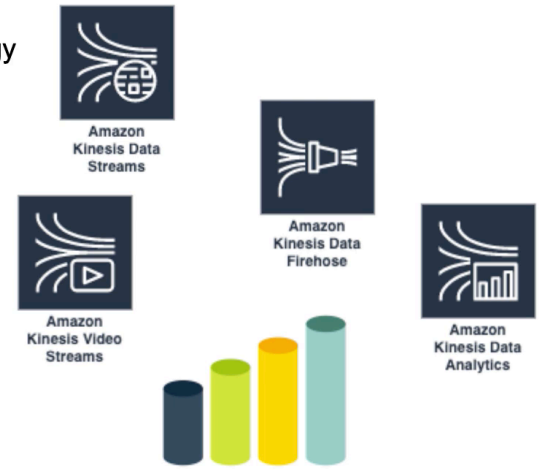
Whizlabs - ML Specialty Exam Course - Data Analysis - part 1

https://www.whizlabs.com/learn/course/aws-mls-practice-tests/video/3489

AWS Machine Learning Exploratory Data Analysis

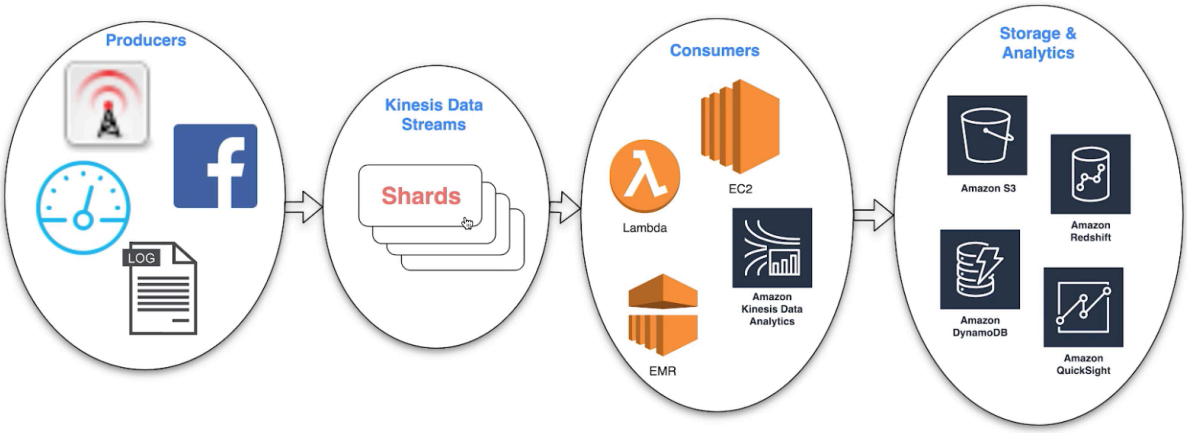
Introduction

- ❏ Basic data analysis concepts and terminology
- ❏ Kinesis Data Streams
- ❏ Kinesis Data Firehose
- ❏ Kinesis Video Streams
- ❏ Kinesis Data Analytics
- ❏ Visualize data for machine learning



