** **Clear** **Reset**

## We now have a simple data pipeline



```

10
11 sc = SparkContext()
12 glueContext = GlueContext(sc)
13 spark = glueContext.spark_session
14 job = Job(glueContext)
15 job.init(args['JOB_NAME'], args)
16 ## @type: DataSource
17 ## @args: [database = "census-data", table_name = "part_00000_314eaf64_e415_4a31_be3c_01ad1d1352fd_c000_snappy_parquet", transformation_ctx = "datasource0"]
18 ## @return: datasource0
19 ## @inputs: []
20 datasource0 = glueContext.create_dynamic_frame.from_catalog(database = "census-data", table_name = "part_00000_314eaf64_e415_4a31_be3c_01ad1d1352fd_c000_snappy_parquet")
21 ## @type: ApplyMapping
22 ## @args: [mapping = [{"age": "long", "age": "long"}, {"workclass": "string", "workclass": "string"}, {"fnlwgt": "long", "fnlwgt": "long"}, {"education": "string", "education": "string"}], transformation_ctx = "applymapping1"]
23 ## @return: applymapping1
24 ## @inputs: [frame = datasource0]
25 applymapping1 = ApplyMapping.apply(frame = datasource0, mappings = [{"age": "long", "age": "long"}, {"workclass": "string", "workclass": "string"}, {"fnlwgt": "long", "fnlwgt": "long"}, {"education": "string", "education": "string"}], transformation_ctx = "applymapping1")
26 ## @type: DataSink
27 ## @args: [connection_type = "s3", connection_options = {"path": "s3://machine-learning-exam/census-data-csv/glue-output"}, format = "csv", transform = datasink2]
28 ## @return: datasink2
29 ## @inputs: [frame = applymapping1]
30 datasink2 = glueContext.write_dynamic_frame.from_options(frame = applymapping1, connection_type = "s3", connection_options = {"path": "s3://machine-learning-exam/census-data-csv/glue-output"}, format = "csv")
31 iob.commit()

```

To note we can add more transform steps in this code

The screenshot shows the AWS Glue Job Editor interface. On the left, there's a preview of the job code:

```

census-data
Database Name part_00000_314eaf64_e
Table Name _be3c_01ad1d1352fd_c
py_parquet

```

Below the code, it says "Transform Name ApplyMapping" and "Path s3://machine-learning-exam/census-data-csv/glue-output".

A modal window titled "Add transform" is open on the right, listing several transform types with their descriptions:

Name	Description
<input type="radio"/> ApplyMapping	Apply mapping to a DynamicFrame
<input type="radio"/> DropFields	Drop fields from a DynamicFrame
<input type="radio"/> DropNullFields	DynamicFrame without null fields.
<input type="radio"/> Filter	Builds a new DynamicFrame by selecting records from the input frame that satisfy the predicate function
<input type="radio"/> FindMatches	Builds a new DynamicFrame that detects records that refer to the same real-world entity based on your trained ML Transform
<input type="radio"/> Join	Join two DynamicFrames
<input type="radio"/> Map	Builds a new DynamicFrame by applying a function to all records in the input DynamicFrame
<input type="radio"/> MapToCollection	Apply a transform to each DynamicFrame in this DynamicFrameCollection
<input type="radio"/> Relationalize	Flatten nested schema and pivot out array columns from the flattened frame
<input type="radio"/> RenameField	Rename a field within a DynamicFrame

Now run job

The screenshot shows the AWS Glue Job Editor interface. On the left, there's a preview of the job code:

```

census-data
Database Name part_00000_314eaf64_e
Table Name _be3c_01ad1d1352fd_c
py_parquet

```

Below the code, it says "Transform Name ApplyMapping" and "Path s3://machine-learning-exam/census-data-csv/glue-output".

A modal window titled "Parameters (optional)" is open on the right, containing the following sections:

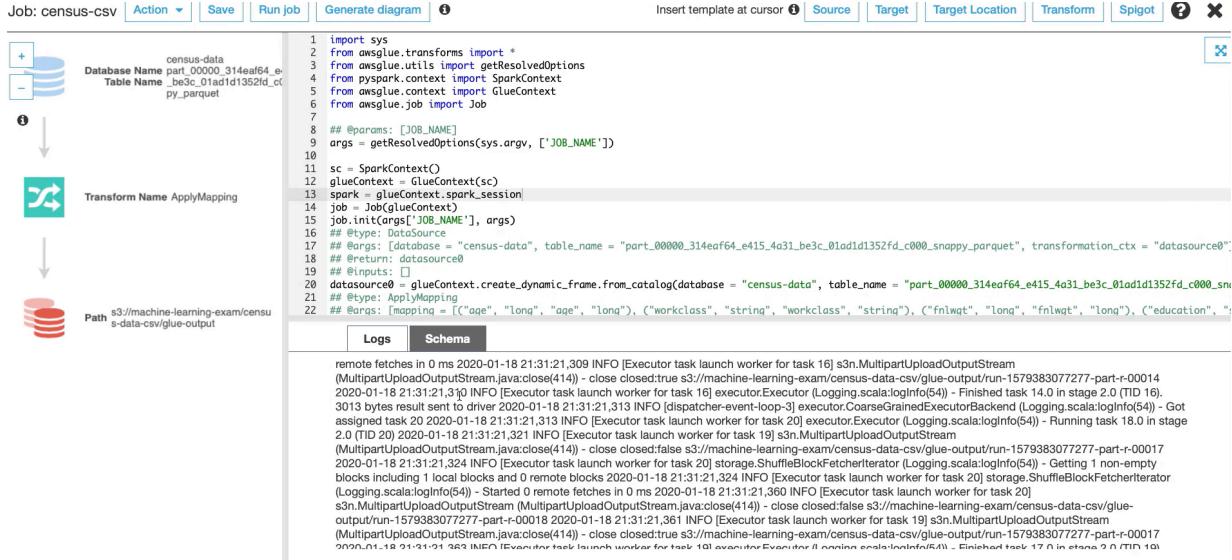
- Advanced properties
- Monitoring options
- Tags
- Security configuration, script libraries, and job parameters

