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
- By Product: 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

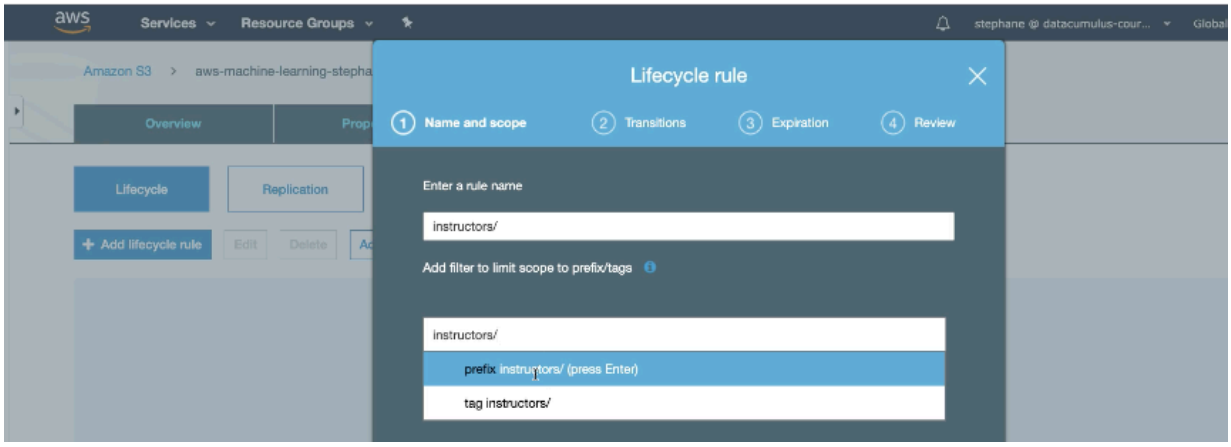
Frequently accessed | Infrequently accessed | Intelligent (new!) | Archives

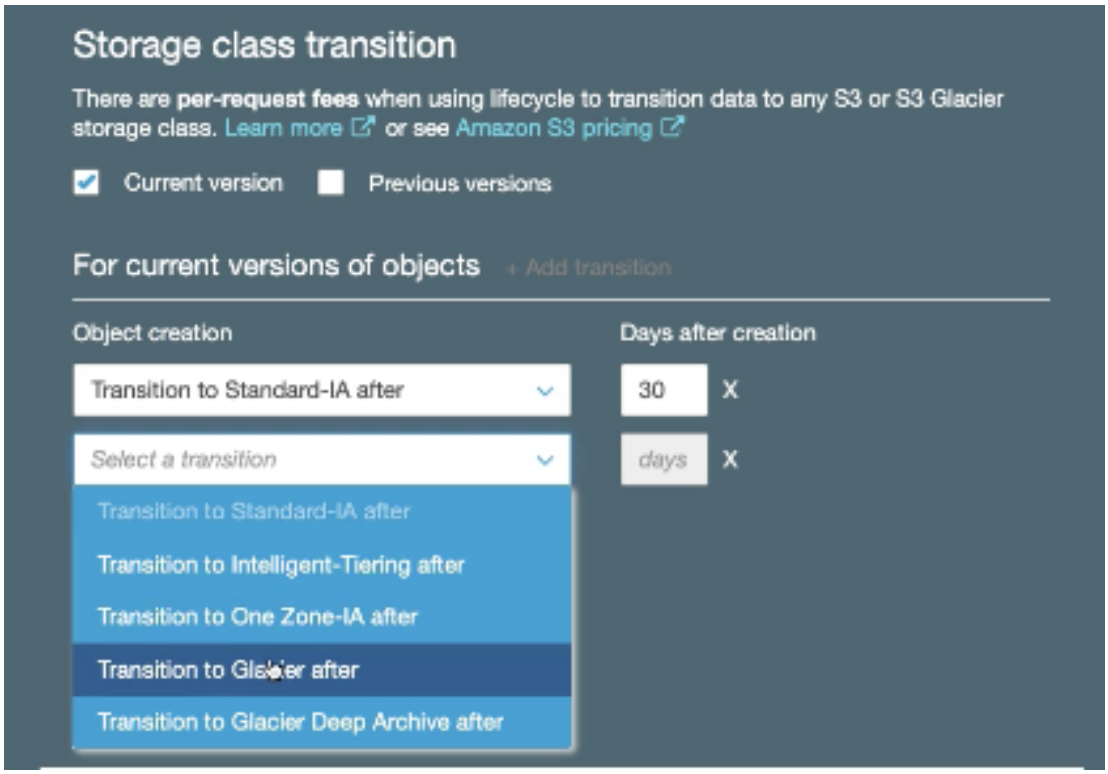
S3 Lifecycle Rules

- 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





S3 Encryption for Objects

- There are 4 methods of encrypting objects in S3

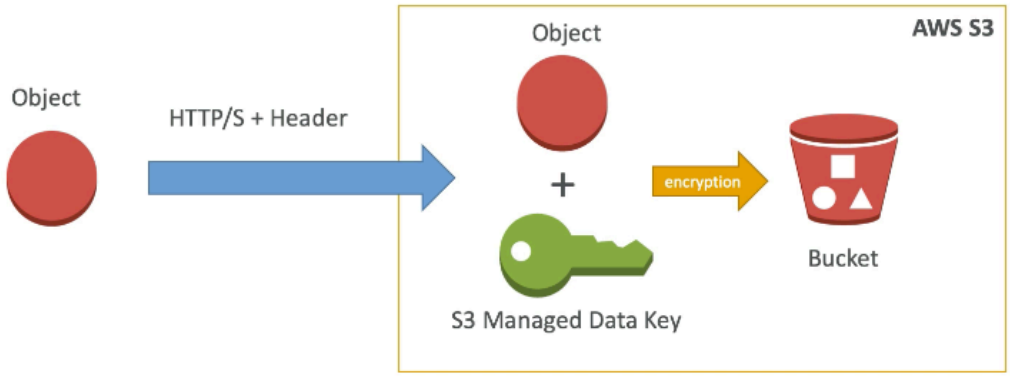
- SSE-S3: encrypts S3 objects using keys handled & managed by AWS
- SSE-KMS: use AWS Key Management Service to manage encryption keys
  - Additional security (user must have access to KMS key)
  - Audit trail for KMS key usage
- SSE-C: when you want to manage your own encryption keys
- Client Side Encryption

- From an ML perspective, SSE-S3 and SSE-KMS will be most likely used

SSE = Server Side Encryption

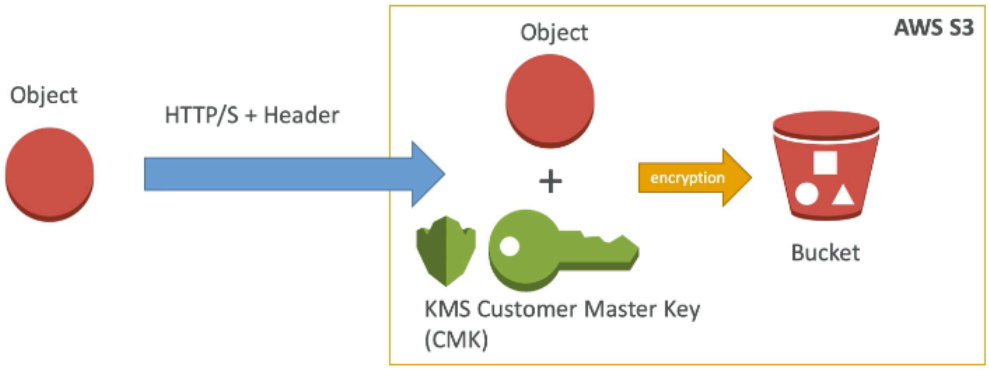
Key managed by S3:

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)

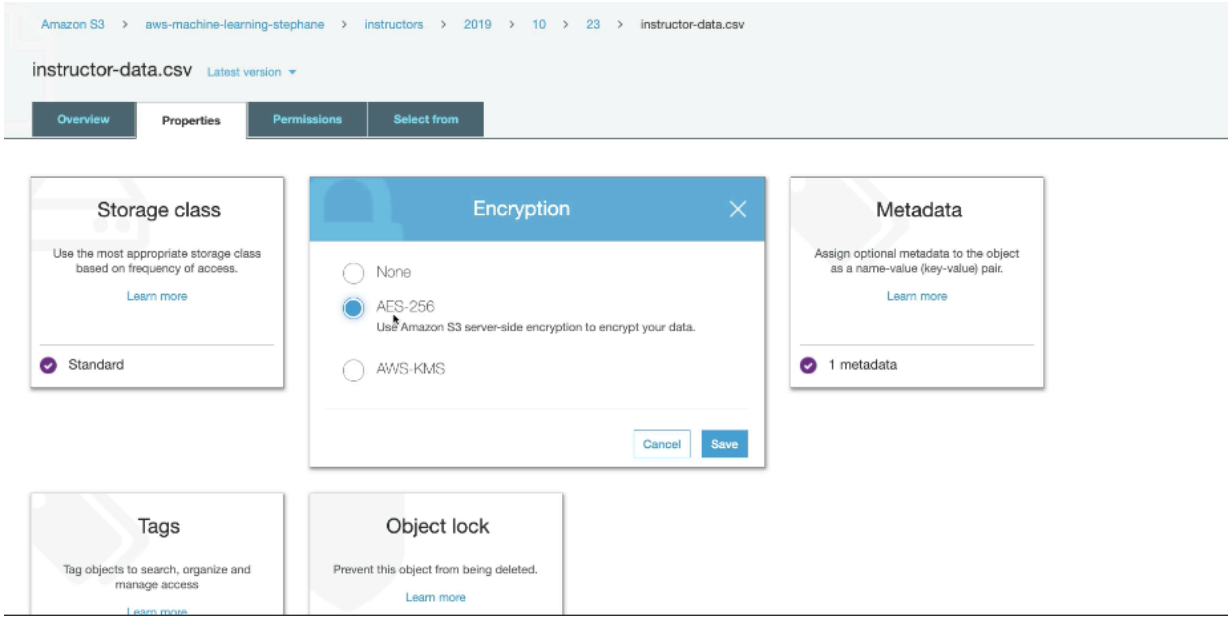
S3 Default Encryption vs Bucket Policies

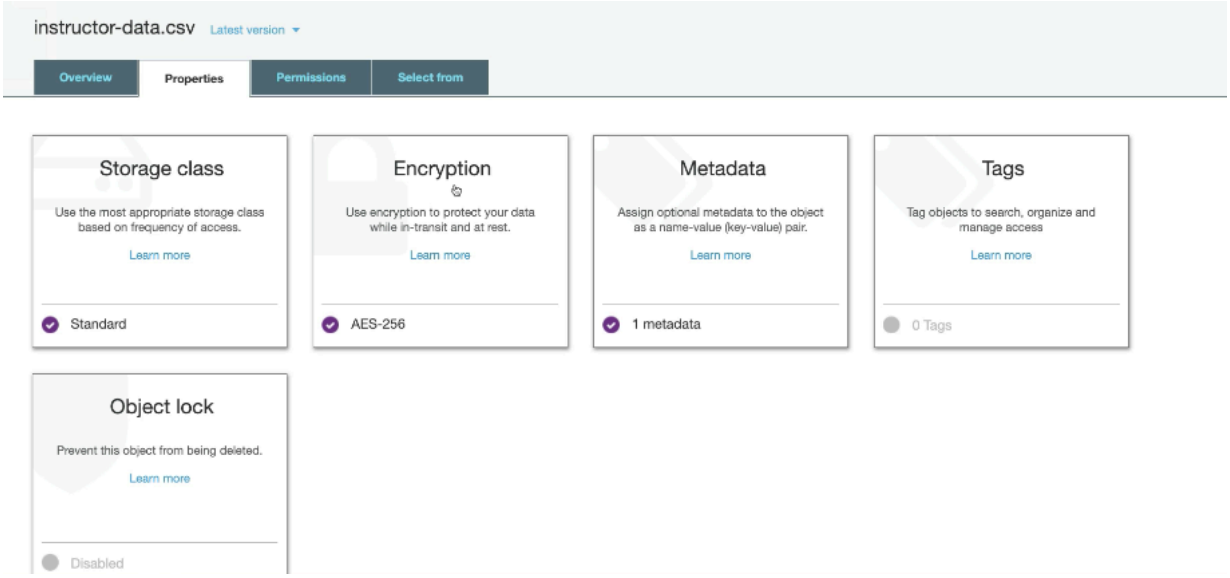
- The old way to enable default encryption was to use a bucket policy and refuse any HTTP command without the proper headers:

```
{
    "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"
                }
            }
        },
```

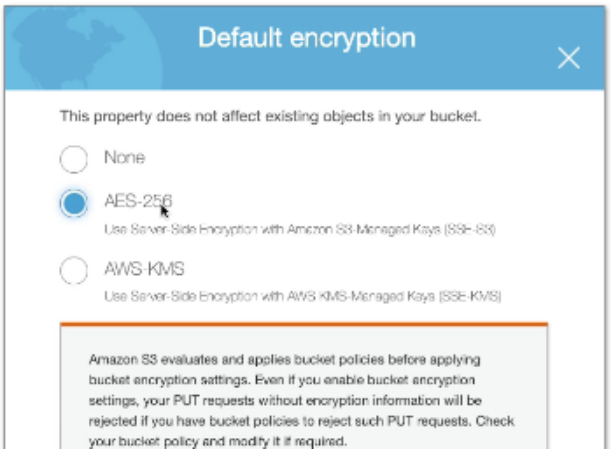
```
        {
            "Sid": "DenyUnEncryptedObjectUploads",
            "Effect": "Deny",
            "Principal": "*",
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::<bucket_name>/*",
            "Condition": {
                "Null": {
                    "s3:x-amz-server-side-encryption":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

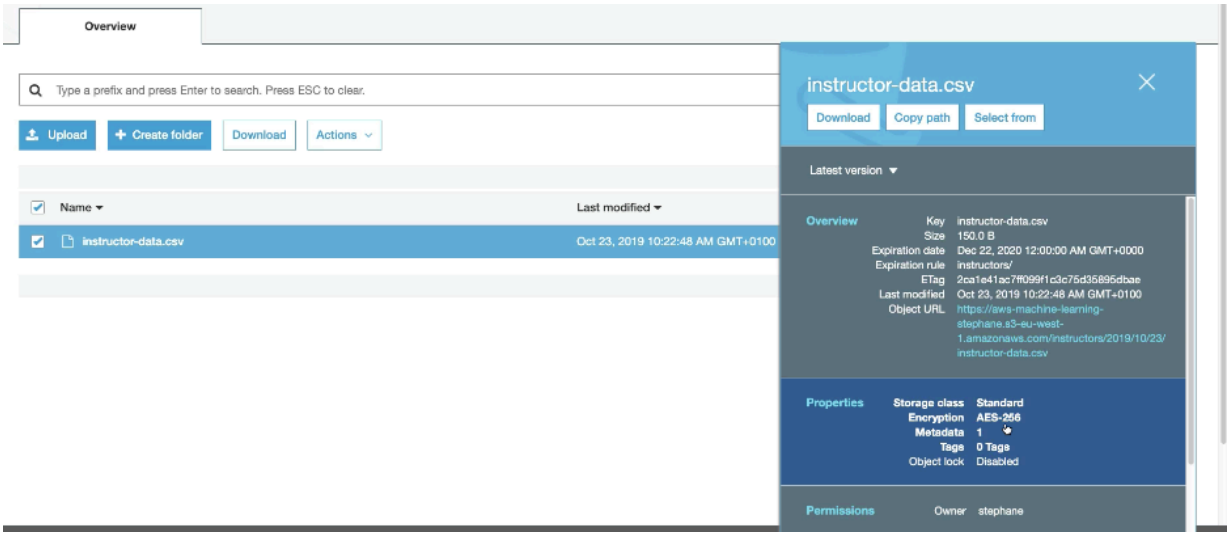
