

# Udemy - 1 - Data Engineering - part 1

## 1. AWS S3

### AWS S3 Overview



- Amazon S3 allows people to store objects (files) in “buckets” (directories)
- Buckets must have a **globally unique name**
- Objects (files) have a Key. The key is the **FULL** path:
  - <my\_bucket>/[my\\_file.txt](#)
  - <my\_bucket>/[my\\_folder1/another\\_folder/my\\_file.txt](#)
- This will be interesting when we look at partitioning
- Max object size is 5TB
- Object Tags (key / value pair – up to 10) – useful for security / lifecycle

### AWS S3 for Machine Learning

- Backbone for many AWS ML services (example: SageMaker)
  - Create a “Data Lake”
    - Infinite size, no provisioning
    - 99.999999999% durability
    - Decoupling of storage (S3) to compute (EC2, Amazon Athena, Amazon Redshift Spectrum, Amazon Rekognition, and AWS Glue)
  - Centralized Architecture
  - Object storage => supports any file format
  - Common formats for ML: CSV, JSON, Parquet, ORC, Avro, Protobuf
-

# AWS S3 Data Partitioning



- Pattern for speeding up range queries (ex: AWS Athena)
- By Date: [s3://bucket/my-data-set/year/month/day/hour/data\\_00.csv](s3://bucket/my-data-set/year/month/day/hour/data_00.csv)
- By Product: [s3://bucket/my-data-set/product-id/data\\_32.csv](s3://bucket/my-data-set/product-id/data_32.csv)
- You can define whatever partitioning strategy you like!
- Data partitioning will be handled by some tools we use (e.g. AWS Glue)

## S3 Storage Tiers Comparison

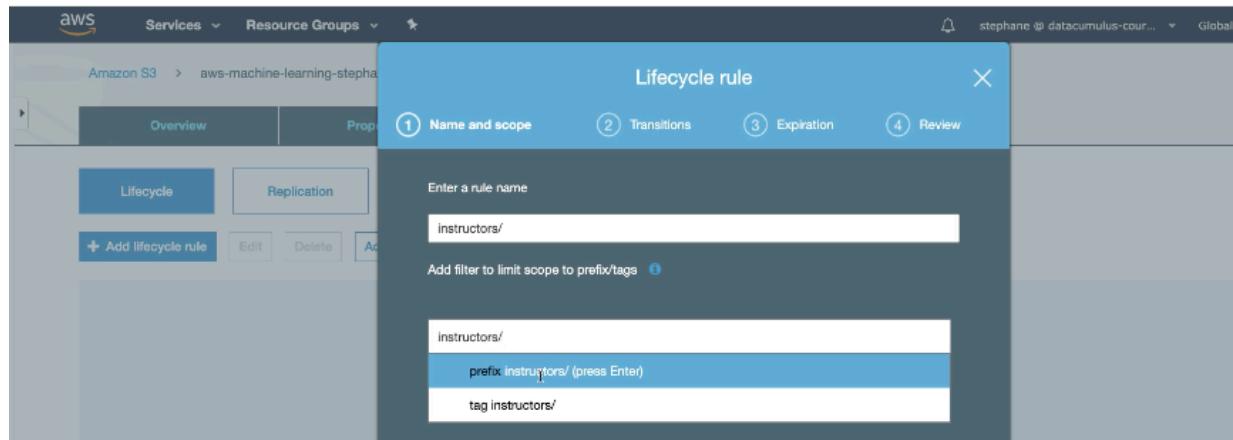


	Standard	Standard - Infrequent Access	One - Infrequent Access	S3 Intelligent-Tiering	Glacier
Durability	99.999999999%	99.999999999%	99.999999999%	99.999999999%	99.999999999%
Availability	99.99%	99.9%	99.5%	99.90%	NA
AZ	≥3	≥3	1	≥3	≥3
Concurrent facility fault tolerance	2	2	0	1	1

Frequently accessed   Infrequently accessed   Intelligent (new!)   Archives

# S3 Lifecycle Rules

- Set of rules to move data between different tiers, to save storage cost
- Example: General Purpose => Infrequent Access => Glacier
- Transition actions: objects are transitioned to another storage class.
  - Move objects to Standard IA class 60 days after creation
  - And move to Glacier for archiving after 6 months
- Expiration actions: S3 deletes expired objects on our behalf
  - Access log files can be set to delete after a specified period of time



**Storage class transition**

There are per-request fees when using lifecycle to transition data to any S3 or S3 Glacier storage class. [Learn more](#) or see [Amazon S3 pricing](#)

Current version    Previous versions

For current versions of objects [+ Add transition](#)

Object creation	Days after creation
Transition to Standard-IA after	30 days
Select a transition	
Transition to Standard-IA after	
Transition to Intelligent-Tiering after	
Transition to One Zone-IA after	
Transition to Glacier after	
Transition to Glacier Deep Archive after	

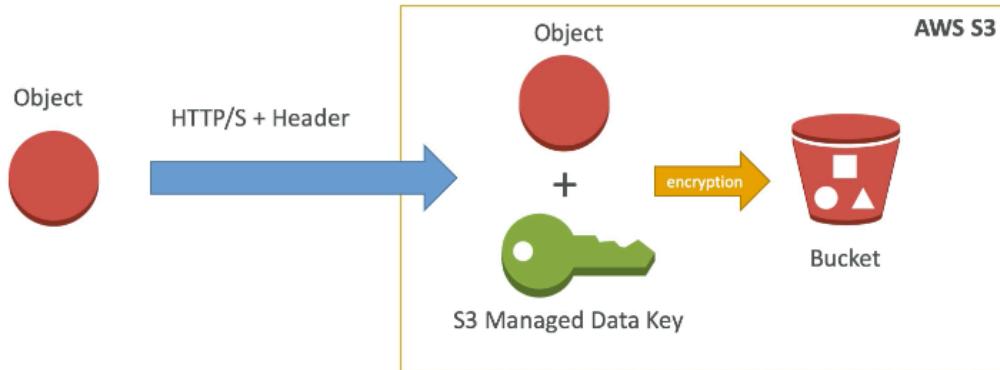
## S3 Encryption for Objects

- There are 4 methods of encrypting objects in S3
- SSE-S3: encrypts S3 objects using keys handled & managed by AWS
- SSE-KMS: use AWS Key Management Service to manage encryption keys
  - Additional security (user must have access to KMS key)
  - Audit trail for KMS key usage
- SSE-C: when you want to manage your own encryption keys
- Client Side Encryption
  - From an ML perspective, SSE-S3 and SSE-KMS will be most likely used

SSE = Server Side Encryption

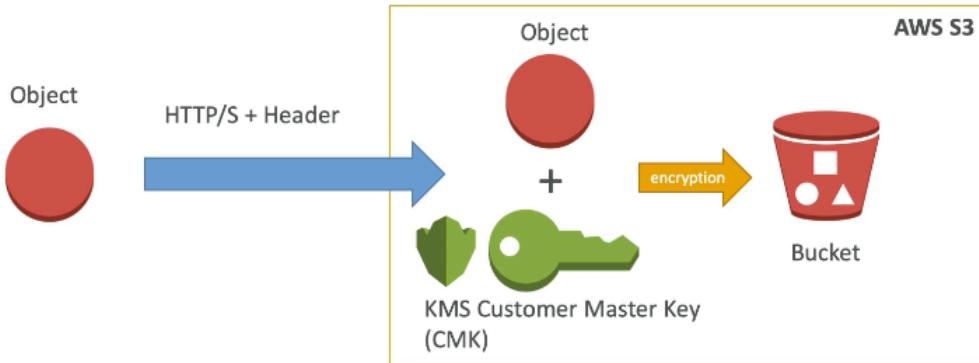
Key managed by S3:

## SSE-S3



Same pattern as SSE-S3, but the key used for encrypting the object is generated with KMS Master key, that WE manage ourselves within AWS

## SSE-KMS



# S3 Security



- User based
  - IAM policies - which API calls should be allowed for a specific user
- Resource Based
  - Bucket Policies - bucket wide rules from the S3 console - allows cross account
  - Object Access Control List (ACL) – finer grain
  - Bucket Access Control List (ACL) – less common

# S3 Bucket Policies



- JSON based policies
  - Resources: buckets and objects
  - Actions: Set of API to Allow or Deny
  - Effect: Allow / Deny
  - Principal: The account or user to apply the policy to
- Use S3 bucket for policy to:
  - Grant public access to the bucket
  - Force objects to be encrypted at upload
  - Grant access to another account (Cross Account)

# S3 Default Encryption vs Bucket Policies

- The old way to enable default encryption was to use a bucket policy and refuse any HTTP command without the proper headers:

```
{  
    "Version": "2012-10-17",  
    "Id": "PutObjPolicy",  
    "Statement": [  
        {  
            "Sid": "DenyIncorrectEncryptionHeader",  
            "Effect": "Deny",  
            "Principal": "*",  
            "Action": "s3:PutObject",  
            "Resource": "arn:aws:s3:::<bucket_name>/*",  
            "Condition": {  
                "StringNotEquals": {  
                    "s3:x-amz-server-side-encryption": "AES256"  
                }  
            },  
        },  
    ],  
}
```

```
{  
    "Sid": "DenyUnEncryptedObjectUploads",  
    "Effect": "Deny",  
    "Principal": "*",  
    "Action": "s3:PutObject",  
    "Resource": "arn:aws:s3:::<bucket_name>/*",  
    "Condition": {  
        "Null": {  
            "s3:x-amz-server-side-encryption": true  
        }  
    }  
}
```

- The new way is to use the “default encryption” option in S3
- Note: Bucket Policies are evaluated before “default encryption”

Amazon S3 > aws-machine-learning-stephanie > instructors > 2019 > 10 > 23 > instructor-data.csv

instructor-data.csv Latest version ▾

Overview Properties Permissions Select from

Storage class

Use the most appropriate storage class based on frequency of access.