At the bottom of the modal, it says "Only job **census-csv** is run. Jobs dependent on the completion of job **census-csv** will not be run. To run a job and trigger dependent jobs, define an on-demand trigger." A "Run job" button is at the bottom right.

=> no resource to configure or manage.

Glue will take care of that for us (server less) - setting up in the background  
Apache Spark jobs

When it's done, we can see the logs



In s3 we can now see the csv files that were created

machine-learning-exam

Overview

Q Type a prefix and press Enter to search. Press ESC to clear.

Upload + Create folder Download Actions ▾

US East (N. Virginia)

Viewing 1 to 20

<input type="checkbox"/>	Name ▾	Last modified ▾	Size ▾	Storage class ▾
<input type="checkbox"/>	run-1579383077277-part-r-00000	Jan 18, 2020 4:31:21 PM GMT-0500	196.1 KB	Standard
<input type="checkbox"/>	run-1579383077277-part-r-00001	Jan 18, 2020 4:31:21 PM GMT-0500	196.7 KB	Standard
<input type="checkbox"/>	run-1579383077277-part-r-00002	Jan 18, 2020 4:31:21 PM GMT-0500	196.7 KB	Standard
<input type="checkbox"/>	run-1579383077277-part-r-00003	Jan 18, 2020 4:31:21 PM GMT-0500	196.3 KB	Standard

It chunked up the info into 20 files.

```

39," Private",""," Some-college",""," Divorced"," Handlers-cleaners"," Not-in-family"," White"," Male",""," United-States"
45," Private",""," HS-grad",""," Married-civ-spouse"," Handlers-cleaners"," Husband"," White"," Male",""," United-States"," <=50K"
33," Local-gov",""," Some-college",""," Never-married"," Adm-clerical"," Unmarried"," Black"," Female",""," United-States"," <=50K"
46," Federal-gov",""," Some-college",""," Divorced"," Machine-op-inspcnt"," Unmarried"," White"," Male",""," United-States"," <=50K"
24," Private",""," 11th",""," Never-married"," Transport-moving"," Own-child"," White"," Male",""," United-States"," <=50K"
57," Local-gov",""," Bachelors",""," Married-civ-spouse"," Exec-managerial"," Husband"," White"," Male",""," United-States"," <=50K"
61," Private",""," HS-grad",""," Married-civ-spouse"," Craft-repair"," Husband"," White"," Male",""," United-States"," <=50K"
24," Private",""," Assoc-voc",""," Married-civ-spouse"," Craft-repair"," Husband"," White"," Male",""," United-States"," <=50K"
54," Local-gov",""," Masters",""," Married-civ-spouse"," Prof-specialty"," Husband"," White"," Male",""," United-States"," >50K"
22," Private",""," Some-college",""," Never-married"," Other-service"," Own-child"," White"," Female",""," United-States"," <=50K"
59," Federal-gov",""," Assoc-acdm",""," Married-civ-spouse"," Adm-clerical"," Husband"," White"," Male",""," United-States"," <=50K"
31," Private",""," HS-grad",""," Never-married"," Sales"," Own-child"," White"," Male",""," United-States"," <=50K"
41," Local-gov",""," Masters",""," Married-civ-spouse"," Prof-specialty"," Husband"," White"," Male",""," United-States"," <=50K"
47," Private",""," Bachelors",""," Married-civ-spouse"," Sales"," Husband"," Black"," Male",""," United-States"," <=50K"
22," Private",""," HS-grad",""," Never-married"," Adm-clerical"," Own-child"," White"," Female",""," United-States"," <=50K"
39," Self-emp-not-inc",""," 9th",""," Married-civ-spouse"," Craft-repair"," Husband"," White"," Male",""," United-States"," <=50K"
57," Self-emp-not-inc",""," Assoc-acdm",""," Married-civ-spouse"," Other-service"," Husband"," White"," Male",""," Iran"," <=50K"
33," Private",""," Assoc-acdm",""," Married-civ-spouse"," Sales"," Husband"," White"," Male",""," Mexico"," <=50K"
54," ?",""," HS-grad",""," Divorced"," ?"," Other-relative"," White"," Male",""," United-States"," <=50K"
53," State-gov",""," Doctorate",""," Married-civ-spouse"," Prof-specialty"," Husband"," White"," Male",""," United-States"," >50K"
53," Federal-gov",""," HS-grad",""," Married-civ-spouse"," Prof-specialty"," Husband"," White"," Male",""," United-States"," >50K"
37," State-gov",""," Assoc-acdm",""," Divorced"," Prof-specialty"," Unmarried"," White"," Female",""," United-States"," <=50K"
    
```