- The new way is to use the "default encryption" option in S3
- Note: Bucket Policies are evaluated before "default encryption"



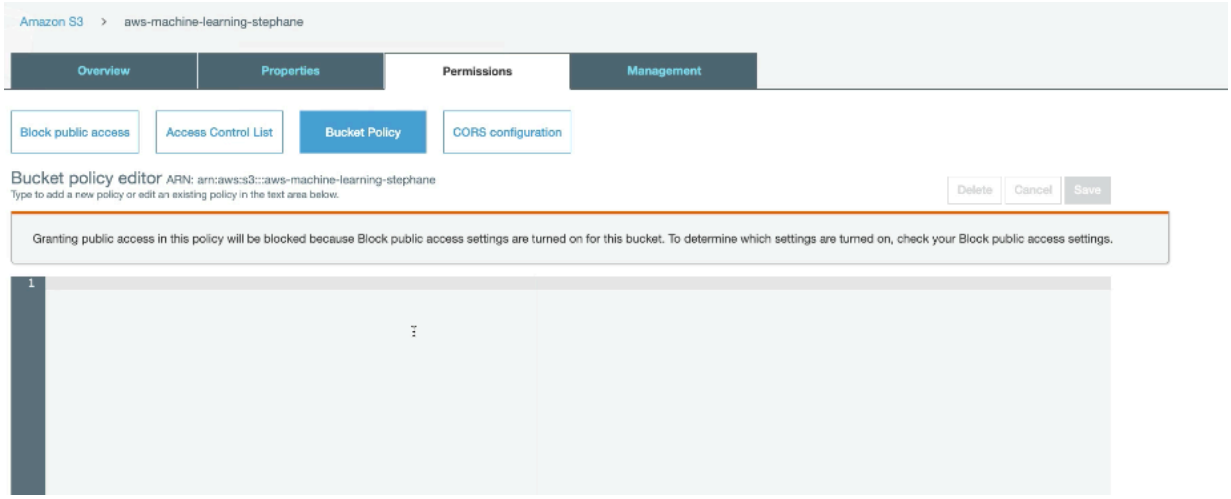


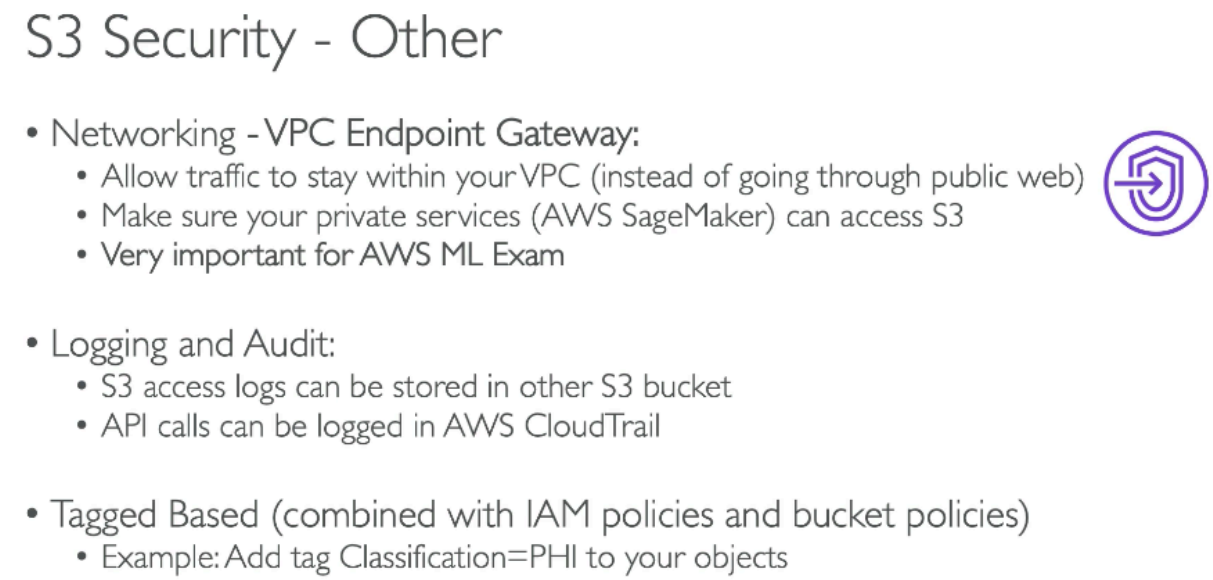
In properties, can set default encryption:





Another way would be to set bucket policy and force encryption





2. **AWS Kinesis**

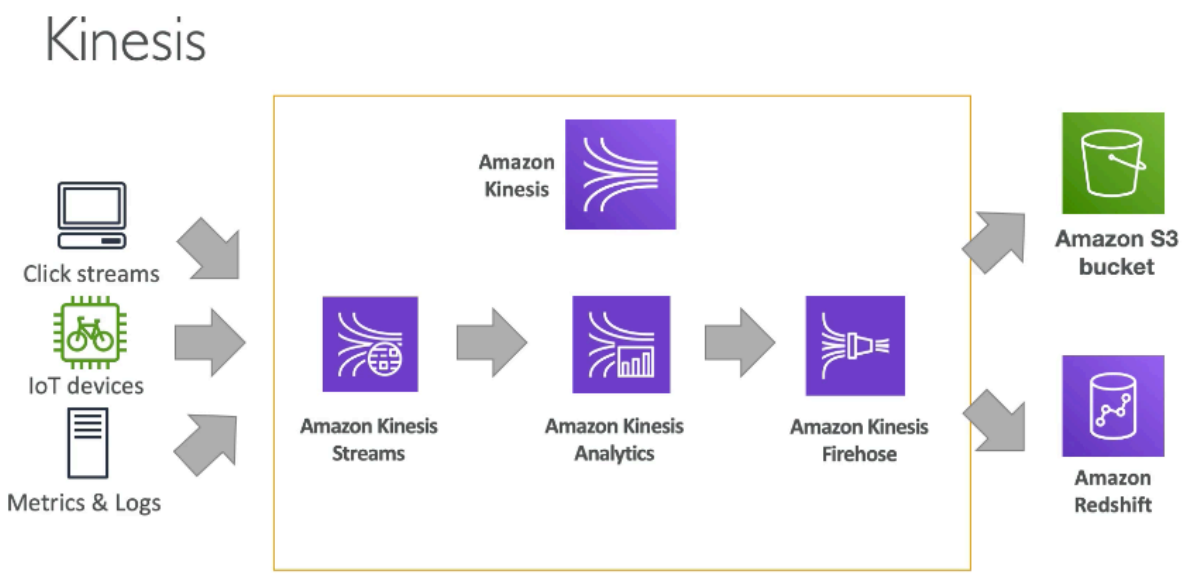
Exam will test what is the difference between the 4 Kinesis steams

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

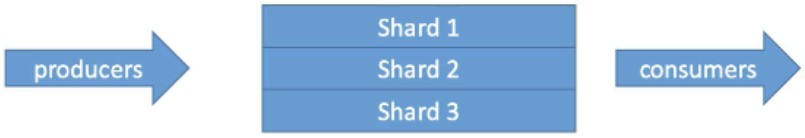
- Kinesis Streams: low latency streaming ingest at scale
- Kinesis Analytics: perform real-time analytics on streams using SQL
- Kinesis Firehose: load streams into S3, Redshift, ElasticSearch & Splunk
- Kinesis Video Streams: meant for streaming video in real-time

One simple architecture of how things work:



Kinesis 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

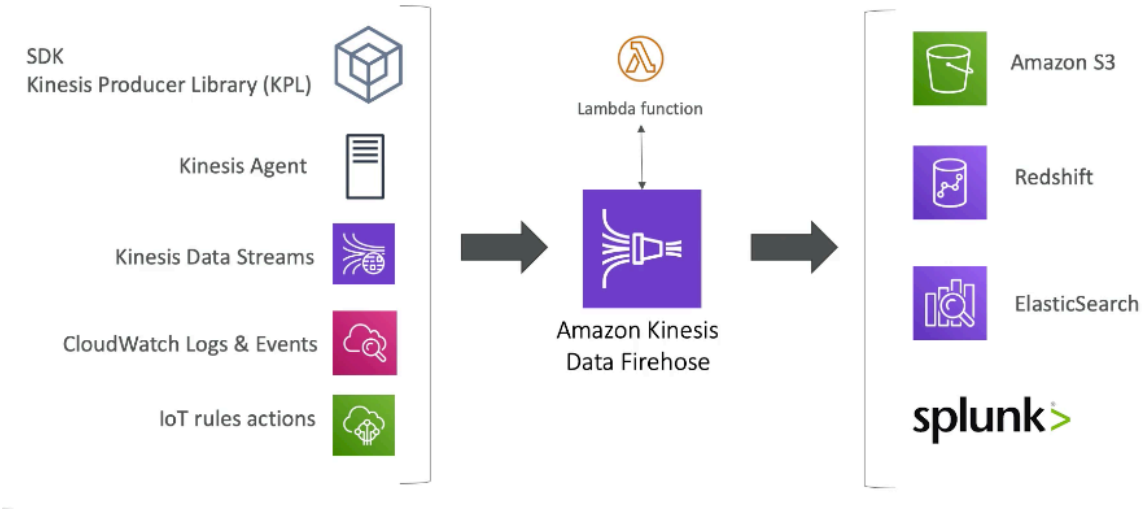
Kinesis Data Streams Limits to know

- 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

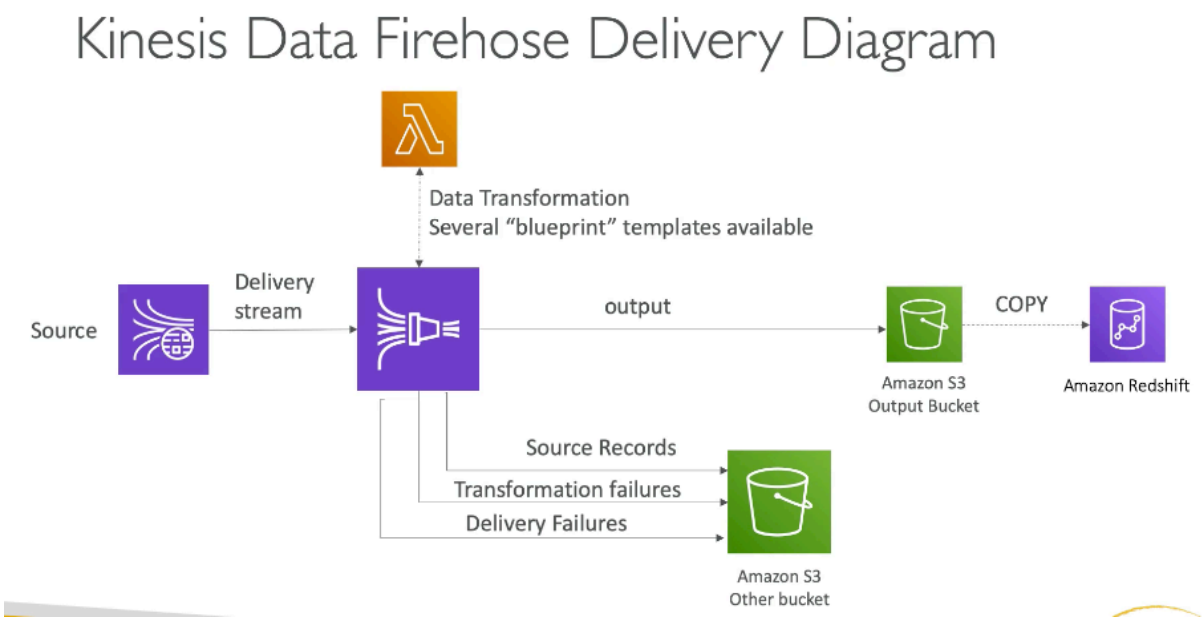
Kinesis Data Firehose

- 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



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.

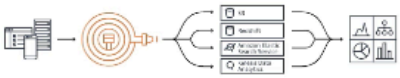
Ingest and process streaming data with Kinesis streams

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

Continously 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

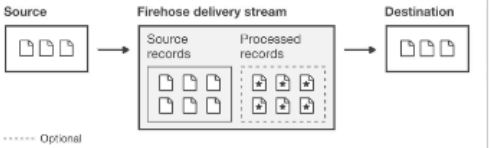
ticker_demo

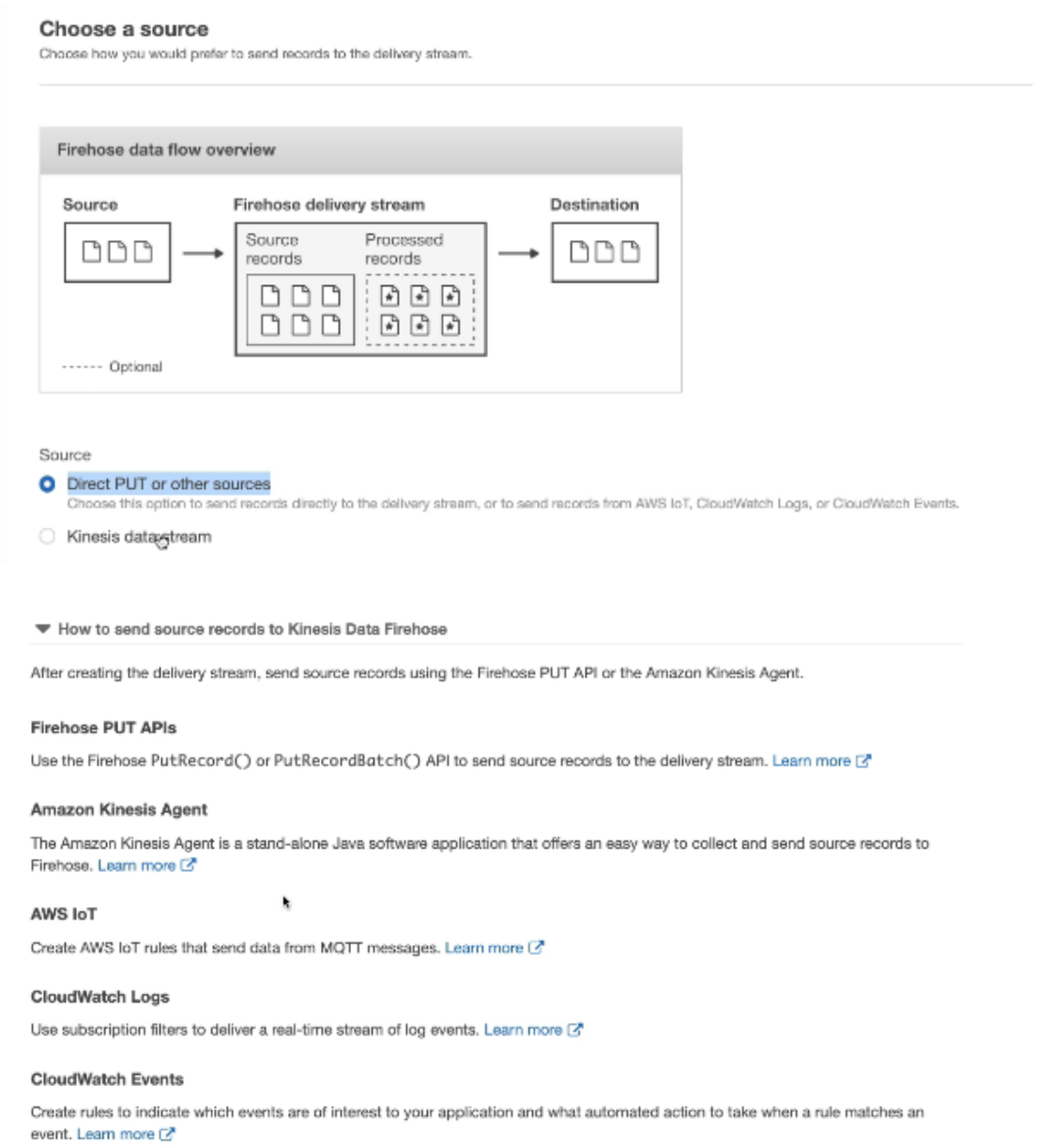
Acceptable characters are uppercase and lowercase letters, numbers, underscores, hyphens, and periods.

Choose a source

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

Firehose data flow overview





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.