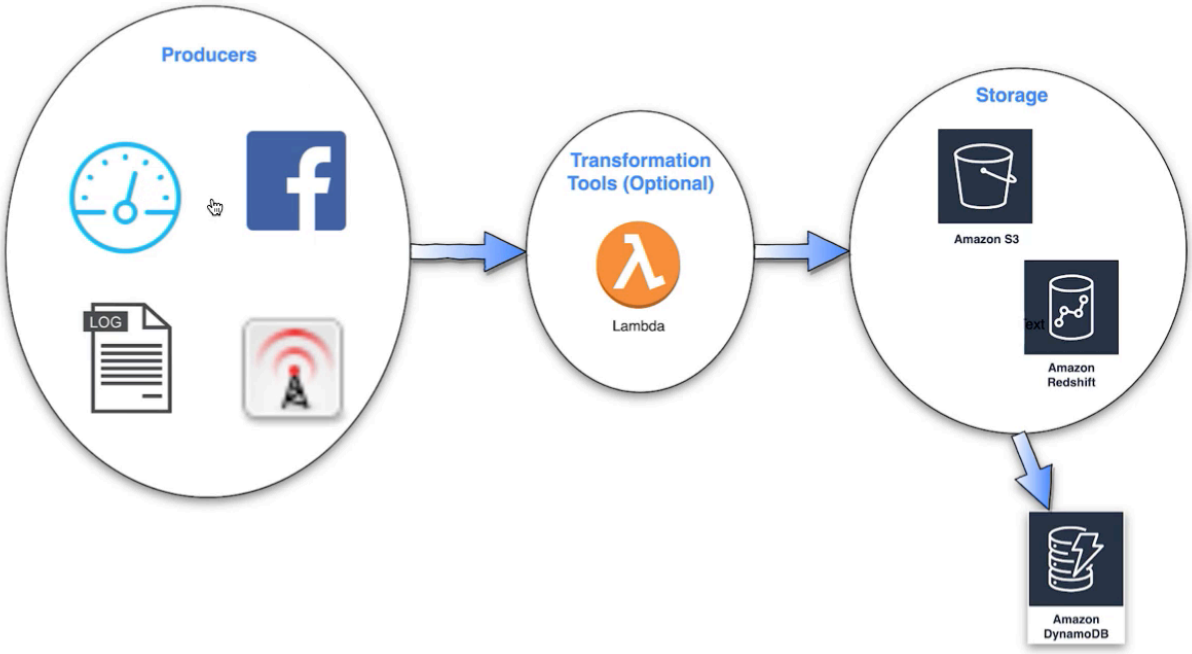
Kinesis Data Streams

- ❏ Gets data from data producers such as IoT, social media
- ❏ Uses shards to stream data to consumers such as EC2, lambda, Kinesis Data Analytics, EMR clusters
- ❏ Consumers then send data to a data repository such as S3, DynamoDB, Redshift, or Business Intelligence Tools



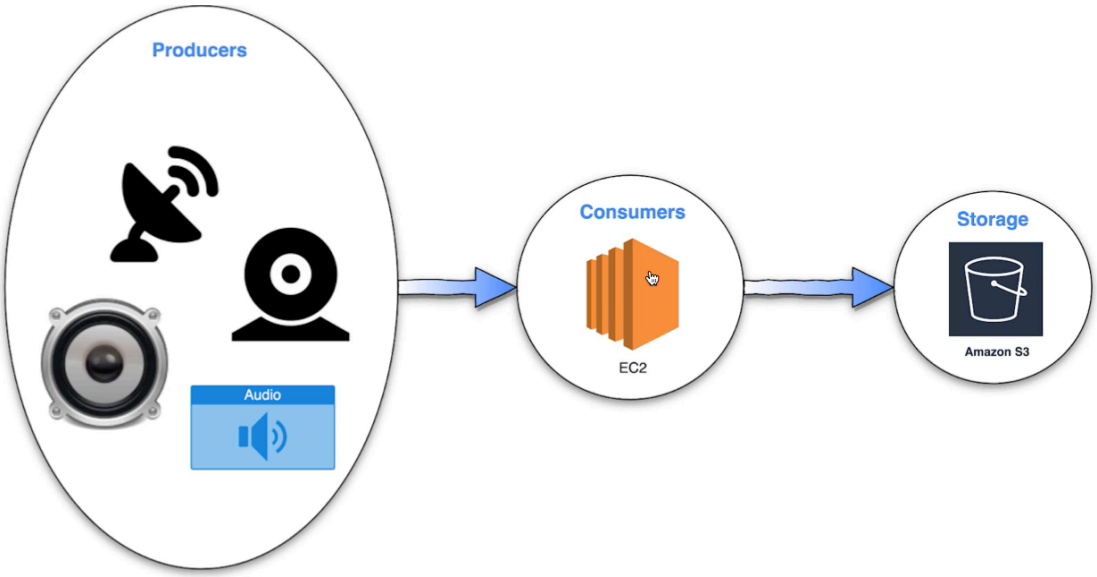
Kinesis Data Firehose

- ❏ Receives data from producers such as IoT, social media
- ❏ Uses Lambda functioning instead of shards to transmit producer data
- ❏ Lambda function puts data to data stores such as S3, Redshift, ElasticSearch, or splunk
- ❏ Can transmit directly from producers through Firehose to the data store (don't have to use lambda intermediary)
- ❏ S3 events to store to DynamoDB