Learn more

Standard

Encryption

None

AES-256 Use Amazon S3 server-side encryption to encrypt your data.

AWS-KMS

Cancel Save

Metadata

Assign optional metadata to the object as a name-value (key-value) pair.

Learn more

1 metadata

Tags

Tag objects to search, organize and manage access

Learn more

Object lock

Prevent this object from being deleted.

Learn more

instructor-data.csv Latest version ▾

Overview Properties Permissions Select from

**Storage class**

Use the most appropriate storage class based on frequency of access.

[Learn more](#)

Standard

**Encryption**

Use encryption to protect your data while in-transit and at rest.

[Learn more](#)

AES-256

**Metadata**

Assign optional metadata to the object as a name-value (key-value) pair.

[Learn more](#)

1 metadata

**Tags**

Tag objects to search, organize and manage access

[Learn more](#)

0 Tags

**Object lock**

Prevent this object from being deleted.

[Learn more](#)

Disabled

In properties, can set default encryption:



Overview

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

[Upload](#) [+ Create folder](#) [Download](#) [Actions ▾](#)

Name	Last modified
<input checked="" type="checkbox"/> instructor-data.csv	Oct 23, 2019 10:22:48 AM GMT+0100

instructor-data.csv

Latest version ▾

Download Copy path Select from

Overview
Key: instructor-data.csv

Size: 150.0 B
Expiration date: Dec 22, 2020 12:00:00 AM GMT+0000

Expiration rule: instructors/
ETag: 2ca1e41ac7f099f1c3c75d36895dbae

Last modified: Oct 23, 2019 10:22:48 AM GMT+0100
Object URL: https://aws-machine-learning-stephanie.s3-eu-west-1.amazonaws.com/instructors/2019/10/23/instructor-data.csv

Properties
Storage class: Standard

Encryption: AES-256
Metadata: 1

Tags: 0
Object lock: Disabled

Permissions
Owner: stephanie

Another way would be to set bucket policy and force encryption

The screenshot shows the 'Bucket Policy' tab selected in the AWS S3 Bucket Policy editor. The ARN of the policy is listed as 'arn:aws:s3:::aws-machine-learning-stephane'. A note at the top states: 'Granting public access in this policy will be blocked because Block public access settings are turned on for this bucket. To determine which settings are turned on, check your Block public access settings.' Below this is a large text area where the policy document is being edited, starting with '1'. At the bottom right are 'Delete', 'Cancel', and 'Save' buttons.

## S3 Security - Other

- Networking - VPC Endpoint Gateway:
  - Allow traffic to stay within your VPC (instead of going through public web)
  - Make sure your private services (AWS SageMaker) can access S3
  - Very important for AWS ML Exam
- Logging and Audit:
  - S3 access logs can be stored in other S3 bucket
  - API calls can be logged in AWS CloudTrail
- Tagged Based (combined with IAM policies and bucket policies)
  - Example: Add tag Classification=PHI to your objects



## 2. AWS Kinesis

Exam will test what is the difference between the 4 Kinesis streams

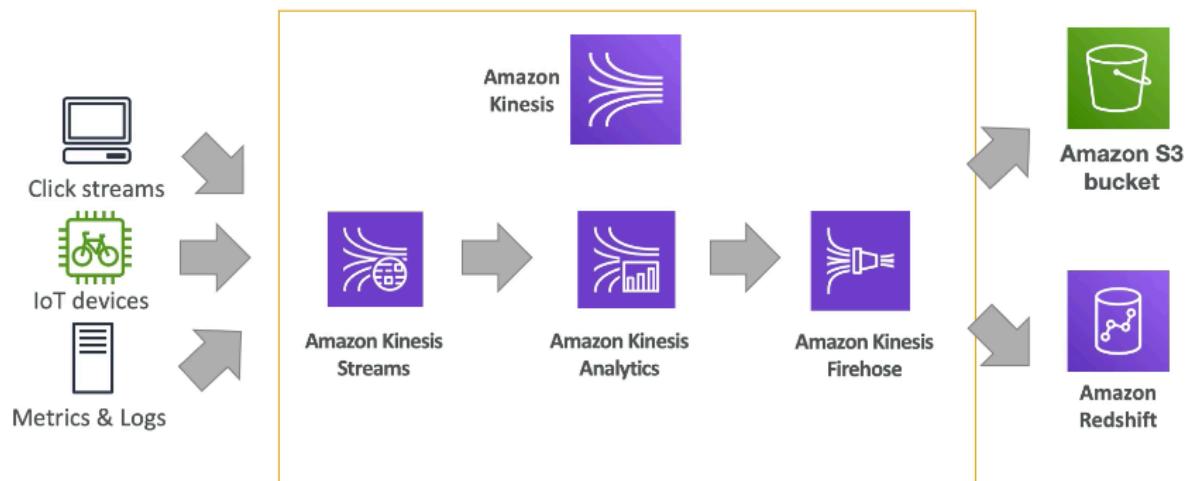
# AWS Kinesis Overview



- Kinesis is a managed alternative to Apache Kafka
  - Great for application logs, metrics, IoT, clickstreams
  - Great for “real-time” big data
  - Great for streaming processing frameworks (Spark, NiFi, etc...)
  - Data is automatically replicated synchronously to 3 AZ
- 
- Kinesis Streams: low latency streaming ingest at scale
  - Kinesis Analytics: perform real-time analytics on streams using SQL
  - Kinesis Firehose: load streams into S3, Redshift, ElasticSearch & Splunk
  - Kinesis Video Streams: meant for streaming video in real-time

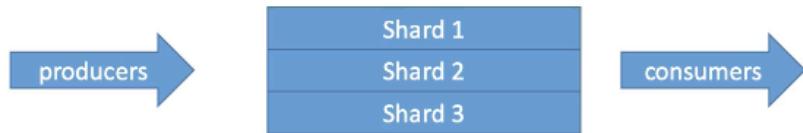
One simple architecture of how things work:

## Kinesis



# Kinesis Streams Overview

- Streams are divided in ordered Shards / Partitions



- Shards have to be provisioned in advance (capacity planning)
- Data retention is 24 hours by default, can go up to 7 days
- Ability to reprocess / replay data
- Multiple applications can consume the same stream
- Once data is inserted in Kinesis, it can't be deleted (immutability)
- Records can be up to 1MB in size

## Kinesis Data Streams Limits to know

- Producer:
  - 1MB/s or 1000 messages/s at write PER SHARD
  - "ProvisionedThroughputException" otherwise
- Consumer Classic:
  - 2MB/s at read PER SHARD across all consumers
  - 5 API calls per second PER SHARD across all consumers
- Data Retention:
  - 24 hours data retention by default
  - Can be extended to 7 days

# Kinesis Data Firehose



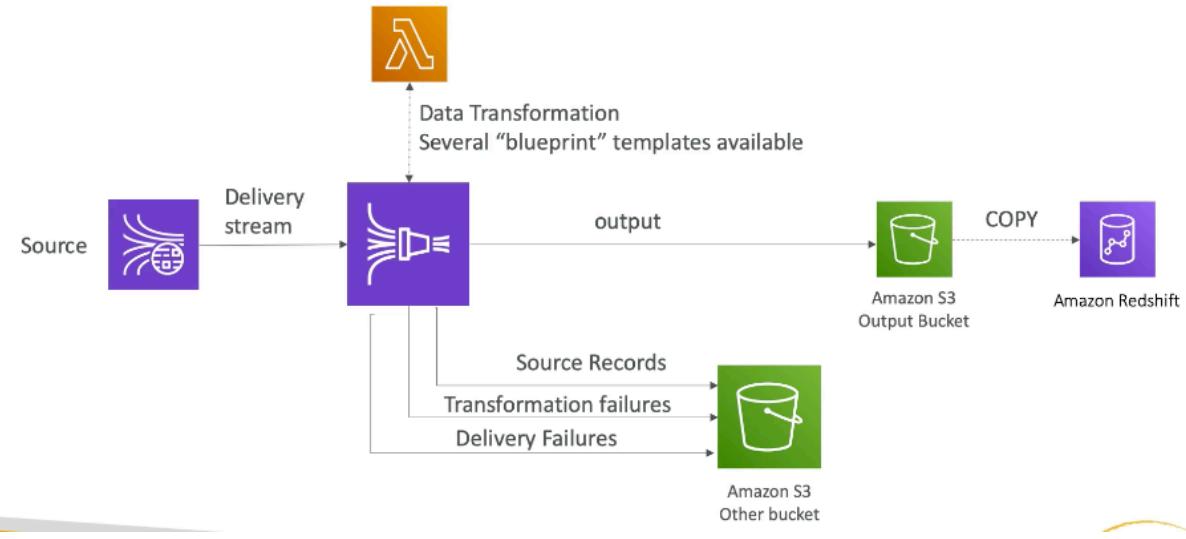
- Fully Managed Service, no administration
- Near Real Time (60 seconds latency minimum for non full batches)
- Data Ingestion into Redshift / Amazon S3 / ElasticSearch / Splunk
- Automatic scaling
- Supports many data formats
- Data Conversions from CSV / JSON to Parquet / ORC (only for S3)
- Data Transformation through AWS Lambda (ex: CSV => JSON)
- Supports compression when target is Amazon S3 (GZIP, ZIP, and SNAPPY)
- Pay for the amount of data going through Firehose