Process records data flow overview

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

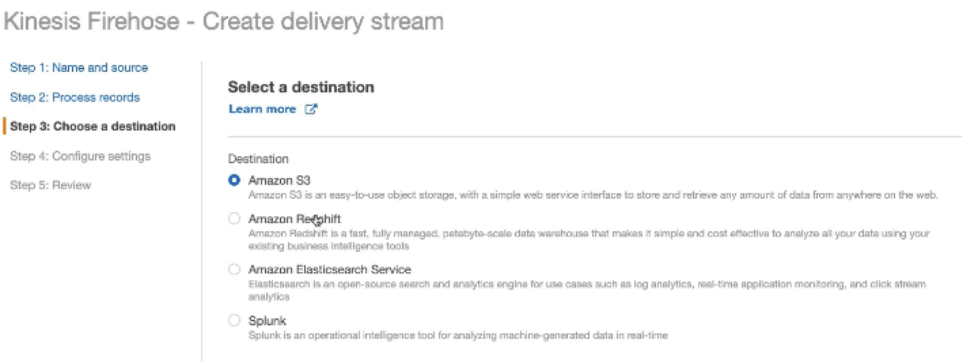
Specify a schema for source records. Kinesis Data Firehose references table definitions stored in AWS Glue. Choose an AWS Glue table to specify a schema for your source records. You can manually create a new table in AWS Glue, or add a crawler in AWS Glue to create a new table using a schema from an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

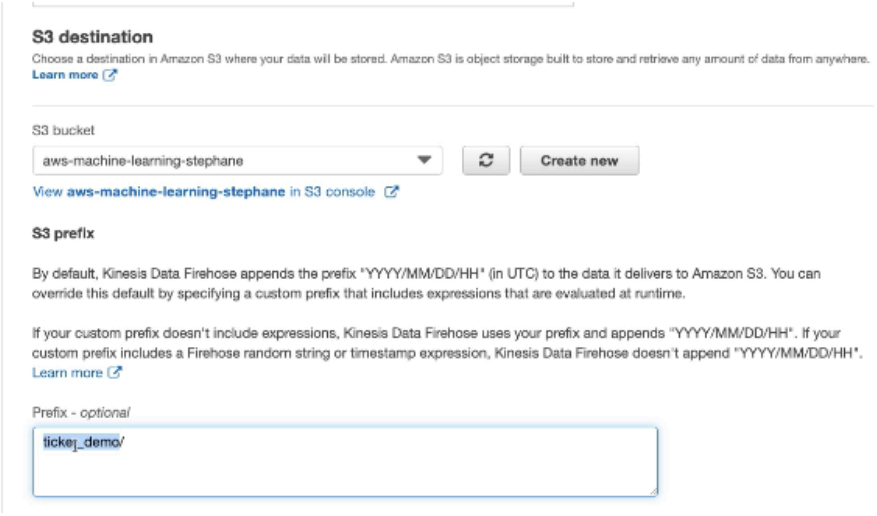
AWS Glue region

Choose a region

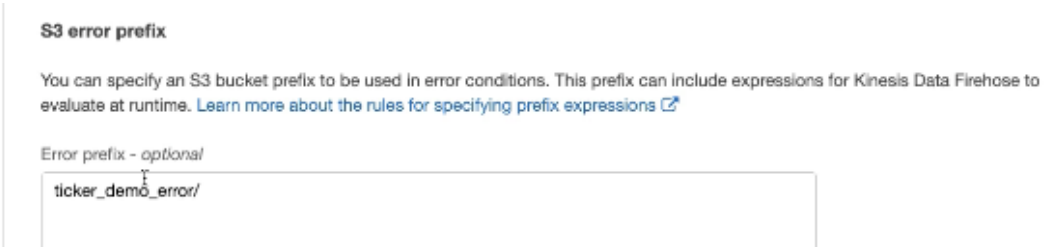
AWS Glue database

Choose a database

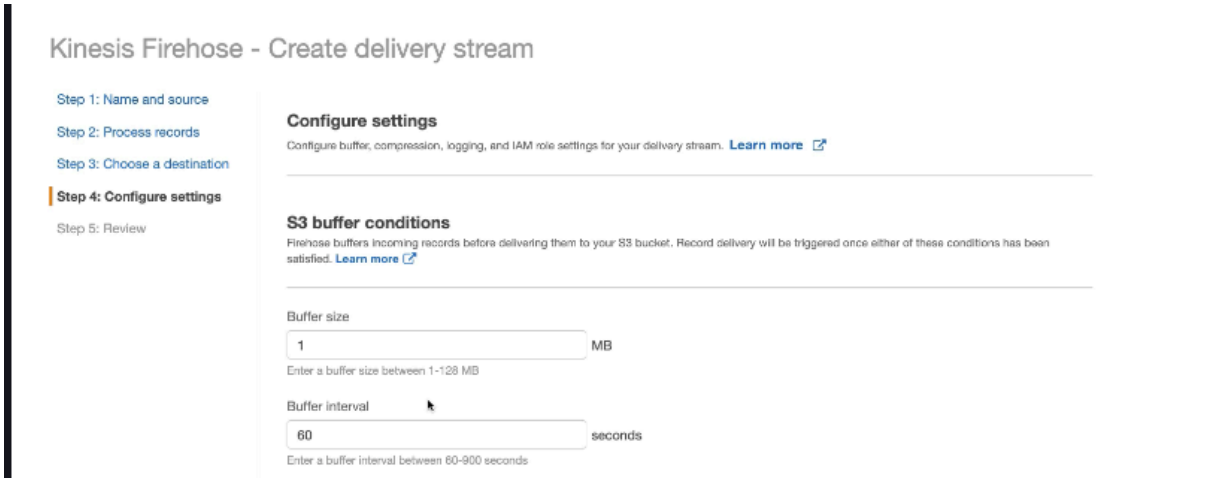




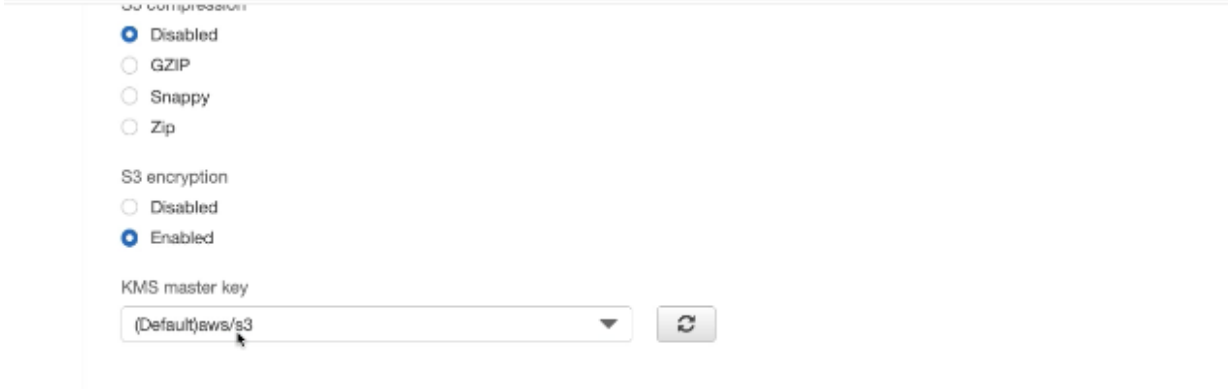
Don’t forget the “/“ in the prefix



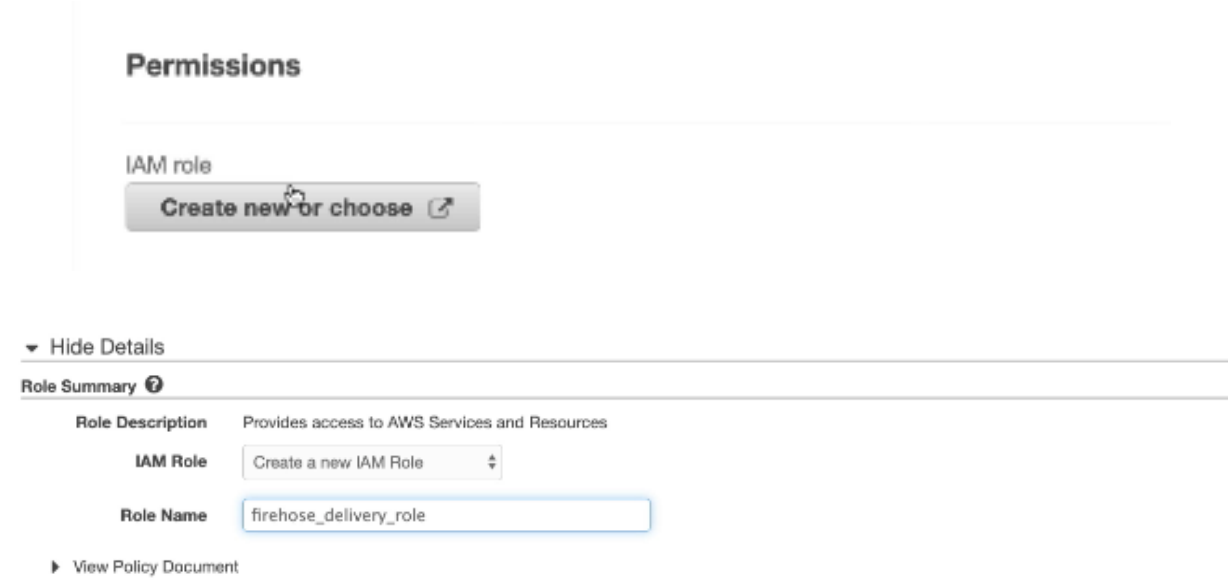
Buffer condition - deliver the data as soon as possible (near real time),.
Cannot have a buffer less than 60 seconds



Possibility to compress and encrypt if we want

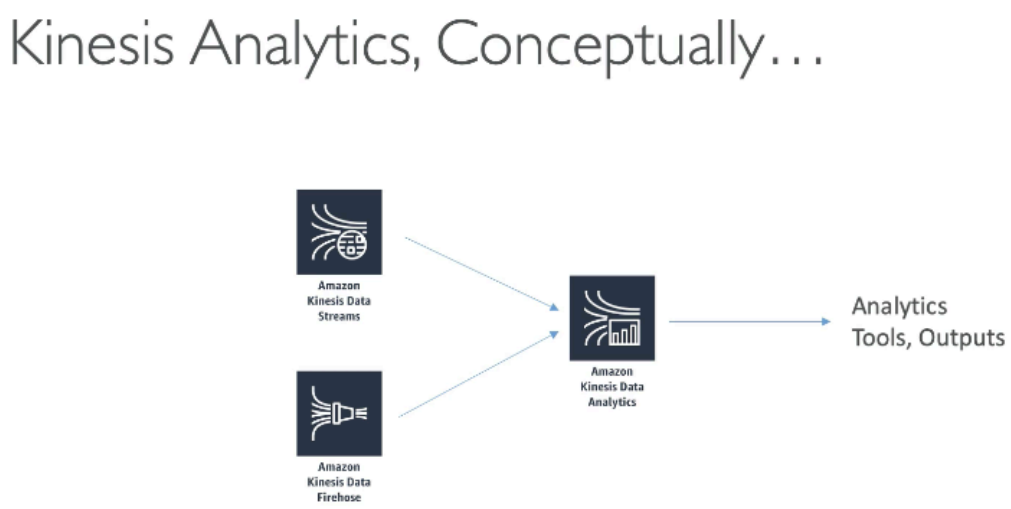


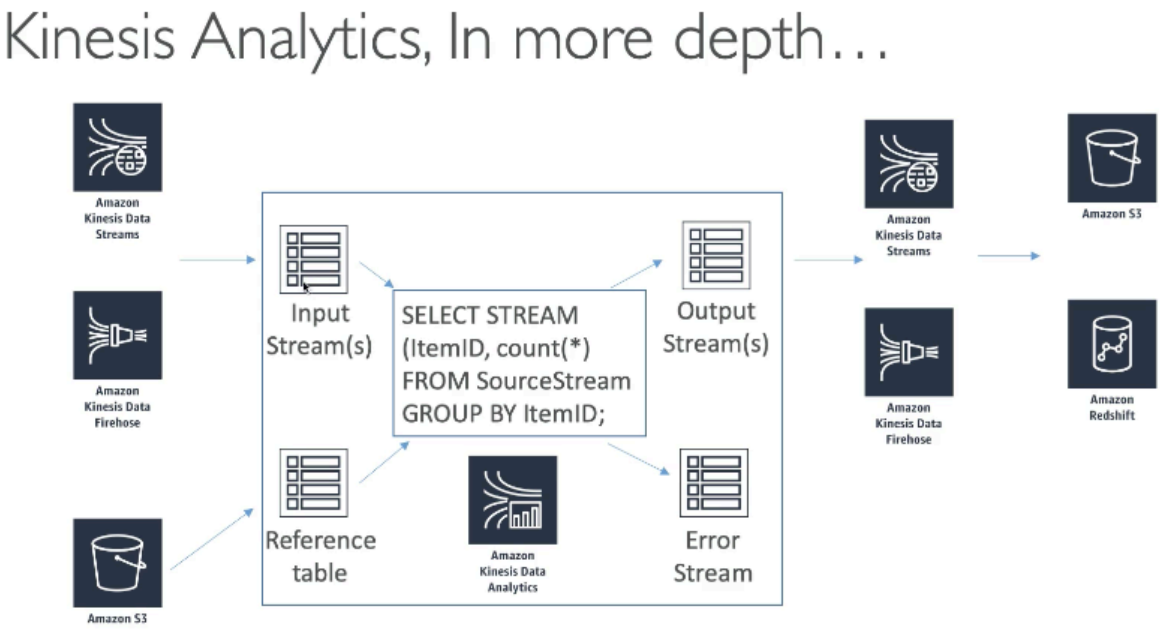
Create nwe IAM role which allow Firehose to send data into S3



Kinesis Data Analytics

Takes data from Firehose or streams





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

Kinesis Data Analytics

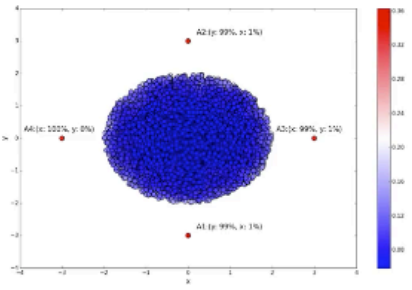
- Use cases
  - Streaming ETL: **select columns**, make simple transformations, on streaming data
  - Continuous metric generation: live leaderboard for a mobile game
  - Responsive analytics: look for certain criteria and build alerting (filtering)

- 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

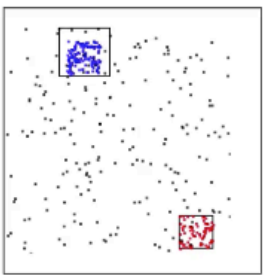
Anomalies and Dense areas in Data Analytics

Machine Learning on Kinesis Data Analytics

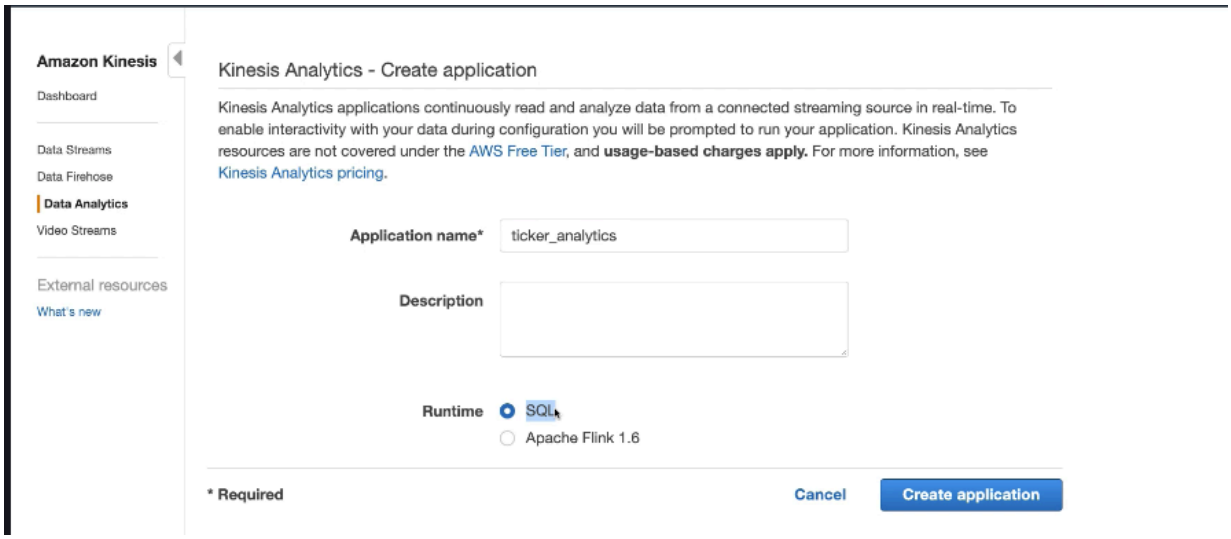
- RANDOM_CUT_FOREST