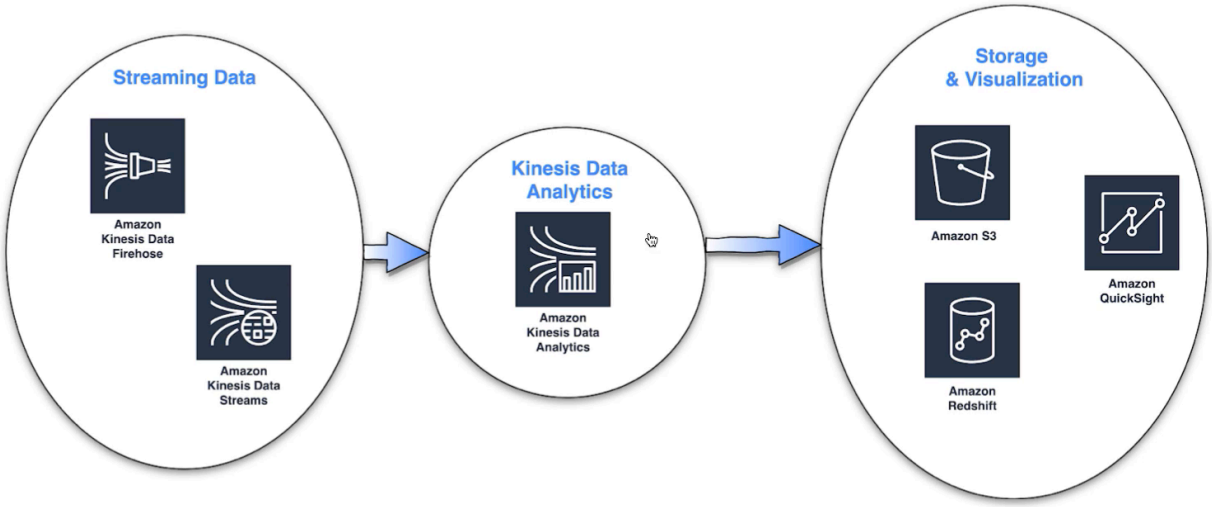
Kinesis Video Streams

- ❑ Build video processing applications such as machine learning models
- ❑ Producers such as web cams, security cameras, audio feeds, images
- ❑ Data Consumers - Kinesis Video Stream applications
- ❑ Stores to S3



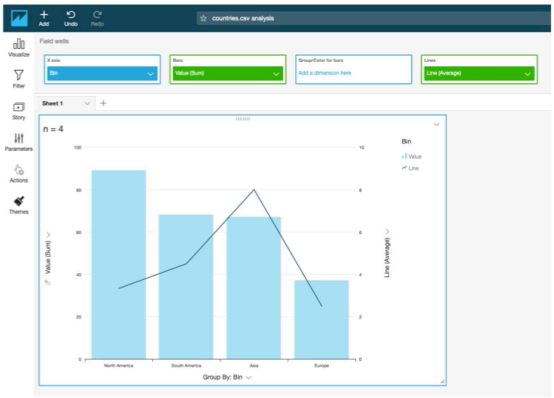
Kinesis Data Analytics

- ❏ Use SQL to process streaming data
- ❏ Sources: Kinesis Data Streams and Kinesis Data Firehose
- ❏ SQL queries put to S3, Redshift, or visualization and Business Intelligence tools



Visualizing Data for Machine Learning

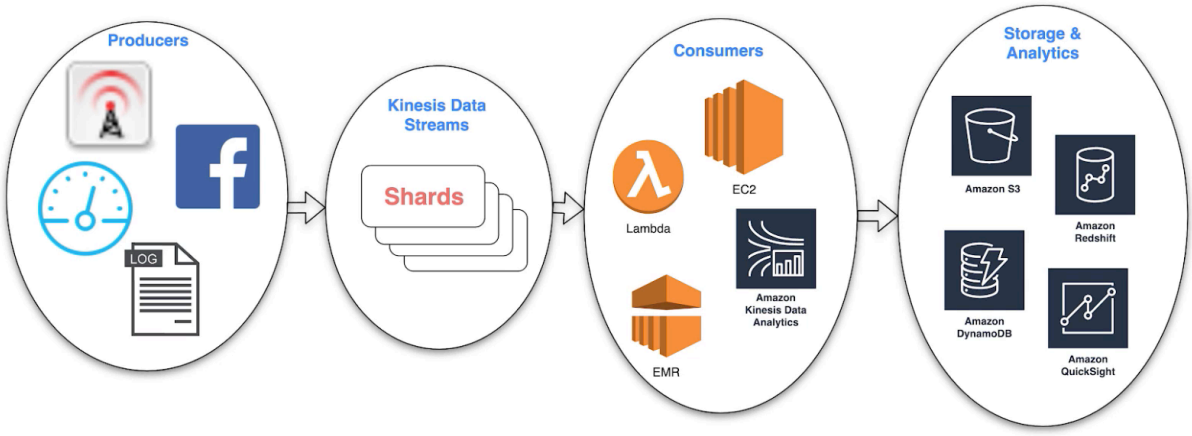
- ❏ Technique using static and interactive visuals to represent large amounts of data.
- ❏ Visualizes patterns, trends and correlations that may be difficult to discern
- ❏ Data visualization helps monetize data as a product