## Kinesis Data Firehose Diagram



Lambda transformation is optional

# Kinesis Data Firehose Delivery Diagram



Streams allows us to build REAL TIME APPLICATIONS that can be replayed

Firehose is an INGESTION SERVICE, fully managed, fully server less, and can do servers transformation with lambda

It is NEAR real time. No need to provision capacity (shards) in advance - auto-scaling. But no data storage

## Kinesis Data Streams vs Firehose

- Streams
  - Going to write custom code (producer / consumer)
  - Real time (~200 ms latency for classic, ~70 ms latency for enhanced fan-out)
  - Must manage scaling (shard splitting / merging)
  - Data Storage for 1 to 7 days, replay capability, multi consumers
- Firehose
  - Fully managed, send to S3, Splunk, Redshift, ElasticSearch
  - Serverless data transformations with Lambda
  - Near real time (lowest buffer time is 1 minute)
  - Automated Scaling
  - No data storage

## Console experience for Kinesis Firehose

## Get started with Amazon Kinesis

To get started, choose an Amazon Kinesis resource to create.

**Ingest and process streaming data with Kinesis streams**

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

**Create data stream**

**Deliver streaming data with Kinesis Firehose delivery streams**

Continuously collect, transform, and load streaming data into destinations such as Amazon S3 and Amazon Redshift.

**Create delivery stream**

**Analyze streaming data with Kinesis analytics applications**

Run continuous analysis on streaming data from Kinesis data streams and Kinesis Firehose delivery streams.

**Create analytics application**

**Ingest and process media streams with Kinesis video streams**

Build applications to process or analyze streaming media.

**Create video stream**



### Kinesis Firehose - Create delivery stream

#### Step 1: Name and source

- Step 2: Process records
- Step 3: Choose a destination
- Step 4: Configure settings
- Step 5: Review

#### New delivery stream

Delivery streams load data, automatically and continuously, to the destinations that you specify. Kinesis Firehose resources are not covered under the [AWS Free Tier](#) and usage-based charges apply. For more information, see [Kinesis Firehose pricing](#). [Learn more](#)

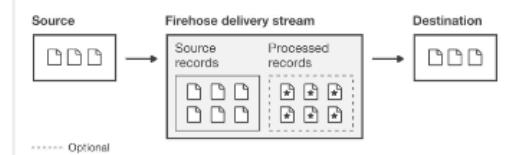
Delivery stream name

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

#### Choose a source

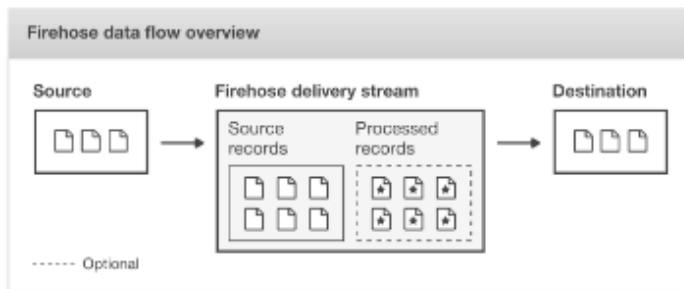
Choose how you would prefer to send records to the delivery stream.

#### Firehose data flow overview



## Choose a source

Choose how you would prefer to send records to the delivery stream.



### Source

#### Direct PUT or other sources

Choose this option to send records directly to the delivery stream, or to send records from AWS IoT, CloudWatch Logs, or CloudWatch Events.

#### Kinesis data stream

### ▼ How to send source records to Kinesis Data Firehose

After creating the delivery stream, send source records using the Firehose PUT API or the Amazon Kinesis Agent.

#### Firehose PUT APIs

Use the Firehose PutRecord() or PutRecordBatch() API to send source records to the delivery stream. [Learn more](#)

#### Amazon Kinesis Agent

The Amazon Kinesis Agent is a stand-alone Java software application that offers an easy way to collect and send source records to Firehose. [Learn more](#)

#### AWS IoT

Create AWS IoT rules that send data from MQTT messages. [Learn more](#)

#### CloudWatch Logs

Use subscription filters to deliver a real-time stream of log events. [Learn more](#)

#### CloudWatch Events

Create rules to indicate which events are of interest to your application and what automated action to take when a rule matches an event. [Learn more](#)

## Lambda transformation

## Kinesis Firehose - Create delivery stream

Step 1: Name and source

Step 2: Process records

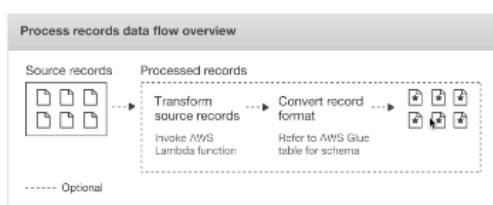
Step 3: Choose a destination

Step 4: Configure settings

Step 5: Review

### Process records

Kinesis Firehose can transform records or convert record format before delivery.



#### Transform source records with AWS Lambda

To return records from AWS Lambda to Kinesis Firehose after transformation, the Lambda function you invoke must be compliant with the required record transformation output model. [Learn more](#)

#### Record transformation

- Disabled  
 Enabled

#### Convert record format

Data in Apache Parquet or Apache ORC format is typically more efficient to query than JSON. Kinesis Data Firehose can convert your JSON-formatted source records using a schema from a table defined in [AWS Glue](#). For records that aren't in JSON format, create a Lambda function that converts them to JSON in the Transform source records with AWS Lambda section above. [Learn more](#)

#### Record format conversion

- Disabled  
 Enabled

If record format conversion is enabled, Firehose can deliver data to Amazon S3 only. Record format conversion will be configured using the OpenX JSON SerDe. For other options use the [AWS CLI](#).

#### Output format

- Apache Parquet  
 Apache ORC

The data is compressed using Snappy compression before it is delivered to S3. To choose another compression method, or to disable data compression, use the AWS CLI. [Learn more](#)

**Specify a schema for source records.** Kinesis Data Firehose references table definitions stored in AWS Glue. Choose an AWS Glue table to specify a schema for your source records. You can [manually create a new table in AWS Glue](#), or [add a crawler in AWS Glue](#) to create a new table using a schema from an existing JSON object in S3. [Learn more](#)

#### AWS Glue region

Choose a region ▾

#### AWS Glue database

Choose a database ▾

## Kinesis Firehose - Create delivery stream

Step 1: Name and source

Step 2: Process records

**Step 3: Choose a destination**

Step 4: Configure settings

Step 5: Review

**Select a destination**

[Learn more](#)

Destination

**Amazon S3**  
Amazon S3 is an easy-to-use object storage, with a simple web service interface to store and retrieve any amount of data from anywhere on the web.

**Amazon Redshift**  
Amazon Redshift is a fast, fully managed, petabyte-scale data warehouse that makes it simple and cost effective to analyze all your data using your existing business intelligence tools.

**Amazon Elasticsearch Service**  
Elasticsearch is an open-source search and analytics engine for use cases such as log analytics, real-time application monitoring, and click stream analytics.

**Splunk**  
Splunk is an operational intelligence tool for analyzing machine-generated data in real-time.

**S3 destination**

Choose a destination in Amazon S3 where your data will be stored. Amazon S3 is object storage built to store and retrieve any amount of data from anywhere.

[Learn more](#)

S3 bucket

aws-machine-learning-stephane [View aws-machine-learning-stephane in S3 console](#)

S3 prefix

By default, Kinesis Data Firehose appends the prefix "YYYY/MM/DD/HH" (in UTC) to the data it delivers to Amazon S3. You can override this default by specifying a custom prefix that includes expressions that are evaluated at runtime.

If your custom prefix doesn't include expressions, Kinesis Data Firehose uses your prefix and appends "YYYY/MM/DD/HH". If your custom prefix includes a Firehose random string or timestamp expression, Kinesis Data Firehose doesn't append "YYYY/MM/DD/HH".