  - SQL function used for **anomaly detection** on numeric columns in a stream
  - Example: detect anomalous subway ridership during the NYC marathon
  - Uses **recent history** to compute model



- 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



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

Choose source | Configure a new stream

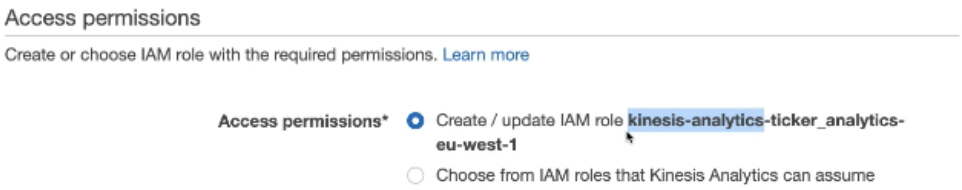
Source*

- 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* ticker_demo | Create new

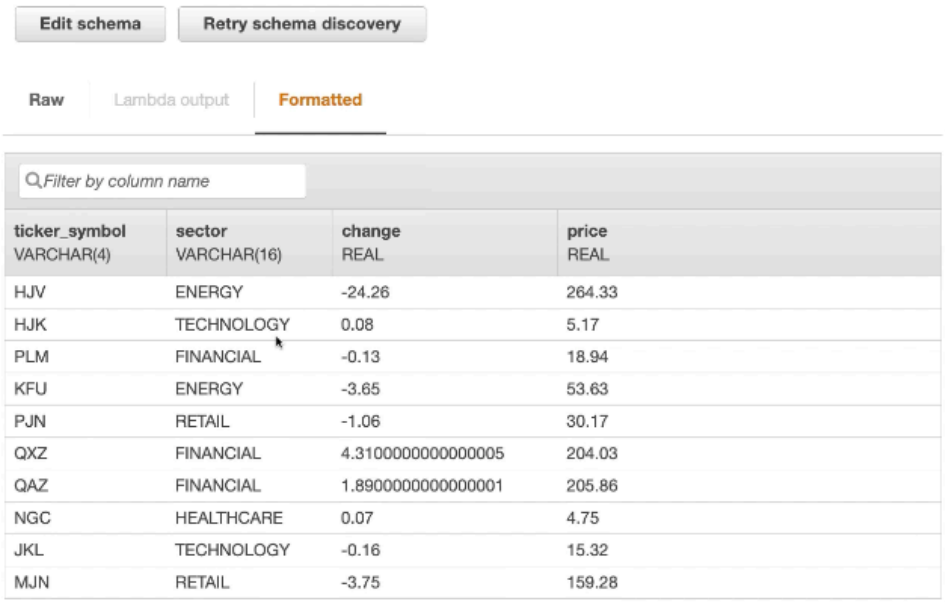
View **ticker_demo** in Kinesis Firehose

In-application stream name In your SQL queries, refer to this source as:

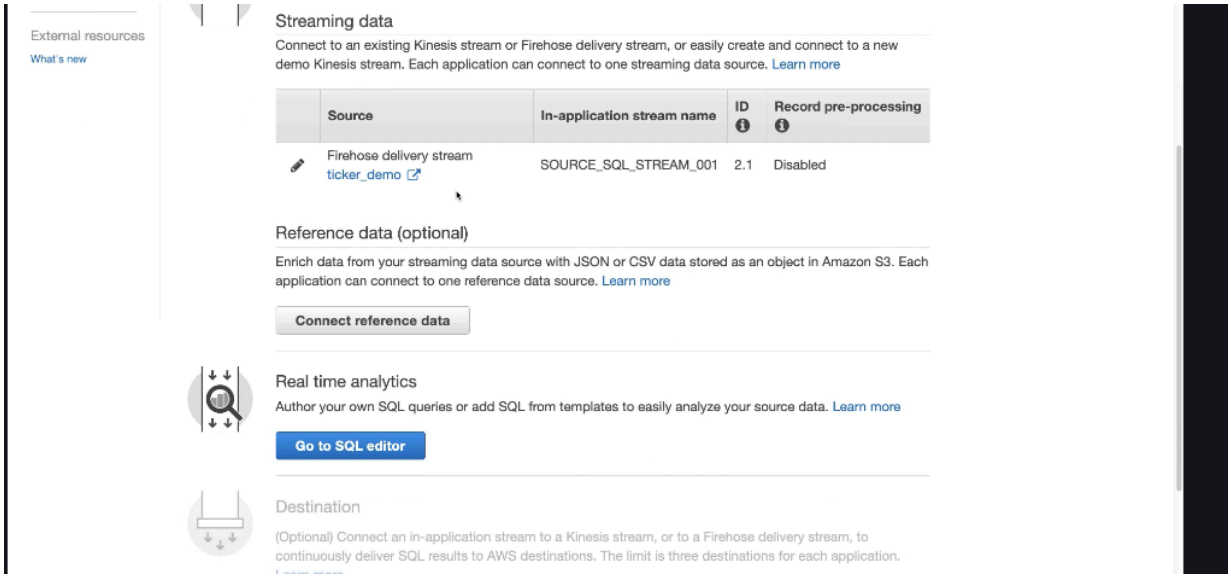


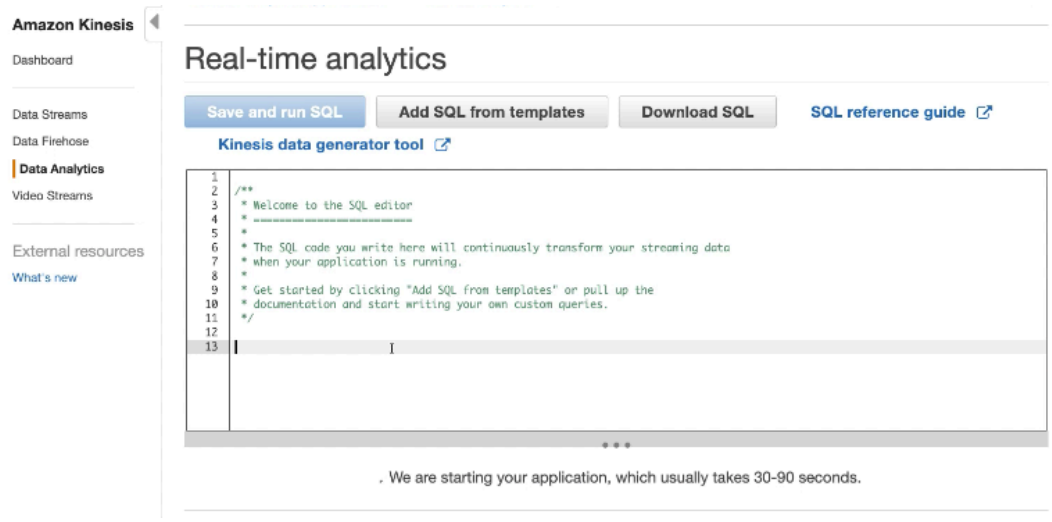
Role to allow Kinesis analytics to perform read on Kinesis Firehose delivery stream

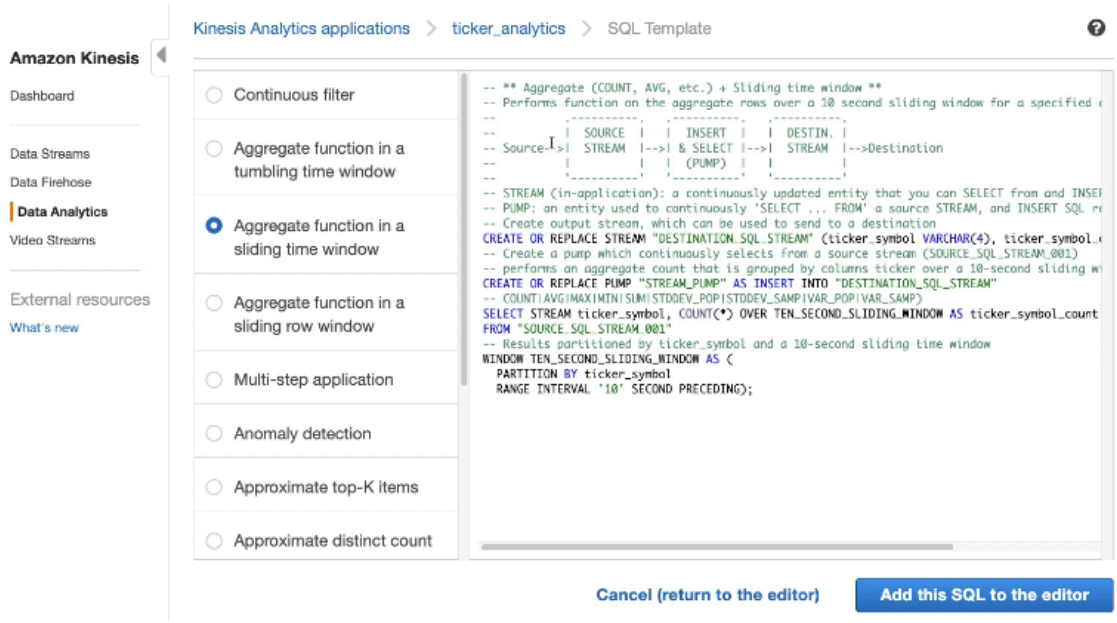
Schema can be automatically inferred

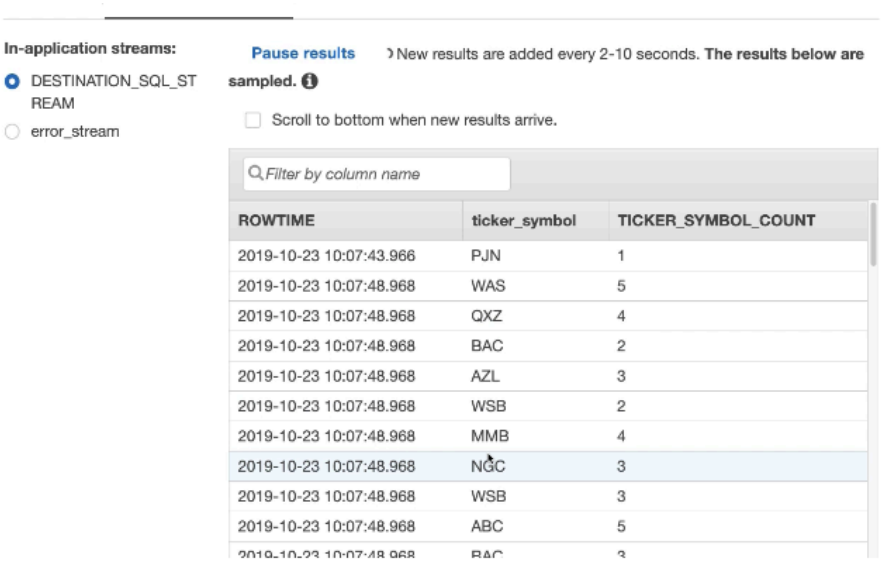


ticker_symbol VARCHAR(4)	sector VARCHAR(16)	change REAL	price REAL
HJV	ENERGY	-24.26	264.33
HJK	TECHNOLOGY	0.08	5.17
PLM	FINANCIAL	-0.13	18.94
KFU	ENERGY	-3.65	53.63
PJN	RETAIL	-1.06	30.17
QXZ	FINANCIAL	4.3100000000000005	204.03
QAZ	FINANCIAL	1.8900000000000001	205.86
NGC	HEALTHCARE	0.07	4.75
JKL	TECHNOLOGY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

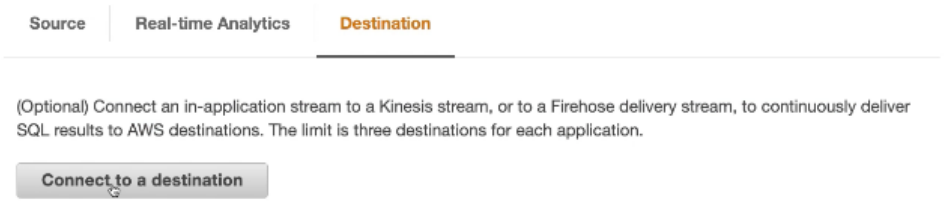


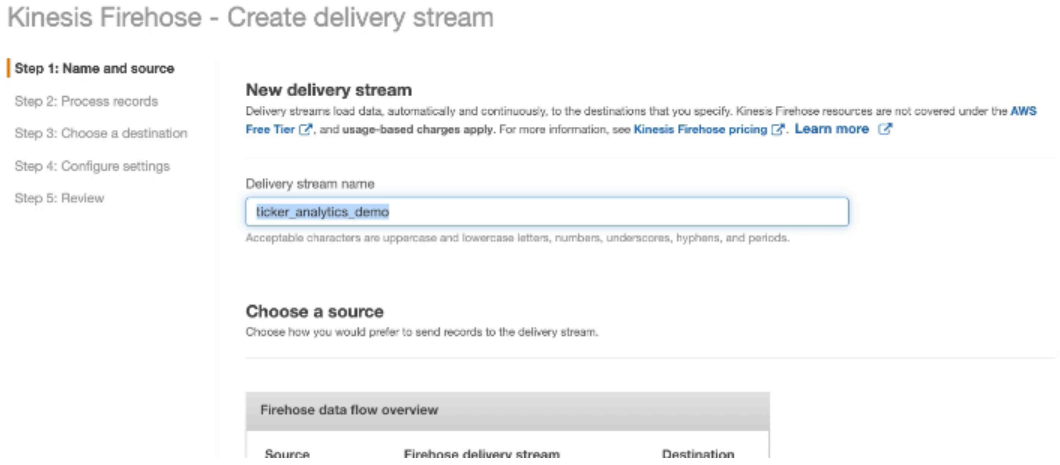

Amazon Kinesis
Dashboard
Data Streams
Data Firehose
Data Analytics
Video Streams
External resources
What's new
Real-time analytics
Save and run SQL
Add SQL from templates
Download SQL
SQL reference guide
Kinesis data generator tool
1 /**
2 * Welcome to the SQL editor
3 * =====================
4 *
5 * 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 */
11
12
13
. We are starting your application, which usually takes 30-90 seconds.