Kinesis Data Streams Details

AWS Machine Learning Exploratory Data Analysis

Kinesis Data Streams



- Massively scalable and durable real-time data streaming service
- Continuously capture gigabytes of data per second from thousands of sources
  - Website clickstreams
  - Database event streams
  - Social media feeds
  - Financial transactions
  - IoT events
  - Application logs
- Enables real-time analytics
  - Real-time dashboards
  - Real-time anomaly detection
  - Dynamic pricing
  - Real-time fraud detection

Kinesis Data Stream - 4 Key concepts

Key Concepts

- Data Producer
  - An application that emits data records as they are generated
- Data Consumer
  - AWS service or distributed Kinesis application that retrieves data from Kinesis Data Streams
- Shard
  - A shard is the base throughput unit of a Kinesis Data Stream
  - Data producers assign partition keys to records
  - Partition keys ultimately determine which shard ingests the data record for a data stream
  - Data consumers retrieve data from all shards in a stream as the data is generated
- Data Stream
  - A logical grouping of shards
  - Data stream will retain data for 24 hours, or up to 7 days with extended retention enabled

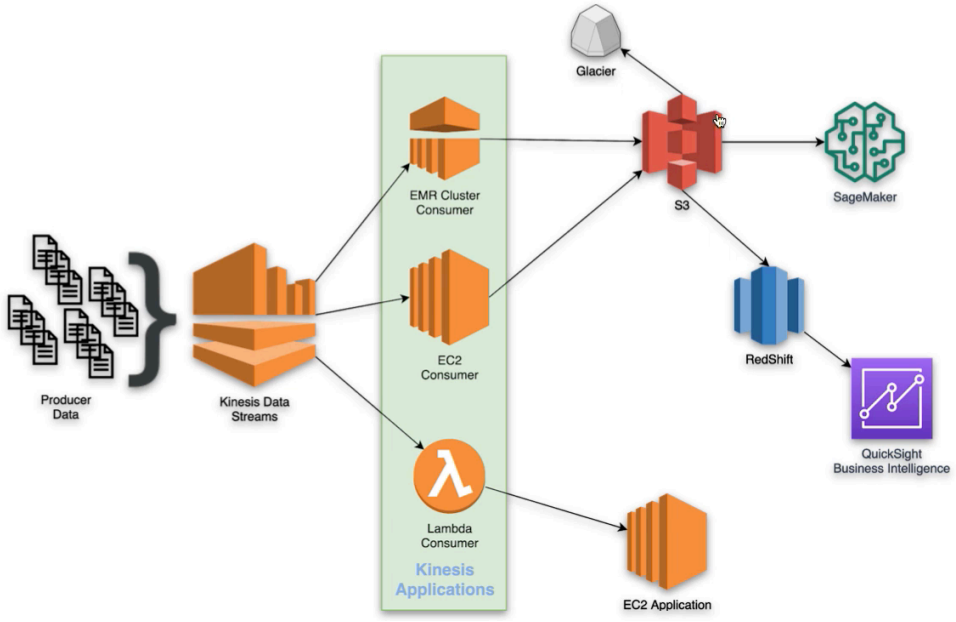
AWS Machine Learning Kinesis Data Streams

Putting Data Into Streams

- Data producers put data into Amazon Kinesis data streams using the Kinesis Data Streams APIs
  - Amazon Kinesis Producer Library
    - Highly configurable library that puts data into an Amazon Kinesis data stream
    - Simple, asynchronous, reliable interface to achieve high producer throughput
  - Amazon Kinesis Agent
    - Pre-built Java application that collects and sends data to your Amazon Kinesis stream
    - Install the agent on web servers, log servers, and database servers
    - Agent monitors files/database resources and continuously sends data to your stream