[Learn more](#)

Prefix - optional  
ticker\_demo/

Don't forget the "/" in the prefix

**S3 error prefix**

You can specify an S3 bucket prefix to be used in error conditions. This prefix can include expressions for Kinesis Data Firehose to evaluate at runtime. [Learn more about the rules for specifying prefix expressions](#)

Error prefix - optional  
ticker\_demo\_error/

Buffer condition - deliver the data as soon as possible (near real time),.  
Cannot have a buffer less than 60 seconds

## Kinesis Firehose - Create delivery stream

Step 1: Name and source

Step 2: Process records

Step 3: Choose a destination

**Step 4: Configure settings**

Step 5: Review

### Configure settings

Configure buffer, compression, logging, and IAM role settings for your delivery stream. [Learn more](#)

#### S3 buffer conditions

Firehose buffers incoming records before delivering them to your S3 bucket. Record delivery will be triggered once either of these conditions has been satisfied. [Learn more](#)

Buffer size

1 MB

Enter a buffer size between 1-128 MB

Buffer interval

60 seconds

Enter a buffer interval between 60-900 seconds

Possibility to compress and encrypt if we want

Compression

Disabled

GZIP

Snappy

Zip

S3 encryption

Disabled

Enabled

KMS master key

(Default)aws/s3



Create new IAM role which allow Firehose to send data into S3

## Permissions

IAM role

[Create new or choose](#)

[▼ Hide Details](#)

### Role Summary

Role Description

Provides access to AWS Services and Resources

IAM Role

[Create a new IAM Role](#)

Role Name

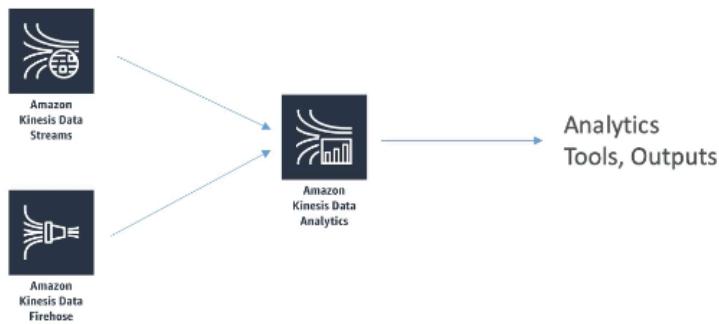
firehose\_delivery\_role

[View Policy Document](#)

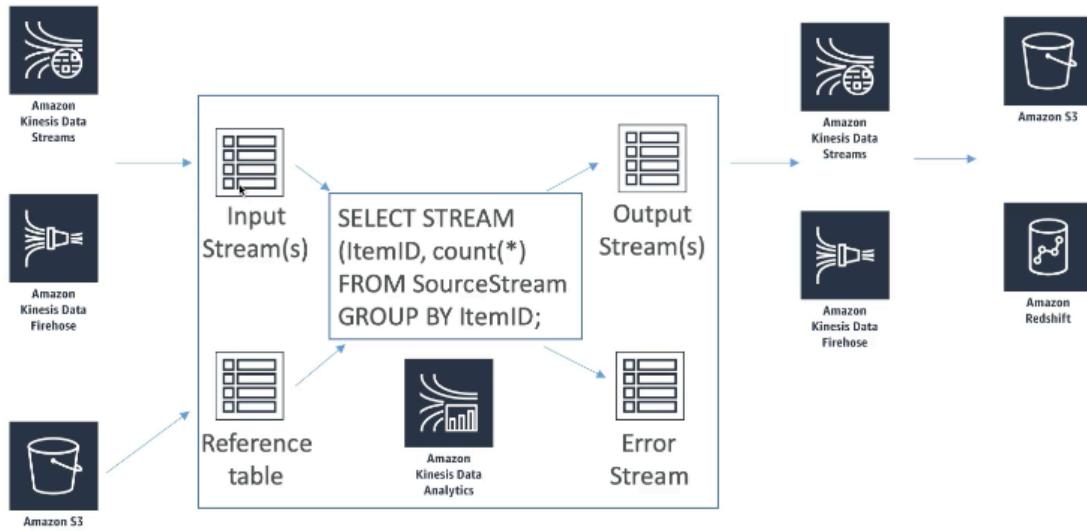
## Kinesis Data Analytics

Takes data from Firehose or streams

## Kinesis Analytics, Conceptually...



## Kinesis Analytics, In more depth...



The above joins input streams to an input lookup table from S3

# Kinesis Data Analytics



- Use cases

- Streaming ETL: select columns, make simple transformations, on streaming data
- Continuous metric generation: live leaderboard for a mobile game
- Responsive analytics: look for certain criteria and build alerting (filtering)

- Features

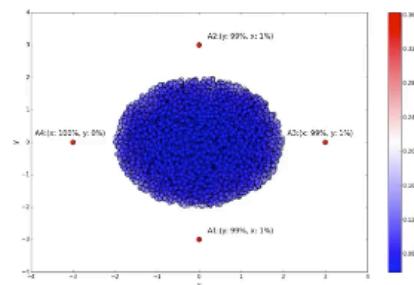
- Pay only for resources consumed (but it's not cheap)
- Serverless; scales automatically
- Use IAM permissions to access streaming source and destination(s)
- SQL or Flink to write the computation
- Schema discovery
- Lambda can be used for pre-processing

Anomalies and Dense areas in Data Analytics

## Machine Learning on Kinesis Data Analytics

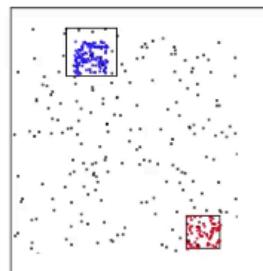
- RANDOM\_CUT\_FOREST

- SQL function used for anomaly detection on numeric columns in a stream
- Example: detect anomalous subway ridership during the NYC marathon
- Uses recent history to compute model



- HOTSPOTS

- locate and return information about relatively dense regions in your data
- Example: a collection of overheated servers in a data center



## Kinesis Data Analytics Lab

**Amazon Kinesis**

- Dashboard
- Data Streams
- Data Firehose
- Data Analytics**
- Video Streams

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*	<input type="text" value="ticker_analytics"/>
Description	<input type="text"/>
Runtime	<input checked="" type="radio"/> SQL <input type="radio"/> Apache Flink 1.6

\* Required      [Cancel](#)      [Create application](#)



### 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](#)

### Choose streaming data source

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

<a href="#">Choose source</a>	<a href="#">Configure a new stream</a>
-------------------------------	--

Source\*  Kinesis stream

A Kinesis 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\*



[Create new](#)

[View ticker\\_demo in Kinesis Firehose](#)

In-application stream name

In your [SQL](#) queries, refer to this source as:

OPTIONAL: Set application role

## Access permissions

Create or choose IAM role with the required permissions. [Learn more](#)

- Access permissions\*  Create / update IAM role **kinesis-analytics-ticker\_analytics-eu-west-1**  
 Choose from IAM roles that Kinesis Analytics can assume

Role to allow Kinesis analytics to perform read on Kinesis Firehose delivery stream

Schema can be automatically inferred

[Edit schema](#) [Retry schema discovery](#)

[Raw](#) [Lambda output](#) **Formatted**

Filter by column name

ticker_symbol VARCHAR(4)	sector VARCHAR(16)	change REAL	price REAL
HJV	ENERGY	-24.26	264.33
HJK	TECHNOLOGY	0.08	5.17
PLM	FINANCIAL	-0.13	18.94
KFU	ENERGY	-3.65	53.63
PJN	RETAIL	-1.06	30.17
QXZ	FINANCIAL	4.3100000000000005	204.03
QAZ	FINANCIAL	1.8900000000000001	205.86
NGC	HEALTHCARE	0.07	4.75
JKL	TECHNOLOGY	-0.16	15.32
MJN	RETAIL	-3.75	159.28

External resources [What's new](#)

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](#)

Source	In-application stream name	ID	Record pre-processing
<a href="#">Firehose delivery stream ticker_demo</a>	SOURCE_SQL_STREAM_001	2.1	Disabled

Reference data (optional)

Enrich data from your streaming data source with JSON or CSV data stored as an object in Amazon S3. Each application can connect to one reference data source. [Learn more](#)

[Connect reference data](#)

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](#)

Destination

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

Amazon Kinesis

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  *
6  * The SQL code you write here will continuously transform your streaming data
7  * when your application is running.
8  *
9  * Get started by clicking "Add SQL From templates" or pull up the
10 * documentation and start writing your own custom queries.
11 */
12
13

```

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

Amazon Kinesis

Kinesis Analytics applications > ticker\_analytics > SQL Template

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