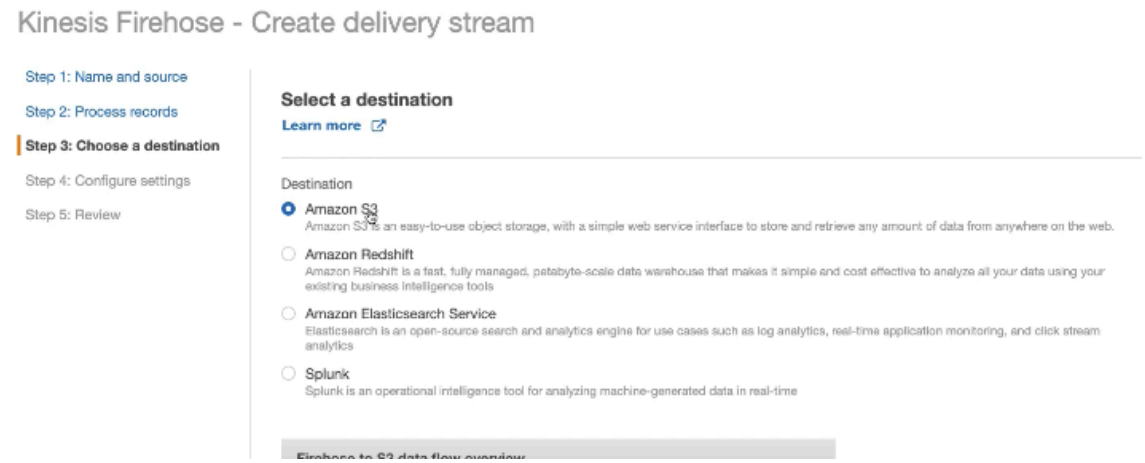

Amazon Kinesis
Dashboard
Data Streams
Data Firehose
Data Analytics
Video Streams
External resources
What's new
Kinesis Analytics applications
ticker_analytics
SQL Template
Continuous filter
Aggregate function in a tumbling time window
Aggregate function in a sliding time window
Aggregate function in 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
-- Source->| SOURCE_STREAM |-->| & SELECT |-->| STREAM |-->Destination
-- STREAM (in-application): a continuously updated entity that you can SELECT from and INSE
-- PUMP: an entity used to continuously 'SELECT ... FROM' a source STREAM, and INSERT SQL r
-- 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_
-- 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 w
CREATE OR REPLACE PUMP "STREAM_PUMP" AS INSERT INTO "DESTINATION_SQL_STREAM"
-- COUNT|AVG|MAX|MIN|SUM|STDDEV_POP|STDDEV_SAMP|VAR_POP|VAR_SAMP)
SELECT STREAM ticker_symbol, COUNT(*) OVER TEN_SECOND_SLIDING_WINDOW AS ticker_symbol_count
FROM "SOURCE_SQL_STREAM_001"
-- Results partitioned by ticker_symbol and a 10-second sliding time window
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

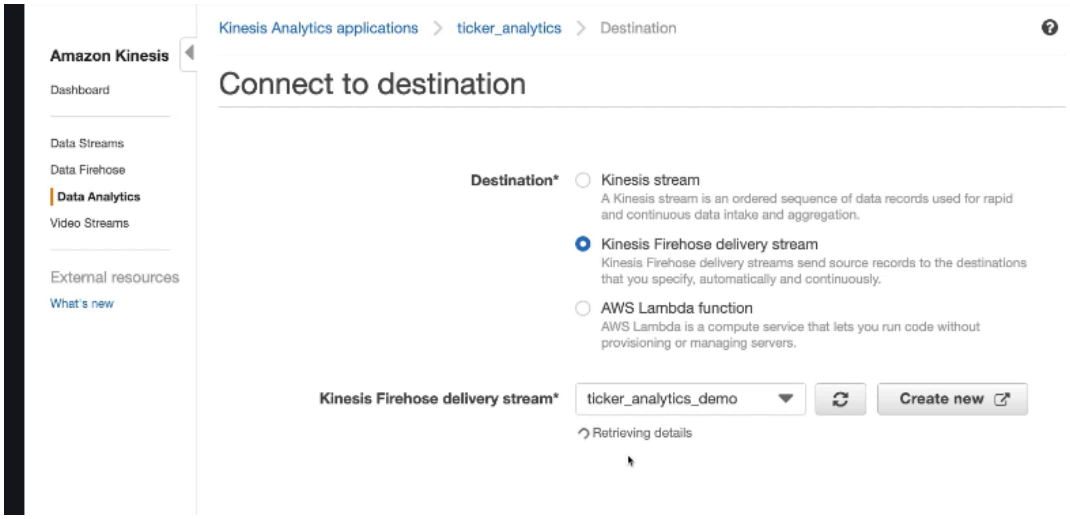


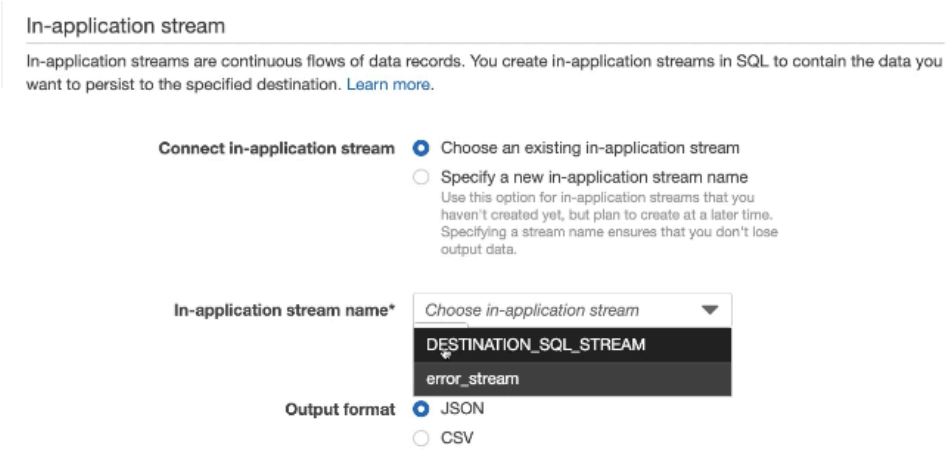
Now connect a destination to send the resulting aggregation via Firehose

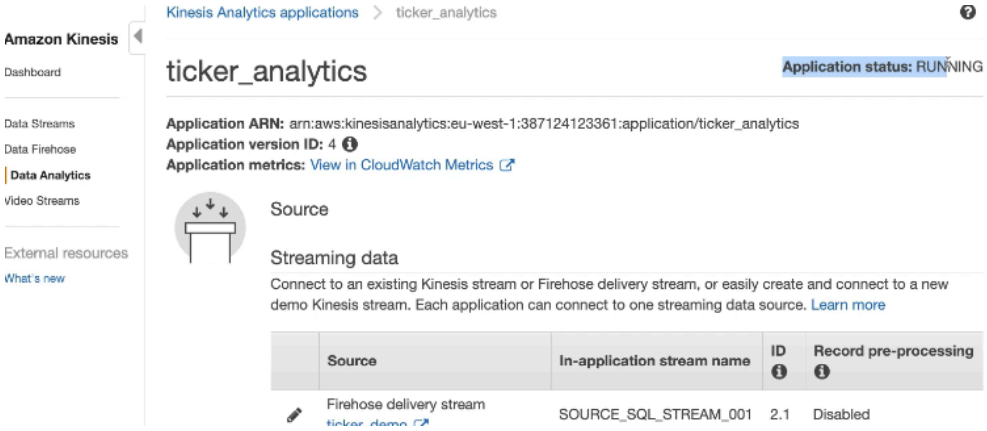


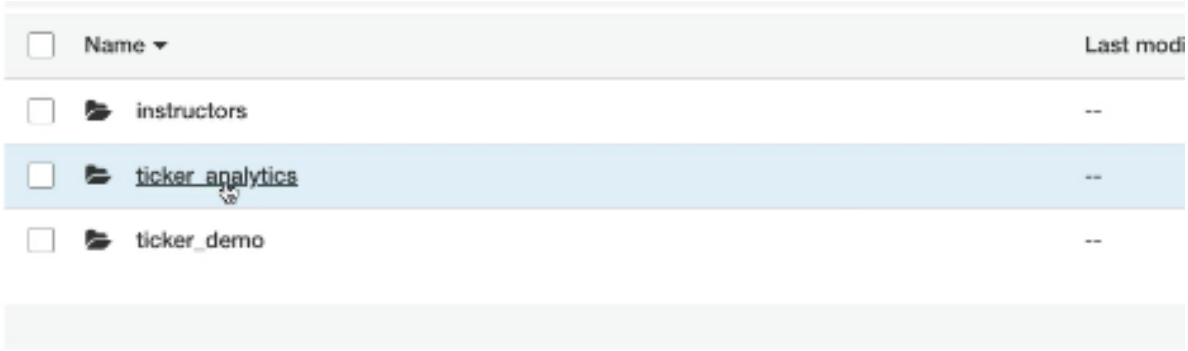


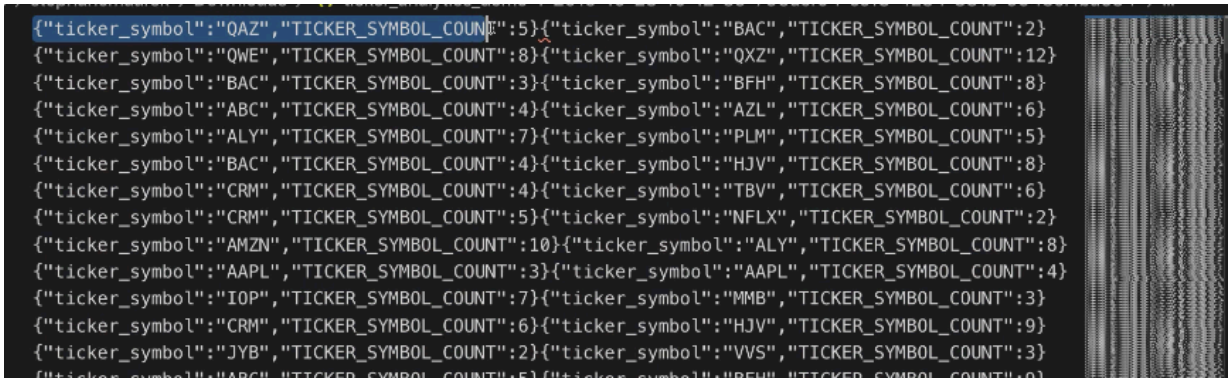

Kinesis Firehose - Create delivery stream
Step 1: Name and source
Step 2: Process records
Step 3: Choose a destination
Step 4: Configure settings
Step 5: Review
Select a destination
Learn more
Destination
Amazon S3
Amazon S3 is an easy-to-use object storage, with a simple web service interface to store 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


Amazon Kinesis
Dashboard
Data Streams
Data Firehose
Data Analytics
Video Streams
External resources
What's new
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*
ticker_analytics_demo
Create new
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*
Choose in-application stream
DESTINATION_SQL_STREAM
error_stream
Output format
JSON
CSV



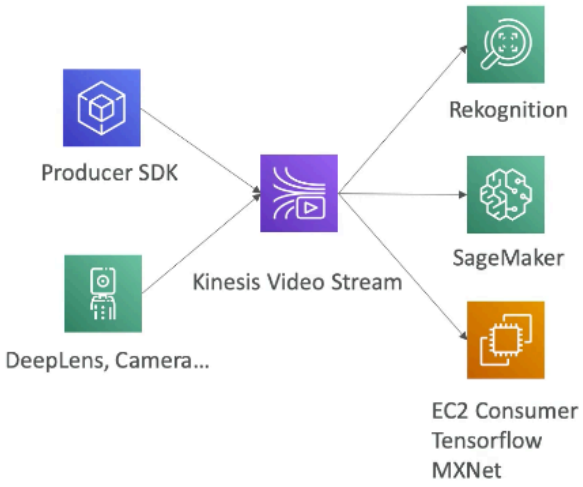




Kinesis Video Stream

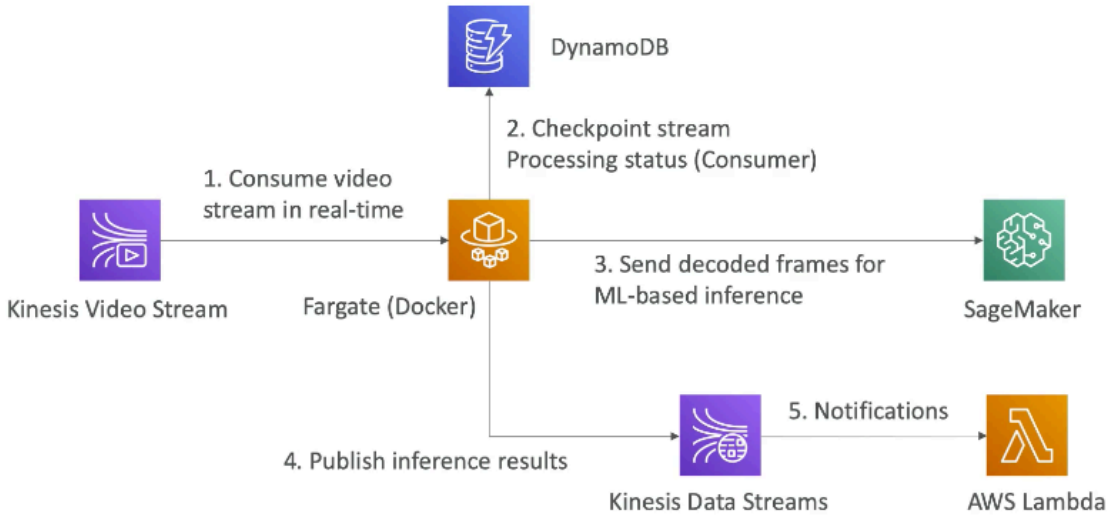
Kinesis Video Stream

- 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/



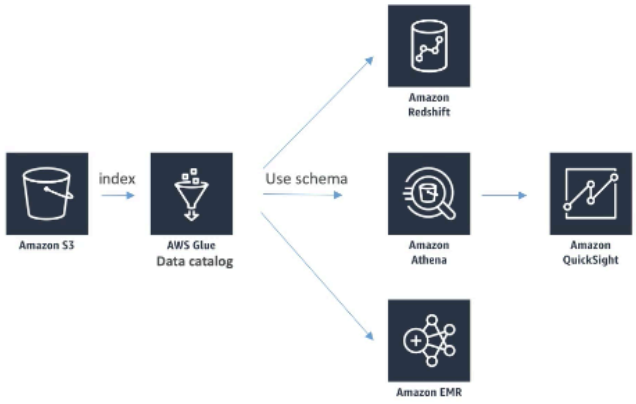
Kinesis Summary – Machine Learning

- 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



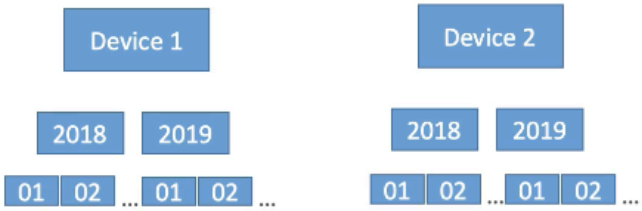
Glue Data Catalog - Crawlers

- 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

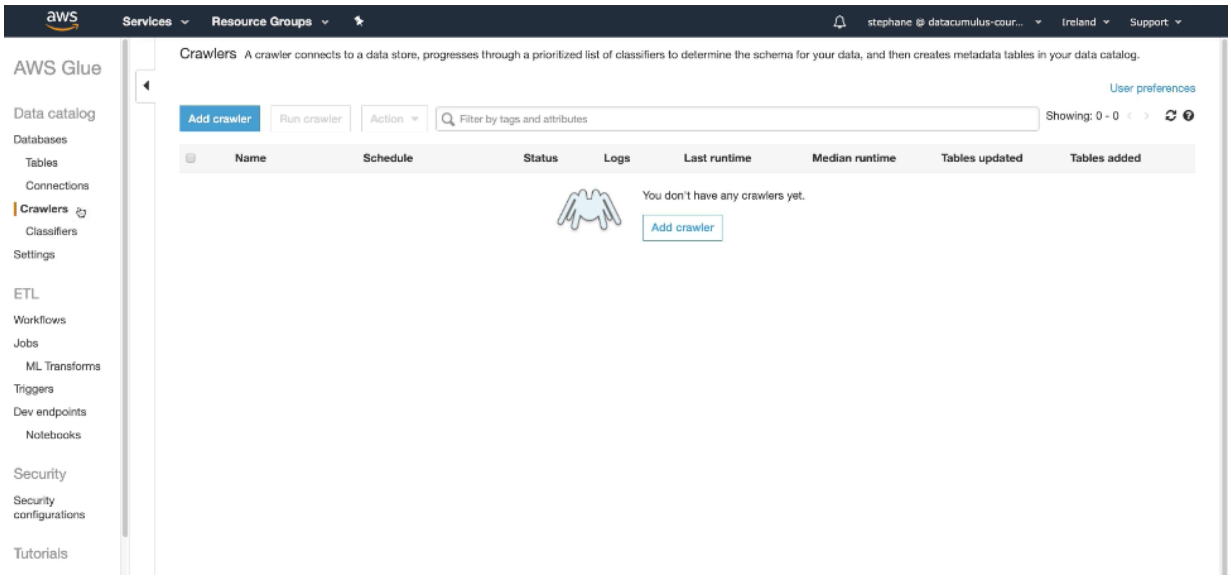
- Need an IAM role / credentials to access the data stores

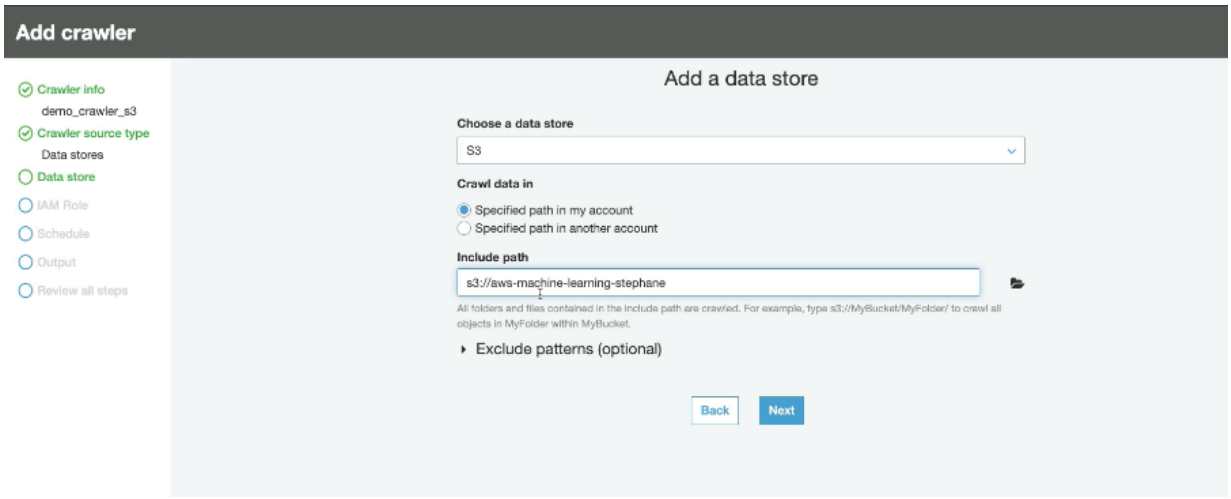
Glue and S3 Partitions

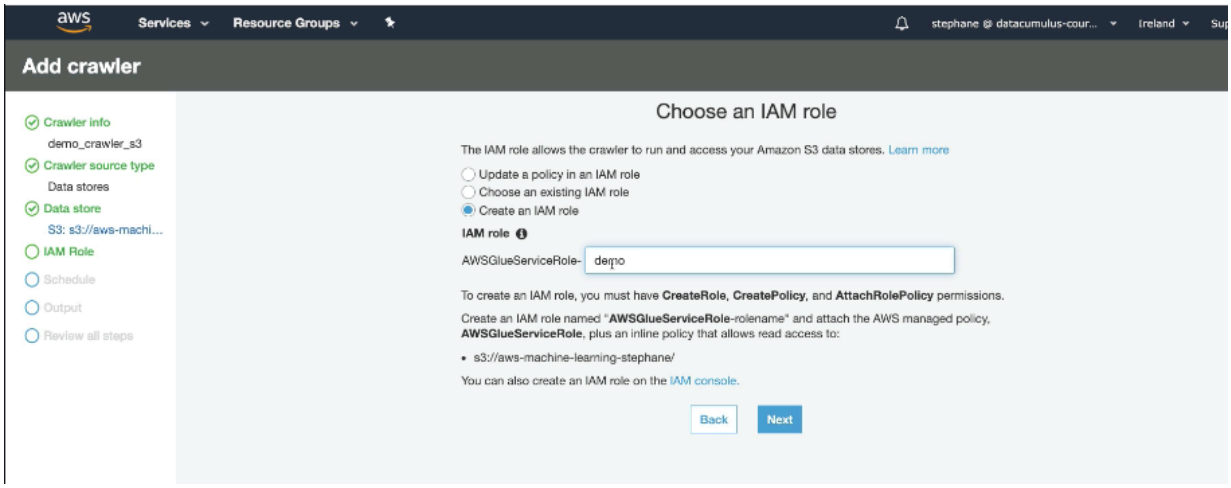
- Glue crawler will extract partitions based on how your S3 data is organized
- Think up front about how you will be querying your data lake in S3
- Example: devices send sensor data every hour
- Do you query primarily by time ranges?