```

-- ** Aggregate (COUNT, AVG, etc.) + Sliding time window **
-- Performs Function on the aggregate rows over a 10 second sliding window for a specified column
-- |-----|-----|-----|
-- | SOURCE | INSERT | DESTIN. |
-- Source->| STREAM | --> & SELECT --> STREAM |-->Destination
-- |-----|-----|-----|
-- STREAM (in-application): a continuously updated entity that you can SELECT from and INSERT into
-- PUMP: an entity used to continuously 'SELECT ... FROM' a source STREAM, and INSERT SQL results into a destination STREAM
-- 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_count BIGINT)
-- 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 sliding window
CREATE OR REPLACE PUMP "STREAM_PUMP" AS INSERT INTO "DESTINATION_SQL_STREAM"
-- COUNT(AVG(MAX(MIN(SUM(STODEV_POP(STODEV_SAMP|VAR_POP|VAR_SAMP))
-- OVER(TEN_SECOND_SLIDING_WINDOW AS ticker_symbol_count
-- FROM "SOURCE_SQL_STREAM_001"
-- WINDOW TEN_SECOND_SLIDING_WINDOW AS (
-- PARTITION BY ticker_symbol
-- RANGE INTERVAL '10' SECOND PRECEDING));

```

Cancel (return to the editor) Add this SQL to the editor

In-application streams:

<input checked="" type="radio"/> DESTINATION_SQL_STREAM	<input type="checkbox"/> REAM	<b>Pause results</b>	→ New results are added every 2-10 seconds. The results below are sampled. <a href="#">?</a>
<input type="radio"/> error_stream	<input type="checkbox"/> Scroll to bottom when new results arrive.		

ROWTIME	ticker_symbol	TICKER_SYMBOL_COUNT
2019-10-23 10:07:43.966	PJN	1
2019-10-23 10:07:48.968	WAS	5
2019-10-23 10:07:48.968	QXZ	4
2019-10-23 10:07:48.968	BAC	2
2019-10-23 10:07:48.968	AZL	3
2019-10-23 10:07:48.968	WSB	2
2019-10-23 10:07:48.968	MMB	4
2019-10-23 10:07:48.968	NOC	3
2019-10-23 10:07:48.968	WSB	3
2019-10-23 10:07:48.968	ABC	5
2019-10-23 10:07:48.968	BAC	2

Now connect a destination to send the resulting aggregation via Firehose

Source    Real-time Analytics    **Destination**

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

[Connect to a destination](#)

### Kinesis Firehose - Create delivery stream

- Step 1: Name and source**
- Step 2: Process records
- Step 3: Choose a destination
- Step 4: Configure settings
- Step 5: Review

#### New delivery stream

Delivery streams load data, automatically and continuously, to the destinations that you specify. Kinesis Firehose resources are not covered under the AWS Free Tier [?](#), and usage-based charges apply. For more information, see [Kinesis Firehose pricing](#) [?](#). [Learn more](#) [?](#)

Delivery stream name

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

#### Choose a source

Choose how you would prefer to send records to the delivery stream.

#### Firehose data flow overview

Source    Firehose delivery stream    Destination

## Kinesis Firehose - Create delivery stream

Step 1: Name and source

Step 2: Process records

### Step 3: Choose a destination

Step 4: Configure settings

Step 5: Review

#### Select a destination

[Learn more](#)

##### Destination

**Amazon S3**

Amazon S3 is an easy-to-use object storage, with a simple web service interface to store and retrieve any amount of data from anywhere on the web.

**Amazon Redshift**

Amazon Redshift is a fast, fully managed, petabyte-scale data warehouse that makes it simple and cost effective to analyze all your data using your existing business intelligence tools

**Amazon Elasticsearch Service**

Elasticsearch is an open-source search and analytics engine for use cases such as log analytics, real-time application monitoring, and click stream analytics

**Splunk**

Splunk is an operational intelligence tool for analyzing machine-generated data in real-time

[Endpoints for KDF data flow connectors](#)

Kinesis Analytics applications > ticker\_analytics > Destination

### Connect to destination

**Destination\***  **Kinesis stream**  
A Kinesis 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.

**AWS Lambda function**  
AWS Lambda is a compute service that lets you run code without provisioning or managing servers.

**Kinesis Firehose delivery stream\***

Retrieving details

#### In-application stream

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

**Connect in-application stream**  **Choose an existing in-application stream**

**Specify a new in-application stream name**

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

**In-application stream name\***

DESTINATION\_SQL\_STREAM  
error\_stream

**Output format**

**JSON**

**CSV**

Kinesis Analytics applications > ticker\_analytics

**ticker\_analytics**

Application status: RUNNING

Application ARN: arn:aws:kinesisanalytics:eu-west-1:387124123361:application/ticker\_analytics

Application version ID: 4 ⓘ

Application metrics: View in CloudWatch Metrics ⓘ

**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](#)

Source	In-application stream name	ID ⓘ	Record pre-processing ⓘ
Firehose delivery stream <a href="#">ticker_demo</a> ⓘ	SOURCE_SQL_STREAM_001	2.1	Disabled

**Name** Last modified

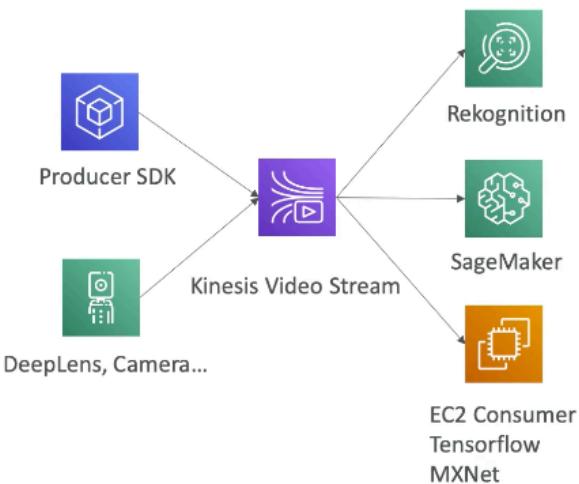
- instructors --
- ticker\_analytics** --
- ticker\_demo --