  - If so, organize your buckets as s3://my-bucket/dataset/yyyy/mm/dd/device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

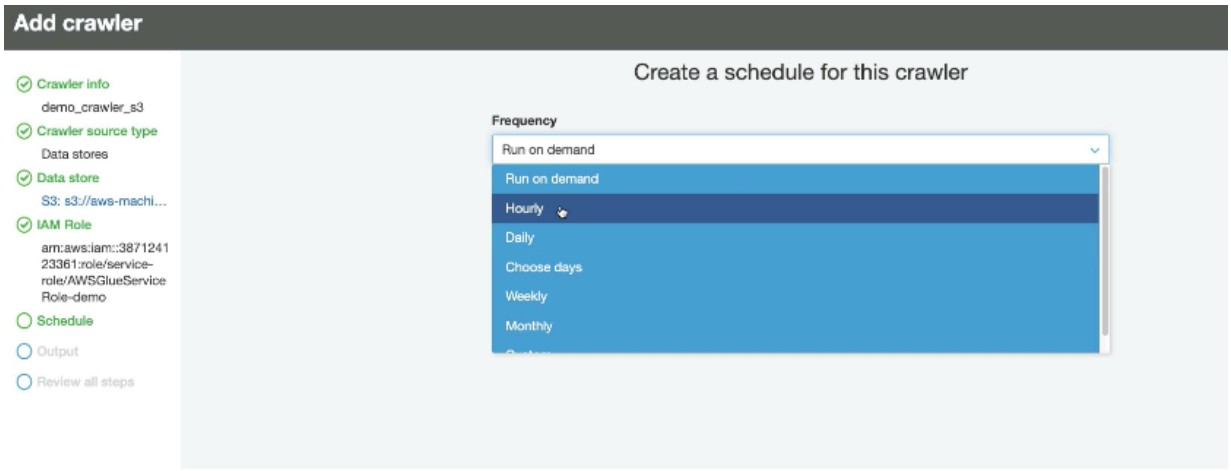


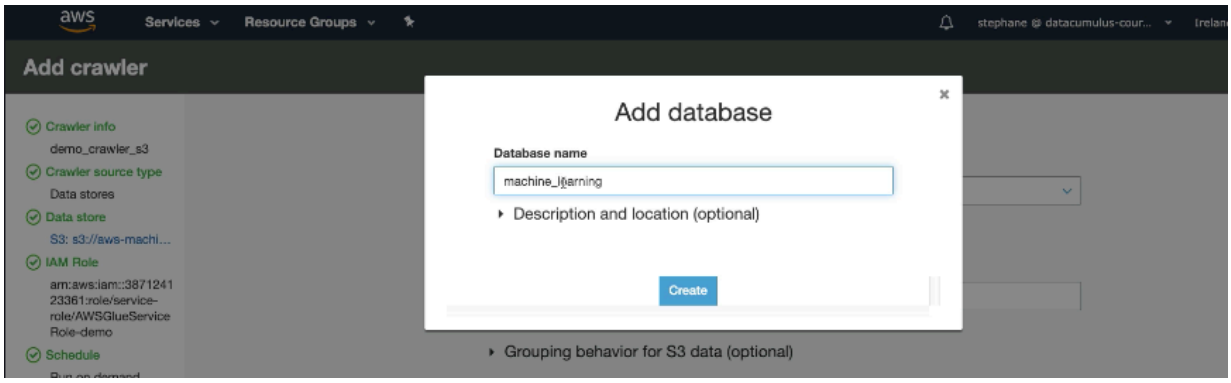
Glue Lab

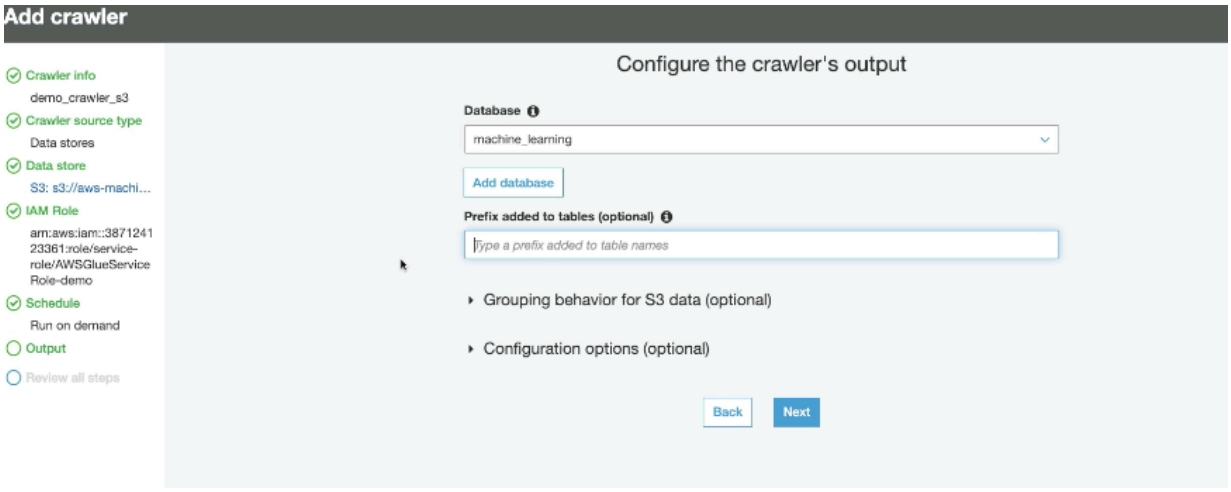

aws
Services
Resource Groups
stephane @ datacumulus-cour...
Ireland
Support
AWS Glue
Data catalog
Databases
Tables
Connections
Crawlers
Classifiers
Settings
ETL
Workflows
Jobs
ML Transforms
Triggers
Dev endpoints
Notebooks
Security
Security configurations
Tutorials
Crawlers A crawler connects to a data store, progresses through a prioritized list of classifiers to determine the schema for your data, and then creates metadata tables in your data catalog.
User preferences
Add crawler
Run crawler
Action
Filter by tags and attributes
Showing: 0 - 0
Name
Schedule
Status
Logs
Last runtime
Median runtime
Tables updated
Tables added
You don't have any crawlers yet.
Add crawler

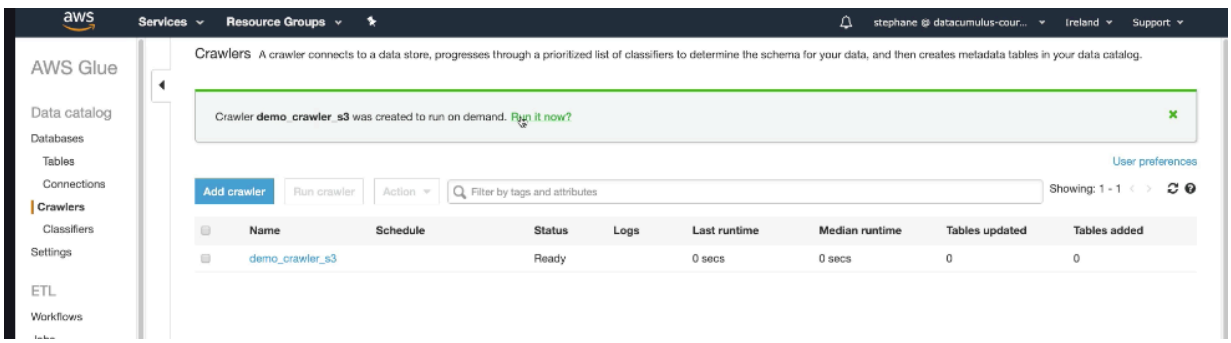

Add crawler
Crawler info
demo_crawler_s3
Crawler source type
Data stores
Data store
IAM Role
Schedule
Output
Review all steps
Add a data store
Choose a data store
S3
Crawl data in
Specified path in my account
Specified path in another account
Include path
s3://aws-machine-learning-stephane
All folders 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


aws
Services
Resource Groups
stephane @ datacumulus-cour...
Ireland
Add crawler
Crawler info
demo_crawler_s3
Crawler source type
Data stores
Data store
S3: s3://aws-machi...
IAM Role
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


Add crawler
Crawler info
demo_crawler_s3
Crawler source type
Data stores
Data store
S3: s3://aws-machi...
IAM Role
arn:aws:iam::387124123361:role/service-role/AWSGlueServiceRole-demo
Schedule
Output
Review all steps
Create a schedule for this crawler
Frequency
Run on demand
Run on demand
Hourly
Daily
Choose days
Weekly
Monthly

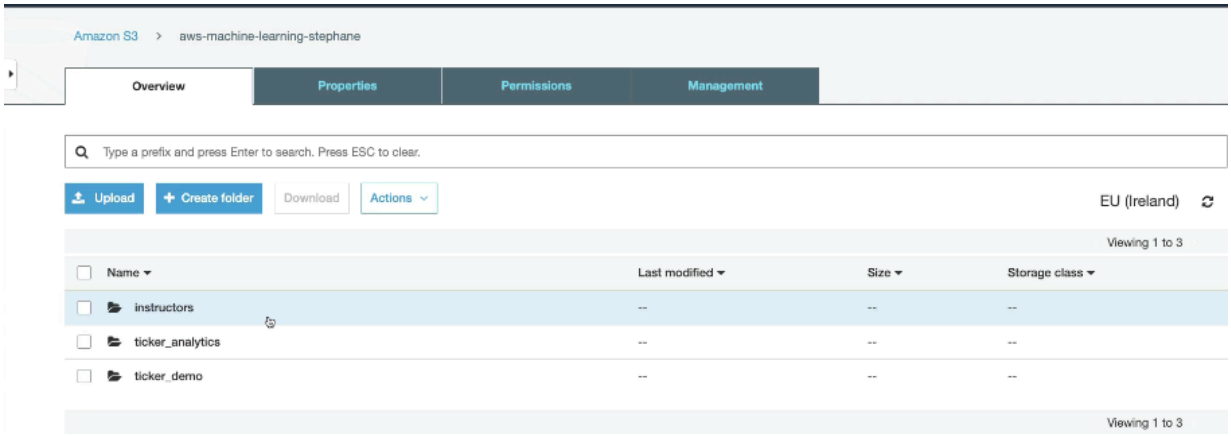

aws
Services
Resource Groups
stephane @ datacumulus-cour...
Add crawler
Crawler info
demo_crawler_s3
Crawler source type
Data stores
Data store
S3: s3://aws-machi...
IAM Role
arn:aws:iam::387124123361:role/service-role/AWSGlueServiceRole-demo
Schedule
Add database
Database name
machine_learning
Description and location (optional)
Create
Grouping behavior for S3 data (optional)


Add crawler
Crawler info
demo_crawler_s3
Crawler source type
Data stores
Data store
S3: s3://aws-machi...
IAM Role
arn:aws:iam::387124123361:role/service-role/AWSGlueServiceRole-demo
Schedule
Run on demand
Output
Review all steps
Configure the crawler's output
Database
machine_learning
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

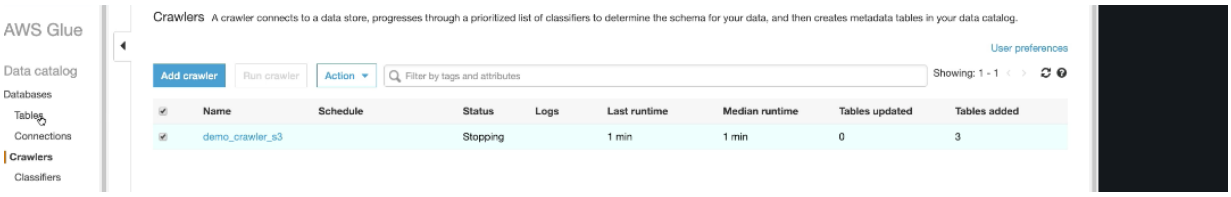

aws
Services
Resource Groups
stephane @ datacumulus-cour...
Ireland
Support
AWS Glue
Data catalog
Databases
Tables
Connections
Crawlers
Classifiers
Settings
ETL
Workflows
Crawlers A crawler connects to a data store, progresses through a prioritized list of classifiers to determine the schema for your data, and then creates metadata tables in your data catalog.
Crawler demo_crawler_s3 was created to run on demand. Run it now?
User preferences
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
Ready
0 secs
0 secs
0
0

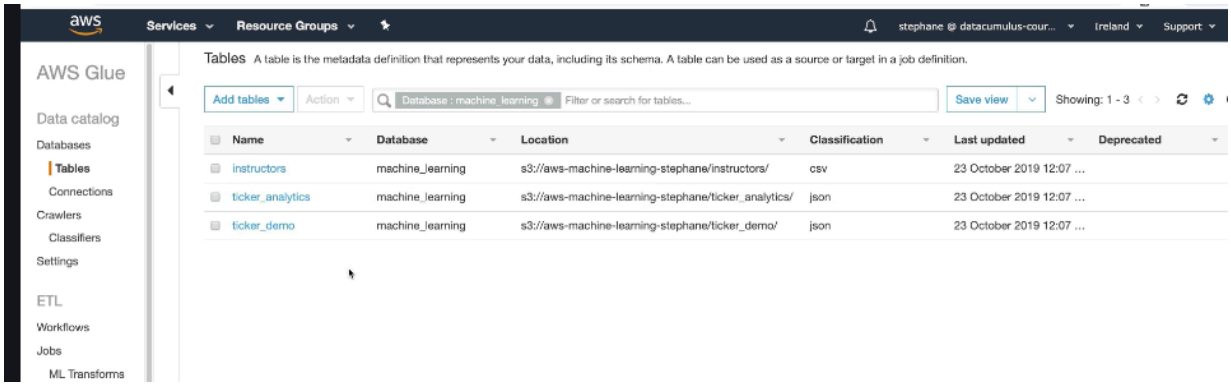
/

The crawler now runs against this bucket

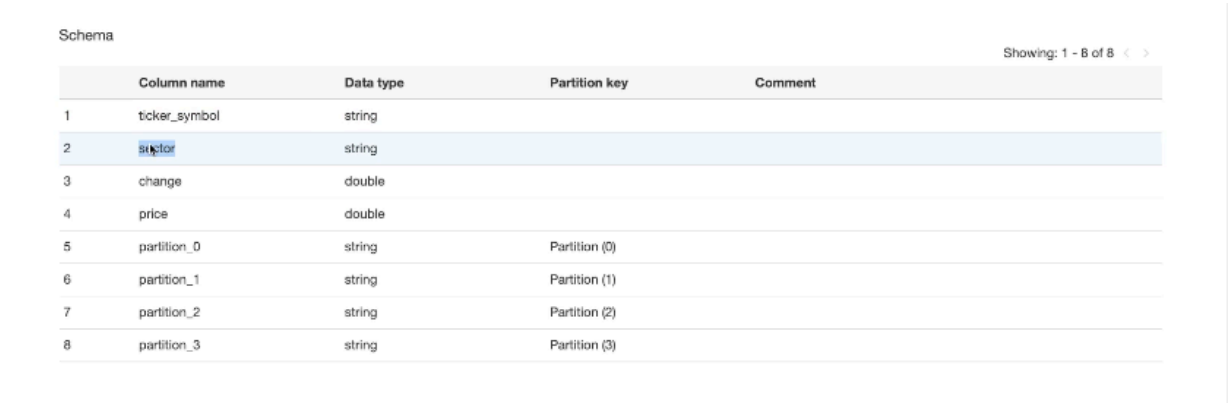


3 tables were added

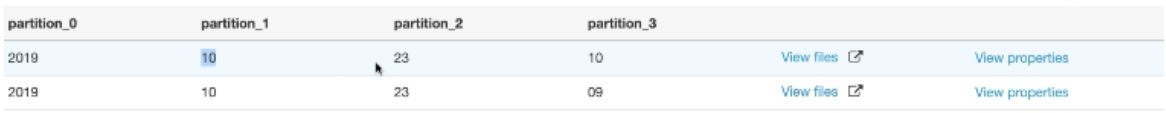




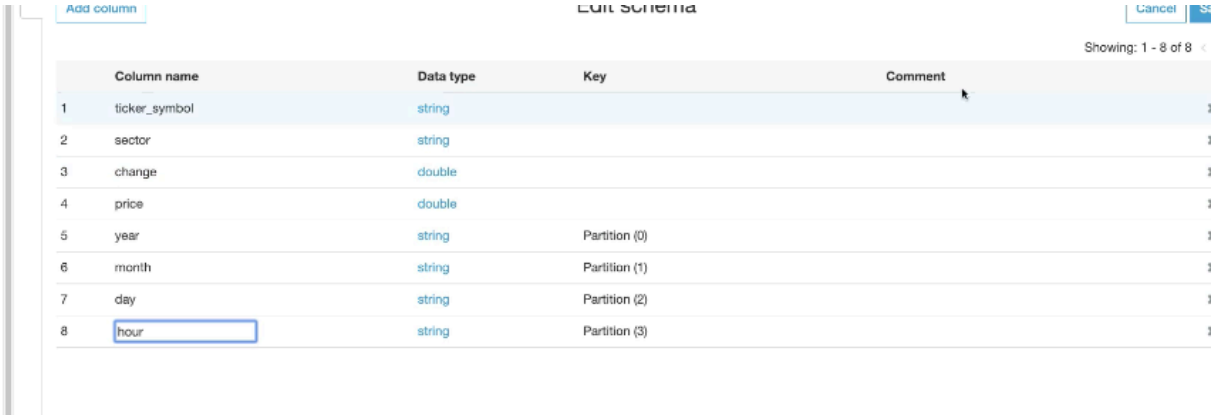
Ex: ticker_demo table



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



To note we can edit the schema



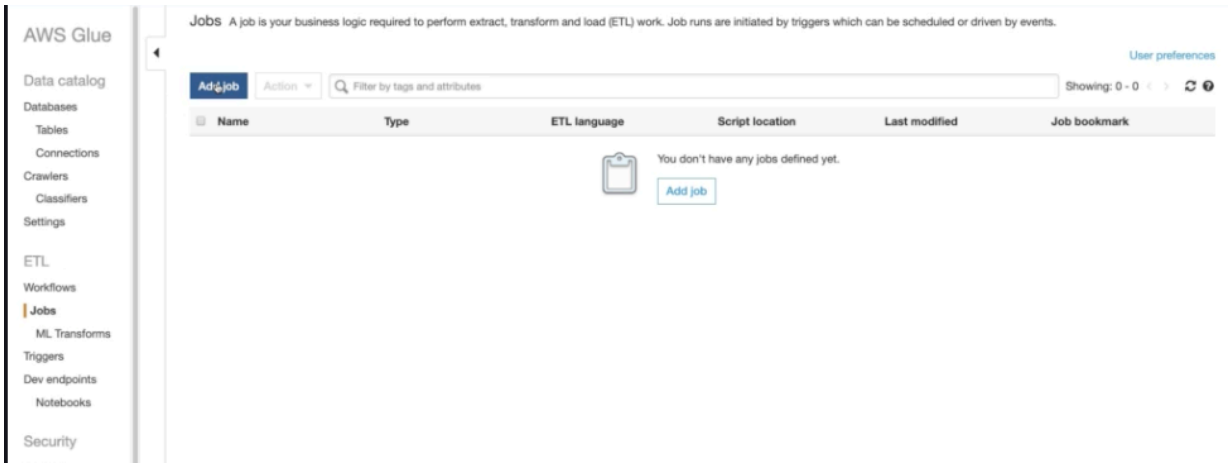
To help make the queries more intituitive

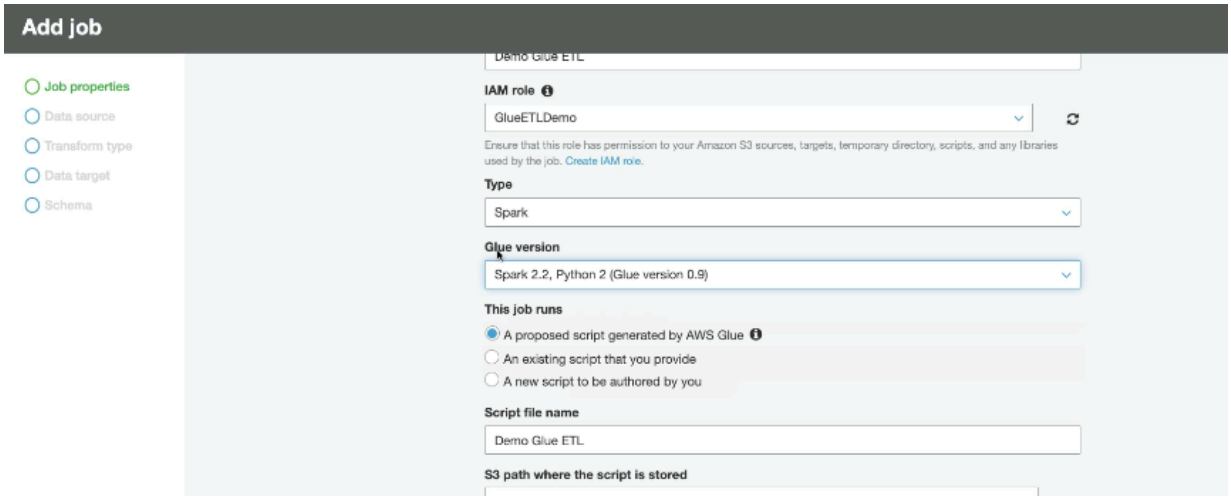
Glue ETL

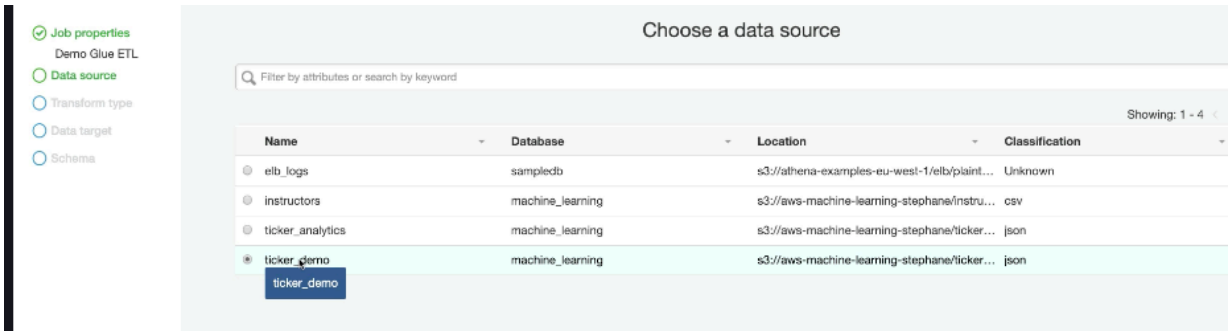
Glue ETL - Transformations

- 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

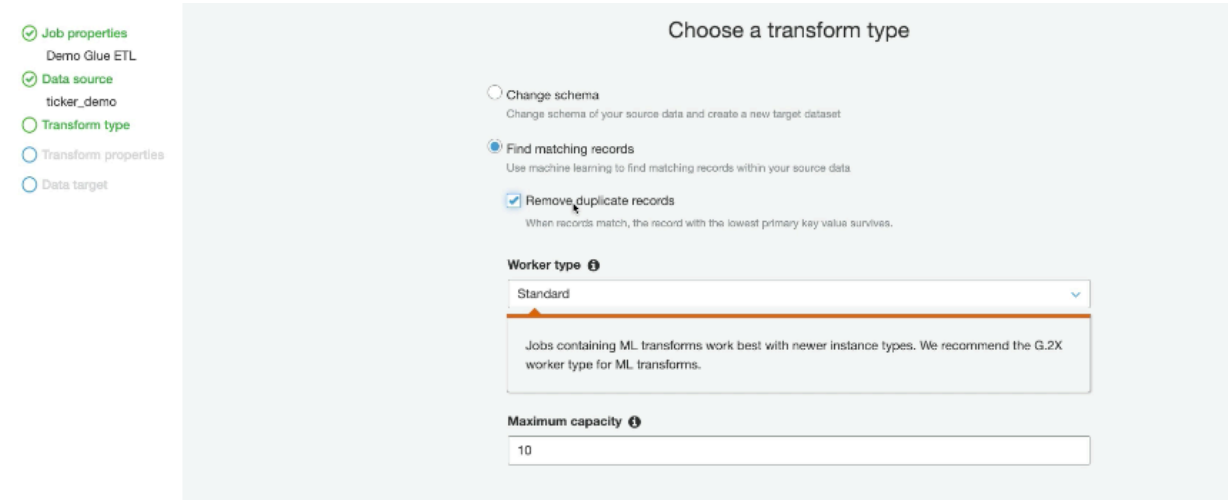
Glue ETL Lab



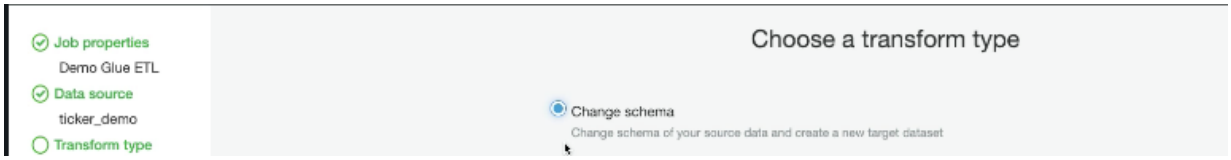


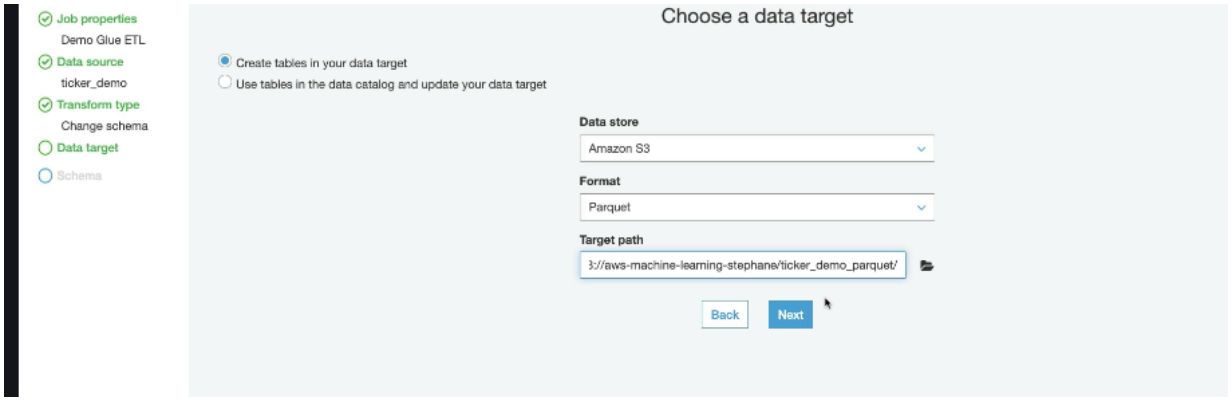


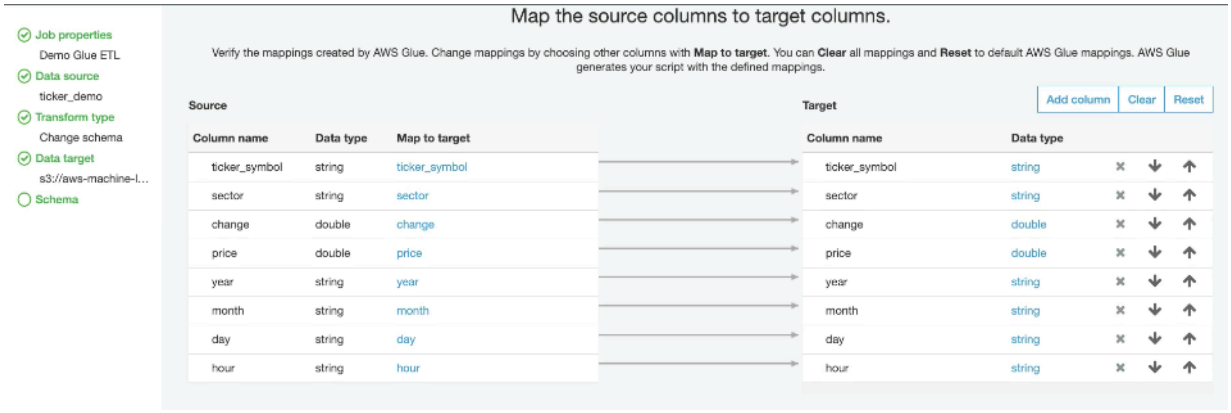
2 Options:
- Change Schema
- Or deduce with ML