```
{"ticker_symbol": "QAZ", "TICKER_SYMBOL_COUNT": 5}, {"ticker_symbol": "BAC", "TICKER_SYMBOL_COUNT": 2}, {"ticker_symbol": "QWE", "TICKER_SYMBOL_COUNT": 8}, {"ticker_symbol": "QXZ", "TICKER_SYMBOL_COUNT": 12}, {"ticker_symbol": "BAC", "TICKER_SYMBOL_COUNT": 3}, {"ticker_symbol": "BFH", "TICKER_SYMBOL_COUNT": 8}, {"ticker_symbol": "ABC", "TICKER_SYMBOL_COUNT": 4}, {"ticker_symbol": "AZL", "TICKER_SYMBOL_COUNT": 6}, {"ticker_symbol": "ALY", "TICKER_SYMBOL_COUNT": 7}, {"ticker_symbol": "PLM", "TICKER_SYMBOL_COUNT": 5}, {"ticker_symbol": "BAC", "TICKER_SYMBOL_COUNT": 4}, {"ticker_symbol": "HJV", "TICKER_SYMBOL_COUNT": 8}, {"ticker_symbol": "CRM", "TICKER_SYMBOL_COUNT": 4}, {"ticker_symbol": "TBV", "TICKER_SYMBOL_COUNT": 6}, {"ticker_symbol": "CRM", "TICKER_SYMBOL_COUNT": 5}, {"ticker_symbol": "NFLX", "TICKER_SYMBOL_COUNT": 2}, {"ticker_symbol": "AMZN", "TICKER_SYMBOL_COUNT": 10}, {"ticker_symbol": "ALY", "TICKER_SYMBOL_COUNT": 8}, {"ticker_symbol": "AAPL", "TICKER_SYMBOL_COUNT": 3}, {"ticker_symbol": "AAPL", "TICKER_SYMBOL_COUNT": 4}, {"ticker_symbol": "IOP", "TICKER_SYMBOL_COUNT": 7}, {"ticker_symbol": "MMB", "TICKER_SYMBOL_COUNT": 3}, {"ticker_symbol": "CRM", "TICKER_SYMBOL_COUNT": 6}, {"ticker_symbol": "HJV", "TICKER_SYMBOL_COUNT": 9}, {"ticker_symbol": "JYB", "TICKER_SYMBOL_COUNT": 2}, {"ticker_symbol": "VVS", "TICKER_SYMBOL_COUNT": 3}, {"ticker_symbol": "ARCL", "TICKER_SYMBOL_COUNT": 5}, {"ticker_symbol": "RELU", "TICKER_SYMBOL_COUNT": 0}
```

## Kinesis Video Stream

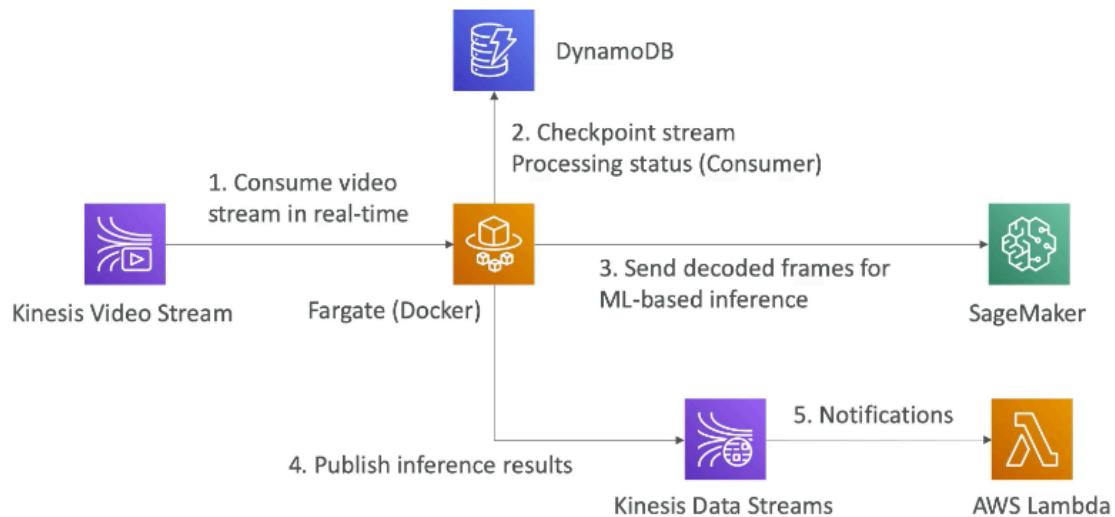
# Kinesis Video Stream

- Producers:
  - security camera, body-worn camera, AWS DeepLens, smartphone camera, audio feeds, images, RADAR data, RTSP camera.
  - One producer per video stream
- Video playback capability
- Consumers
  - build your own (MXNet, Tensorflow)
  - AWS SageMaker
  - Amazon Rekognition Video
- Keep data for 1 hour to 10 years



## Kinesis Video Streams use cases

<https://aws.amazon.com/blogs/machine-learning/analyze-live-video-at-scale-in-real-time-using-amazon-kinesis-video-streams-and-amazon-sagemaker/>



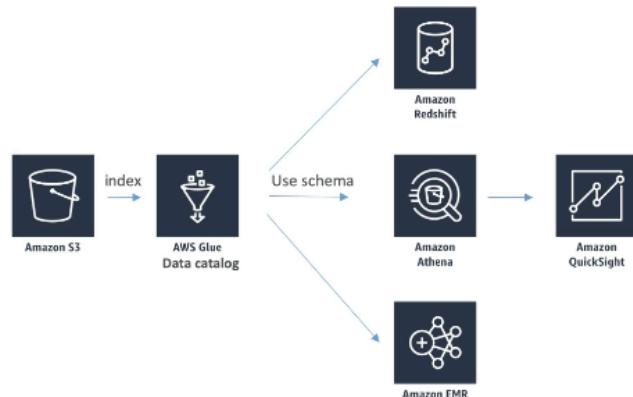
# Kinesis Summary – Machine Learning

- Kinesis Data Stream: create real-time machine learning applications
- Kinesis Data Firehose: ingest massive data near-real time
- Kinesis Data Analytics: real-time ETL / ML algorithms on streams
- Kinesis Video Stream: real-time video stream to create ML applications

## 3. Glue

### Glue Data Catalog

- Metadata repository for all your tables
  - Automated Schema Inference
  - Schemas are versioned
- Integrates with Athena or Redshift Spectrum (schema & data discovery)
- Glue Crawlers can help build the Glue Data Catalog

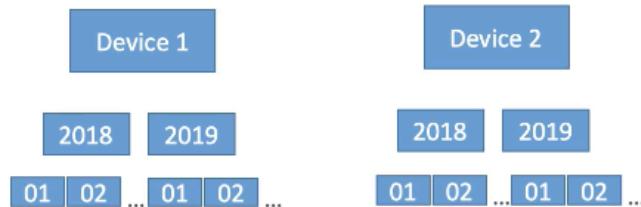


# Glue Data Catalog - Crawlers

- Crawlers go through your data to infer schemas and partitions
- Works JSON, Parquet, CSV, relational store
- Crawlers work for: S3, Amazon Redshift, Amazon RDS
- Run the Crawler on a Schedule or On Demand
- Need an IAM role / credentials to access the data stores

## Glue and S3 Partitions

- Glue crawler will extract partitions based on how your S3 data is organized
- Think up front about how you will be querying your data lake in S3
- Example: devices send sensor data every hour
  - Do you query primarily by **time ranges**?
    - If so, organize your buckets as s3://my-bucket/dataset/**yyyy/mm/dd/device**
  - Do you query primarily by **device**?
    - If so, organize your buckets as s3://my-bucket/dataset/**device/yyyy/mm/dd**



## Glue Lab

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.

Add crawler Run crawler Action Filter by tags and attributes Showing: 0 - 0 User preferences

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

You don't have any crawlers yet. Add crawler

### Add crawler

Crawler info demo\_crawler\_s3

Crawler source type Data stores

Data store

IAM Role

Schedule

Output

Review all steps

Add a data store

Choose a data store S3

Crawl data in

Specified path in my account

Specified path in another account

Include path s3://aws-machine-learning-stephane

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

Exclude patterns (optional)

Back Next

### Add crawler

Crawler info demo\_crawler\_s3

Crawler source type Data stores

Data store S3: s3://aws-machi...

IAM Role AWSGlueServiceRole

Schedule

Output

Review all steps

Choose an IAM role

The IAM role allows the crawler to run and access your Amazon S3 data stores. [Learn more](#)

Update a policy in an IAM role

Choose an existing IAM role

Create an IAM role

IAM role [AWSGlueServiceRole](#) demo

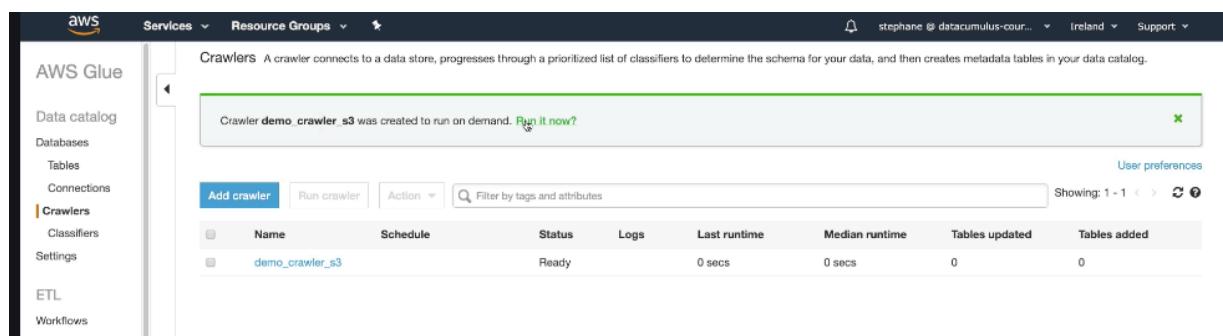
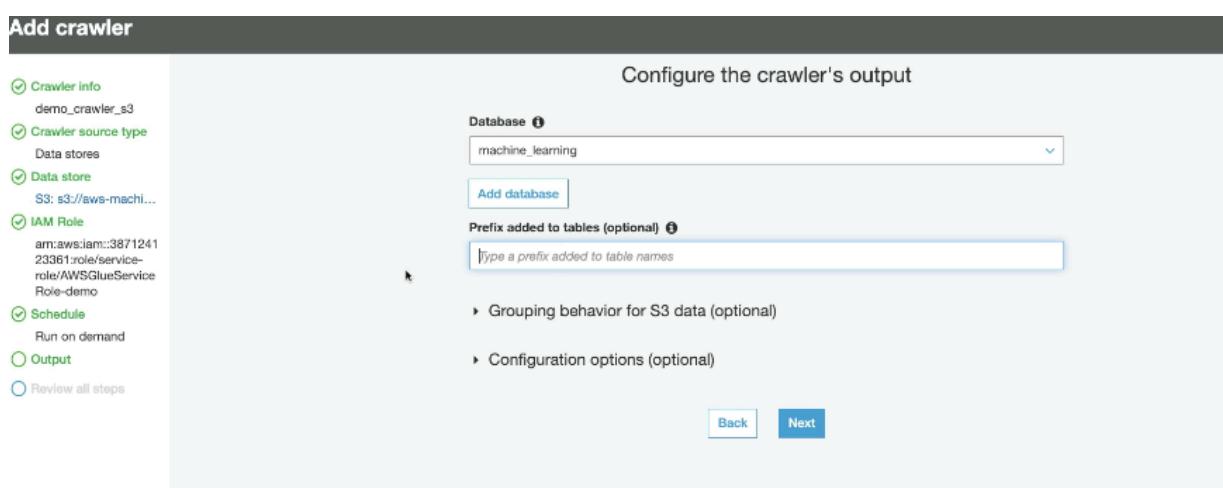
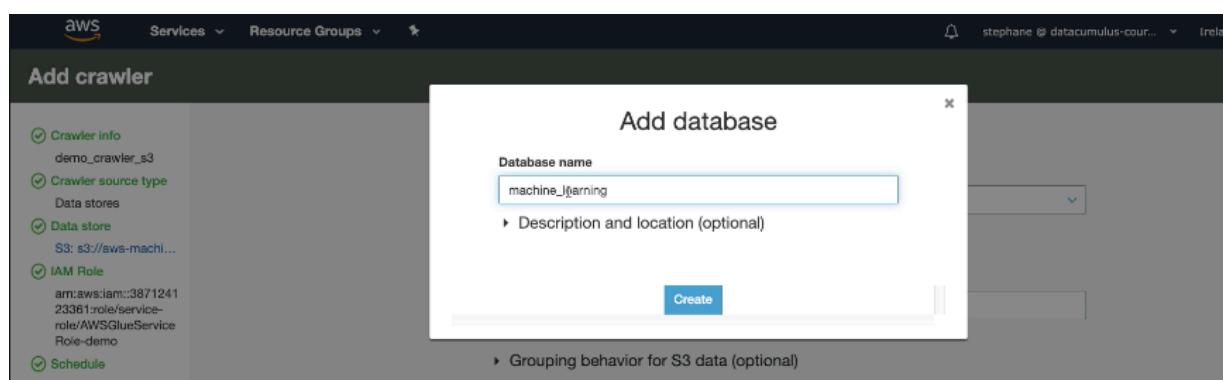
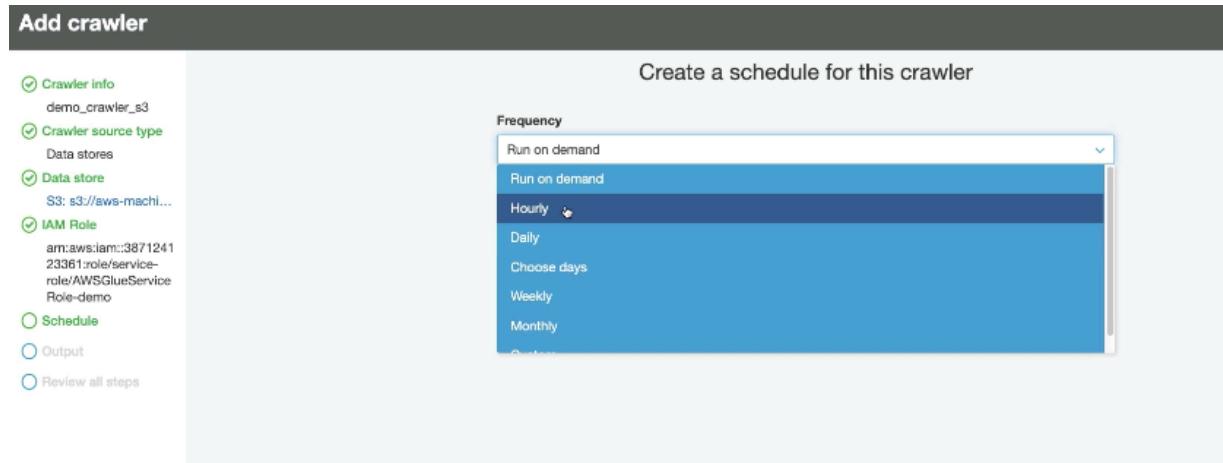
To create an IAM role, you must have **CreateRole**, **CreatePolicy**, and **AttachRolePolicy** permissions.

Create an IAM role named "**AWSGlueServiceRole-rolename**" and attach the AWS managed policy, **AWSGlueServiceRole**, plus an inline policy that allows read access to:

- s3://aws-machine-learning-stephane/

You can also create an IAM role on the [IAM console](#).

Back Next



The crawler now runs against this bucket

Amazon S3 > aws-machine-learning-stephane

Overview Properties Permissions Management

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

Upload Create folder Download Actions EU (Ireland) Viewing 1 to 3

Name	Last modified	Size	Storage class
instructors	--	--	--
ticker_analytics	--	--	--
ticker_demo	--	--	--

Viewing 1 to 3

3 tables were added

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.

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

Name	Schedule	Status	Logs	Last runtime	Median runtime	Tables updated	Tables added