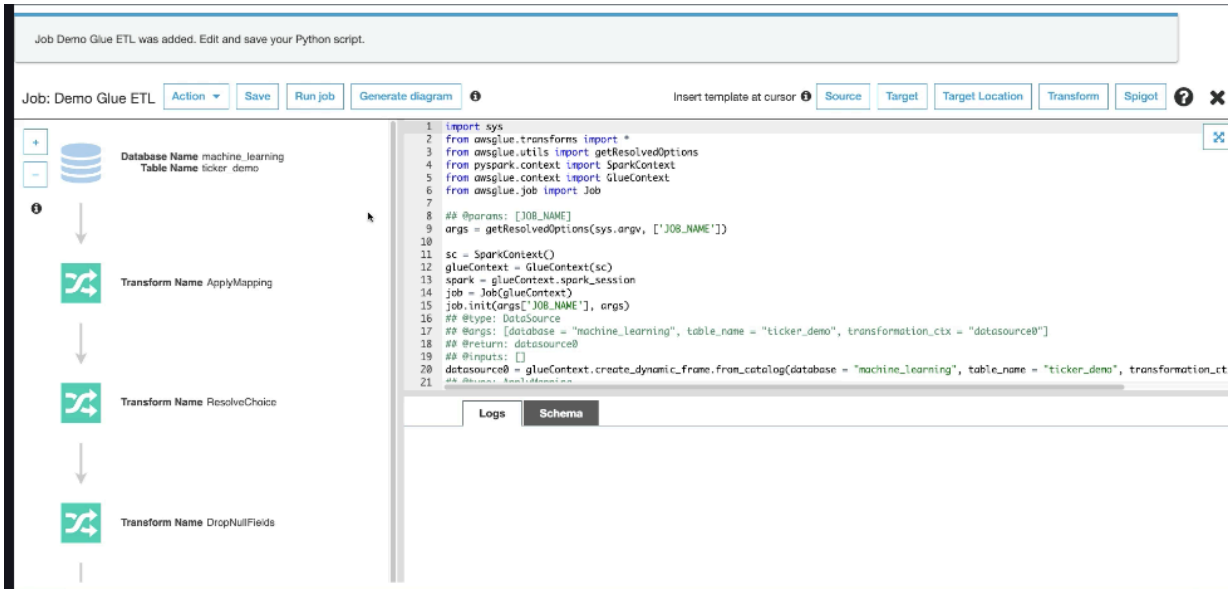
We will use change Schema for this lab

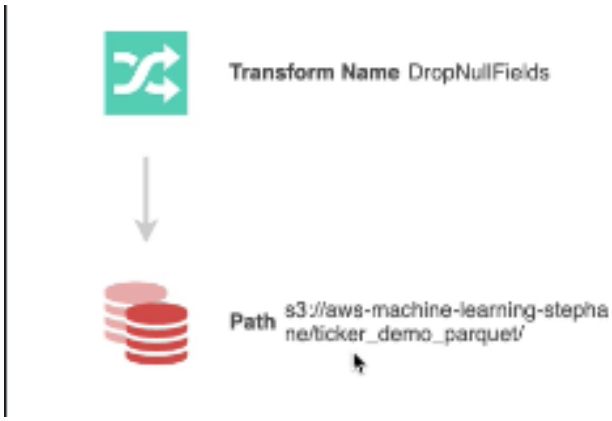






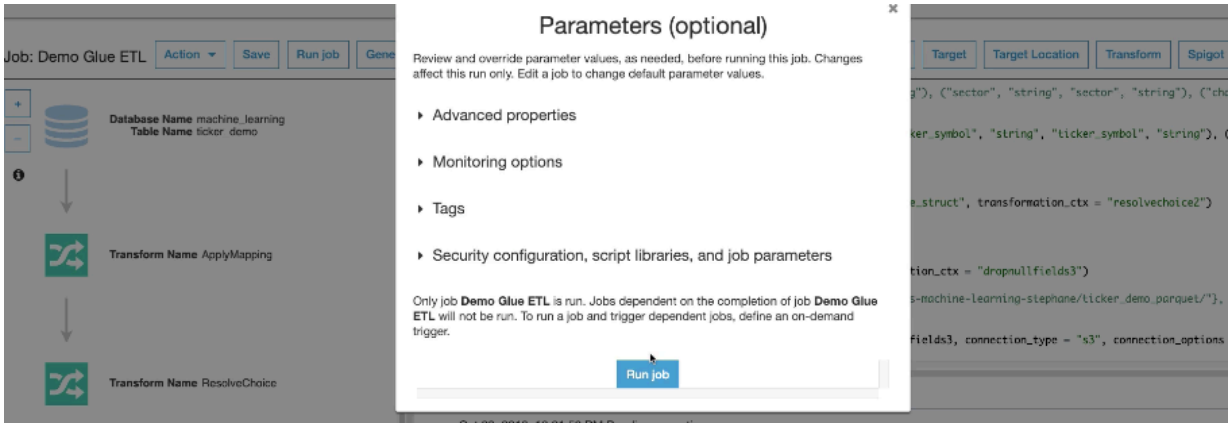
Now an entire ETL Glue job was generated for us



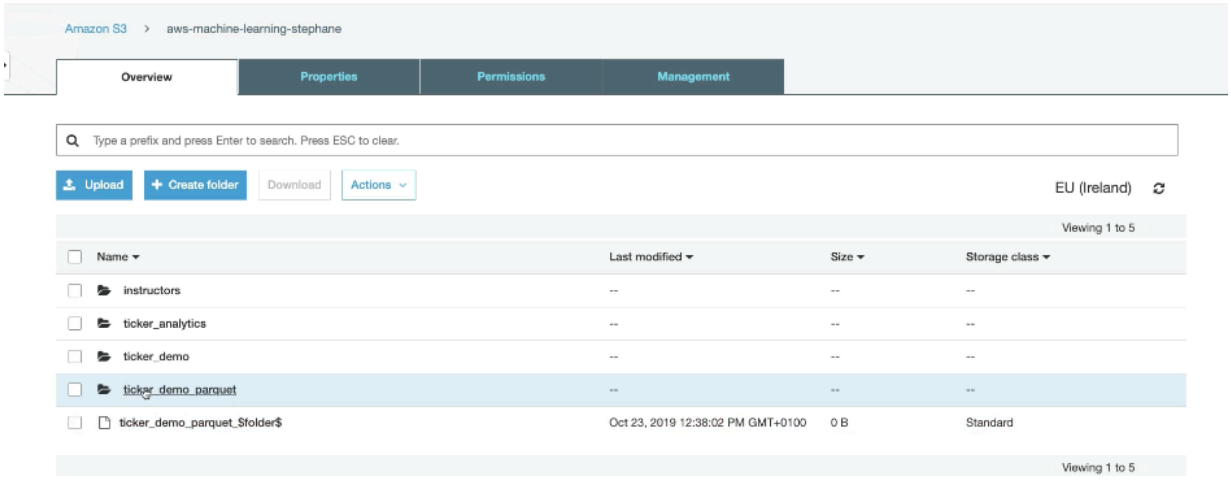


On the right side, there his spark code, that we can edit

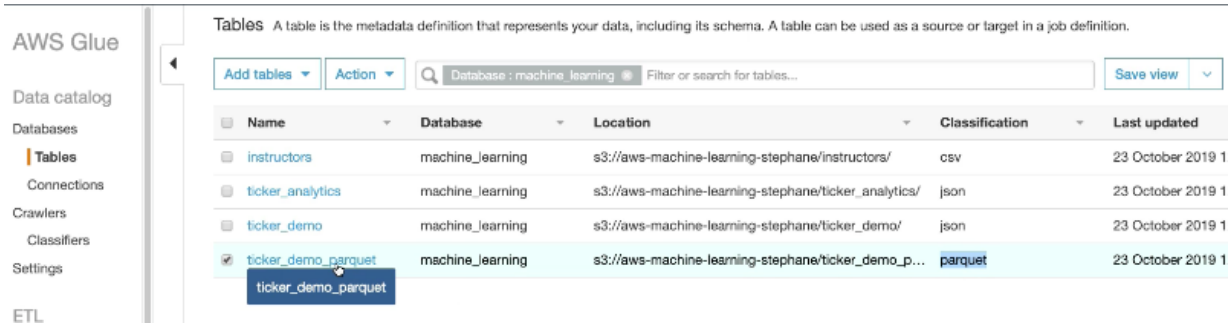
Now run the job

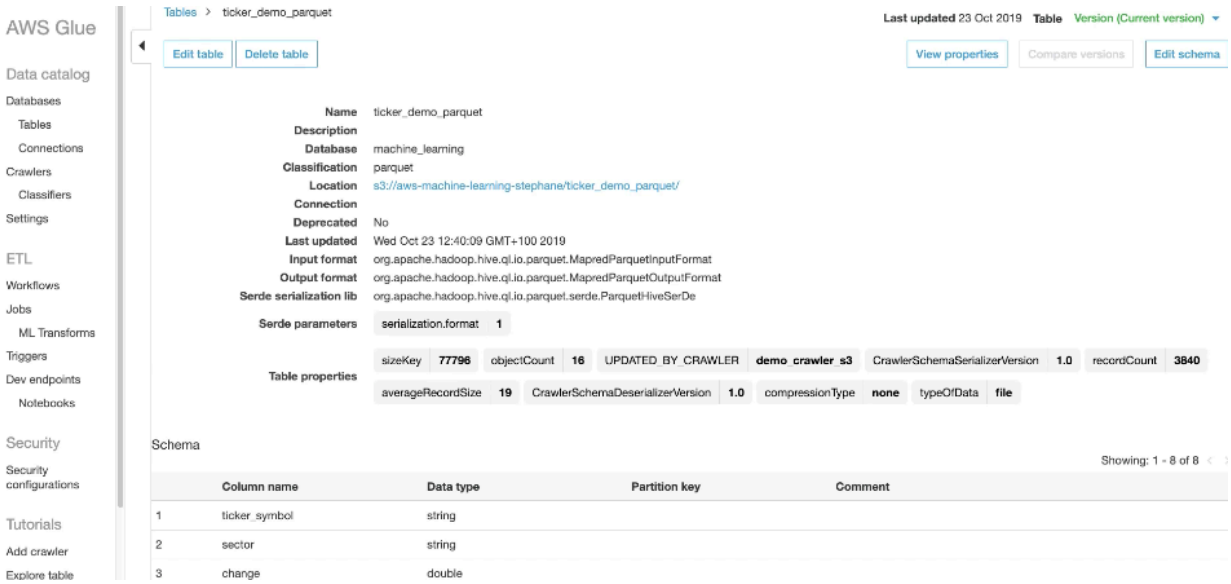


We can now see the parquet files in s3



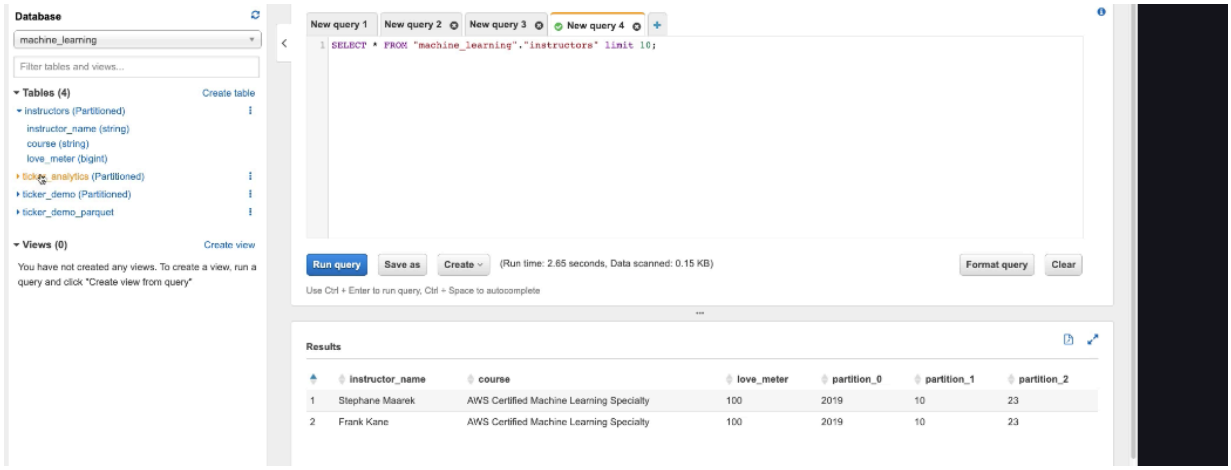
If we crawl again the same s3 bucket, the new parquet files are detected and a new table is added to the catalog



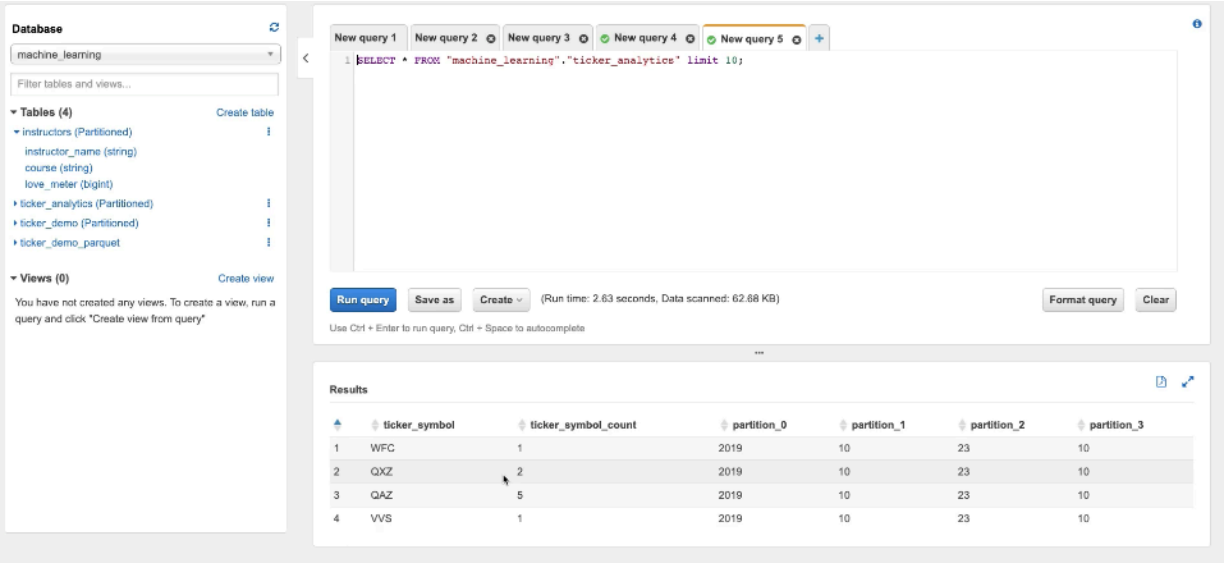


Athena Lab

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



We can also run SQL query against JSON


Database
machine_learning
Filter tables and views...
Tables (4)
Create table
instructors (Partitioned)
instructor_name (string)
course (string)
love_meter (bigint)
ticker_analytics (Partitioned)
ticker_demo (Partitioned)
ticker_demo_parquet
Views (0)
Create view
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
SELECT * FROM "machine_learning"."ticker_analytics" limit 10;
Run query
Save as
Create
(Run time: 2.63 seconds, Data scanned: 62.68 KB)
Format query
Clear
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