demo_crawler_s3		Stopping		1 min	1 min	0	3

AWS Glue

Services Resource Groups

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: machine\_learning Filter or search for tables... Save view Showing: 1 - 3

Name	Database	Location	Classification	Last updated	Deprecated
instructors	machine_learning	s3://aws-machine-learning-stephane/instructors/	csv	23 October 2019 12:07 ...	
ticker_analytics	machine_learning	s3://aws-machine-learning-stephane/ticker_analytics/	json	23 October 2019 12:07 ...	
ticker_demo	machine_learning	s3://aws-machine-learning-stephane/ticker_demo/	json	23 October 2019 12:07 ...	

Ex: ticker\_demo table

Tables > ticker\_demo

Last updated 23 Oct 2019 Table Version (Current version) ▾

[Edit table](#) [Delete table](#) [View partitions](#) [Compare versions](#) [Edit schema](#)

Name	ticker_demo
Description	
Database	macfine_learning
Classification	json
Location	s3://aws-machine-learning-stephanie/ticker_demo/
Connection	
Deprecated	No
Last updated	Wed Oct 23 12:07:41 GMT+100 2019
Input format	org.apache.hadoop.mapred.TextInputFormat
Output format	org.apache.hadoop.hive.io.HiveIgnoreKeyTextOutputFormat
Serde serialization lib	org.openx.data.jsonserde.JsonSerDe
Serde parameters	paths change,price,sector,ticker_symbol
	sizeKey 275633 objectCount 16 UPDATED_BY_CRAWLER demo_crawler_s3 CrawlerSchemaSerializerVersion 1.0
Table properties	recordCount 3876 averageRecordSize 71 CrawlerSchemaDeserializerVersion 1.0 compressionType none
	typeOfData file

Schema

Showing: 1 - 8 of 8 < >

Column name	Data type	Partition key	Comment
1 ticker_symbol	string		
2 sector	string		
3 change	double		
4 price	double		
5 partition_0	string	Partition (0)	
6 partition_1	string	Partition (1)	
7 partition_2	string	Partition (2)	
8 partition_3	string	Partition (3)	

To note it has figured out we have partitions (corresponding to our folders)

Amazon S3 > aws-machine-learning-stephanie > ticker\_demo > 2019 > 10 > 23 > 09

partition_0	partition_1	partition_2	partition_3		
2019	10	23	10	<a href="#">View files</a>	<a href="#">View properties</a>
2019	10	23	09	<a href="#">View files</a>	<a href="#">View properties</a>

To note we can edit the schema

Column name	Data type	Key	Comment
1 ticker_symbol	string		1
2 sector	string		1
3 change	double		1
4 price	double		1
5 year	string	Partition (0)	1
6 month	string	Partition (1)	1
7 day	string	Partition (2)	1
8 hour	string	Partition (3)	1

To help make the queries more intituitive

## Glue ETL

- # Glue ETL
- Transform data, Clean Data, Enrich Data (before doing analysis)
    - Generate ETL code in Python or Scala, you can modify the code
    - Can provide your own Spark or PySpark scripts
    - Target can be S3, JDBC (RDS, Redshift), or in Glue Data Catalog
  - Fully managed, cost effective, pay only for the resources consumed
  - Jobs are run on a serverless Spark platform
  - Glue Scheduler to schedule the jobs
  - Glue Triggers to automate job runs based on “events”

# Glue ETL - Transformations

- Bundled Transformations:
  - DropFields, DropNullFields – remove (null) fields
  - Filter – specify a function to filter records
  - Join – to enrich data
  - Map - add fields, delete fields, perform external lookups
- Machine Learning Transformations:
  - FindMatches ML: identify duplicate or matching records in your dataset, even when the records do not have a common unique identifier and no fields match exactly.
- Format conversions: CSV, JSON, Avro, Parquet, ORC, XML

## Glue ETL Lab

The screenshot shows the AWS Glue Jobs interface. On the left, there's a sidebar with navigation links for Data catalog, Databases, Tables, Connections, Crawlers, Classifiers, Settings, ETL, Workflows, and Security. Under ETL, the 'Jobs' link is selected. The main area has a header 'Jobs' with a sub-instruction '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.' Below the header is a search bar with 'Add job' and 'Action' buttons, and a 'Filter by tags and attributes' dropdown. A table lists columns: Name, Type, ETL language, Script location, Last modified, and Job bookmark. A message 'You don't have any jobs defined yet.' is displayed above a large 'Add job' button. At the top right, there are 'User preferences' and a 'Showing: 0 - 0' status indicator.

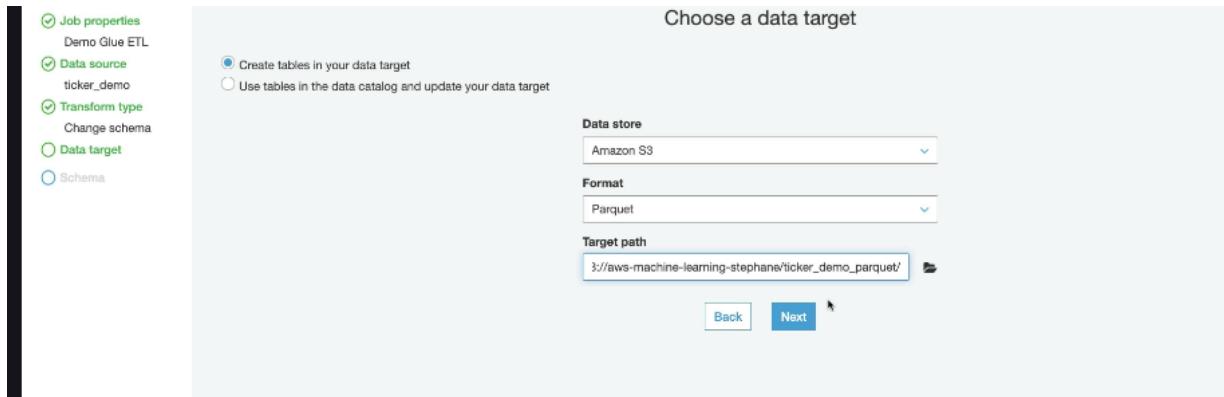
The screenshot shows the 'Add job' configuration dialog. At the top, it says 'Demo Glue ETL'. On the left, there's a sidebar with tabs: 'Job properties' (selected), 'Data source', 'Transform type', 'Data target', and 'Schema'. The main area contains several input fields:

- IAM role:** A dropdown menu set to 'GlueETLDemo'.
- Type:** A dropdown menu set to 'Spark'.
- Glue version:** A dropdown menu set to 'Spark 2.2, Python 2 (Glue version 0.9)'.
- This job runs:** A section with three radio buttons:
  - A proposed script generated by AWS Glue
  - An existing script that you provide
  - A new script to be authored by you
- Script file name:** A text input field containing 'Demo Glue ETL'.
- S3 path where the script is stored:** A text input field.

2 Options:

- Change Schema
- Or deduce with ML

We will use change Schema for this lab



**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	
ticker_symbol	string	ticker_symbol	ticker_symbol	string	X ↴ ↺
sector	string	sector	sector	string	X ↴ ↺
change	double	change	change	double	X ↴ ↺
price	double	price	price	double	X ↴ ↺
year	string	year	year	string	X ↴ ↺
month	string	month	month	string	X ↴ ↺
day	string	day	day	string	X ↴ ↺
hour	string	hour	hour	string	X ↴ ↺

Now an entire ETL Glue job was generated for us

Job Demo Glue ETL was added. Edit and save your Python script.

Job: Demo Glue ETL Action Save Run job Generate diagram ⚙

Insert template at cursor Source Target Target Location Transform Spigot ? X

```

1 import sys
2 from awsglue.transforms import *
3 from awsglue.utils import getResolvedOptions
4 from pyspark.context import SparkContext
5 from awsglue.context import GlueContext
6 from awsglue.job import Job
7
8 ## @params: [JOB_NAME]
9 args = getResolvedOptions(sys.argv, ['JOB_NAME'])
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 = "machine_learning", table_name = "ticker_demo", transformation_ctx = "datasource0"]
18 ## @return: datasource0
19 ## @inputs: []
20 datasource0 = glueContext.create_dynamic_frame.from_catalog(database = "machine_learning", table_name = "ticker_demo", transformation_ct
21

```

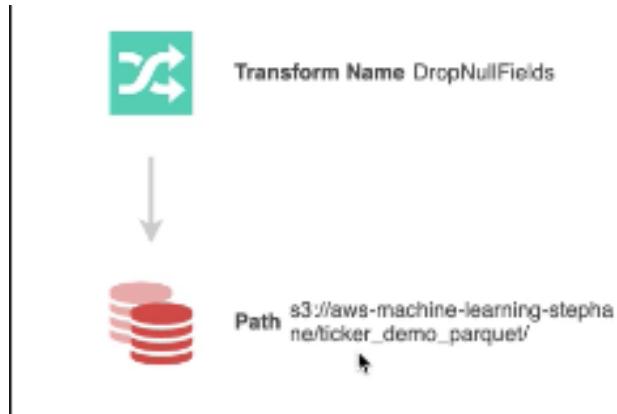
Logs Schema

Transform Name Database Name machine Learning Table Name ticker demo

Transform Name ApplyMapping

Transform Name ResolveChoice

Transform Name DropNullFields



On the right side, there is his spark code, that we can edit

Now run the job

The screenshot shows the AWS Glue job editor interface. On the left, there is a flowchart of the ETL job with three steps: 'Database Name machine\_learning' (with 'Table Name ticker\_demo'), 'Transform Name ApplyMapping', and 'Transform Name ResolveChoice'. On the right, there is a panel titled "Parameters (optional)" containing sections for "Advanced properties", "Monitoring options", "Tags", and "Security configuration, script libraries, and job parameters". A note at the bottom states: "Only job **Demo Glue ETL** is run. Jobs dependent on the completion of job **Demo Glue ETL** will not be run. To run a job and trigger dependent jobs, define an on-demand trigger." At the bottom right of the panel is a blue "Run Job" button.

We can now see the parquet files in s3

The screenshot shows the Amazon S3 console. The path is "Amazon S3 > aws-machine-learning-stephanie". The "Overview" tab is selected. The "Actions" dropdown menu is open, showing options like "Upload", "Create folder", "Download", and "Actions". The table below lists objects in the "ticker\_demo\_parquet" folder:

Name	Last modified	Size	Storage class
instructors	--	--	--
ticker_analytics	--	--	--
ticker_demo	--	--	--
<b>ticker_demo_parquet</b>	--	--	--
ticker_demo_parquet_Shorter\$	Oct 23, 2019 12:38:02 PM GMT+0100	0 B	Standard

If we crawl again the same s3 bucket, the new parquet files are detected and a new table is added to the catalog

AWS Glue Data catalog

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

Name	Database	Location	Classification	Last updated
instructors	machine_learning	s3://aws-machine-learning-stephanie/instructors/	csv	23 October 2019 1
ticker_analytics	machine_learning	s3://aws-machine-learning-stephanie/ticker_analytics/	json	23 October 2019 1
ticker_demo	machine_learning	s3://aws-machine-learning-stephanie/ticker_demo/	json	23 October 2019 1
<b>ticker_demo_parquet</b>	machine_learning	s3://aws-machine-learning-stephanie/ticker_demo_p...	parquet	23 October 2019 1

AWS Glue Data catalog

**Tables > ticker\_demo\_parquet**

Last updated 23 Oct 2019 Table Version (Current version) ▾

Name	Description	Database	Classification	Location	Connection	DDeprecated	Last updated	Input format	Output format	Serde serialization lib	Serde parameters
ticker_demo_parquet		machine_learning	parquet	s3://aws-machine-learning-stephanie/ticker_demo_parquet/		No	Wed Oct 23 12:40:09 GMT+100 2019	org.apache.hadoop.hive.serde2.parquet.MapredParquetInputFormat	org.apache.hadoop.hive.serde2.parquet.MapredParquetOutputFormat	org.apache.hadoop.hive.serde2.parquet.serde.ParquetHiveSerDe	serialization.format 1

Table properties

sizeKey	77796	objectCount	16	UPDATED BY CRAWLER	demo crawler s3	CrawlerSchemaSerializerVersion	1.0	recordCount	3840
averageRecordSize	19	CrawlerSchemaDeserializerVersion	1.0	compressionType	none	typeOfData	file		

Schema

Column name	Data type	Partition key	Comment
1	string		
2	string		
3	double		

## Athena Lab

We can find the 4 tables previously created by Glue Crawler in catalog

Database machine\_learning

Tables (4)

- instructors (Partitioned)
  - instructor\_name (string)
  - course (string)
  - love\_meter (bigint)
- ticker\_analytics (Partitioned)
- ticker\_demo (Partitioned)
- ticker\_demo\_parquet**

Views (0)

Results

```
1 SELECT * FROM "machine_learning"."instructors" limit 10;
```

instructor_name	course	love_meter	partition_0	partition_1	partition_2
Stephane Maarek	AWS Certified Machine Learning Specialty	100	2019	10	23
Frank Kane	AWS Certified Machine Learning Specialty	100	2019	10	23

We can also run SQL query against JSON

The screenshot shows a database interface with a sidebar on the left and a main query editor on the right.

**Database:** machine\_learning

**Tables (4):**

- instructors (Partitioned)
  - instructor\_name (string)
  - course (string)
  - love\_meter (bigint)
- ticker\_analytics (Partitioned)
- ticker\_demo (Partitioned)
- ticker\_demo\_parquet

**Views (0):**

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

**New query 1** | **New query 2** | **New query 3** | **New query 4** | **New query 5** | **New query 6**

```
SELECT * FROM "machine_learning"."ticker_analytics" limit 10;
```

**Run query** | **Save as** | **Create** | (Run time: 2.83 seconds, Data scanned: 62.88 KB)

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

**Results**

	ticker_symbol	ticker_symbol_count	partition_0	partition_1	partition_2	partition_3
1	WFC	1	2019	10	23	10
2	QXZ	2	2019	10	23	10
3	QAZ	5	2019	10	23	10
4	VVS	1	2019	10	23	10

Or parquet