Kinesis Data Stream architecture

Kinesis Data Streams Big Data Architecture

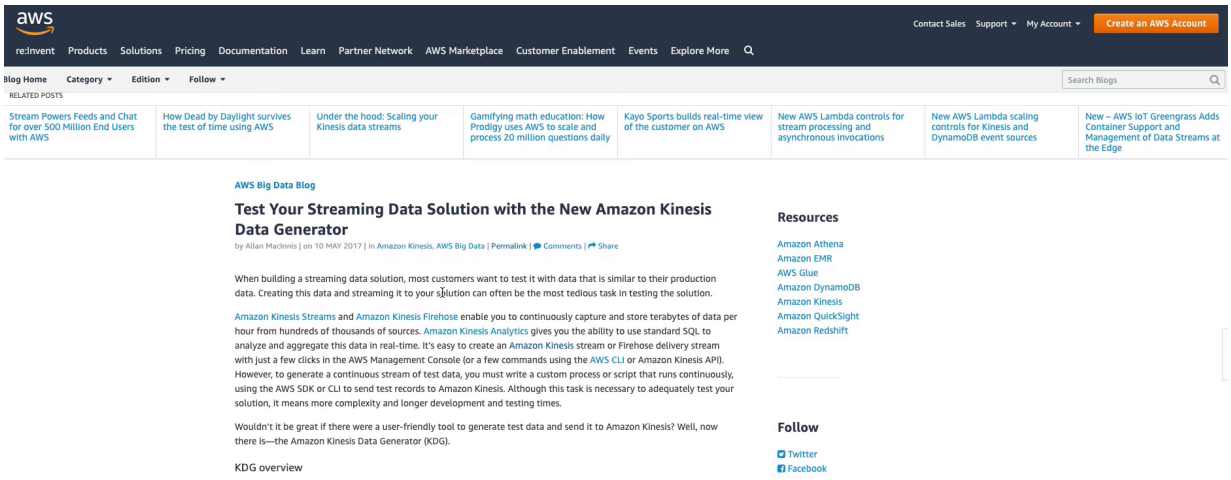


Additional Key Points

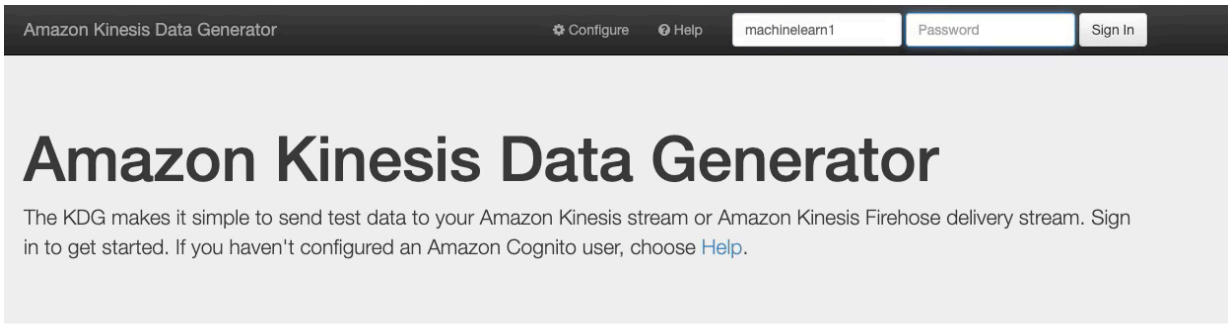
- ❑ Shards are append-only logs
- ❑ Shards contain ordered sequence of records ordered by arrival time
- ❑ One shard can ingest up to 1000 data records per second, or 1MB/sec
- ❑ Specify the number of shards needed when you create a stream
- ❑ Add/remove shards from stream dynamically as throughput changes via API, Lambda, auto scaling
- ❑ Enhanced fan-out: one shard allows 1MB/sec in and 2MB/sec out for each consumer
- ❑ Non-enhanced fan-out: one shard allows 1MB/sec in and 2MB/sec out shared across consumers
- ❑ Monitor shard-level metrics in Amazon Kinesis Data Streams

Kinesis Data Stream - Lab

Using Kinesis Data Generator - KDG for this lab

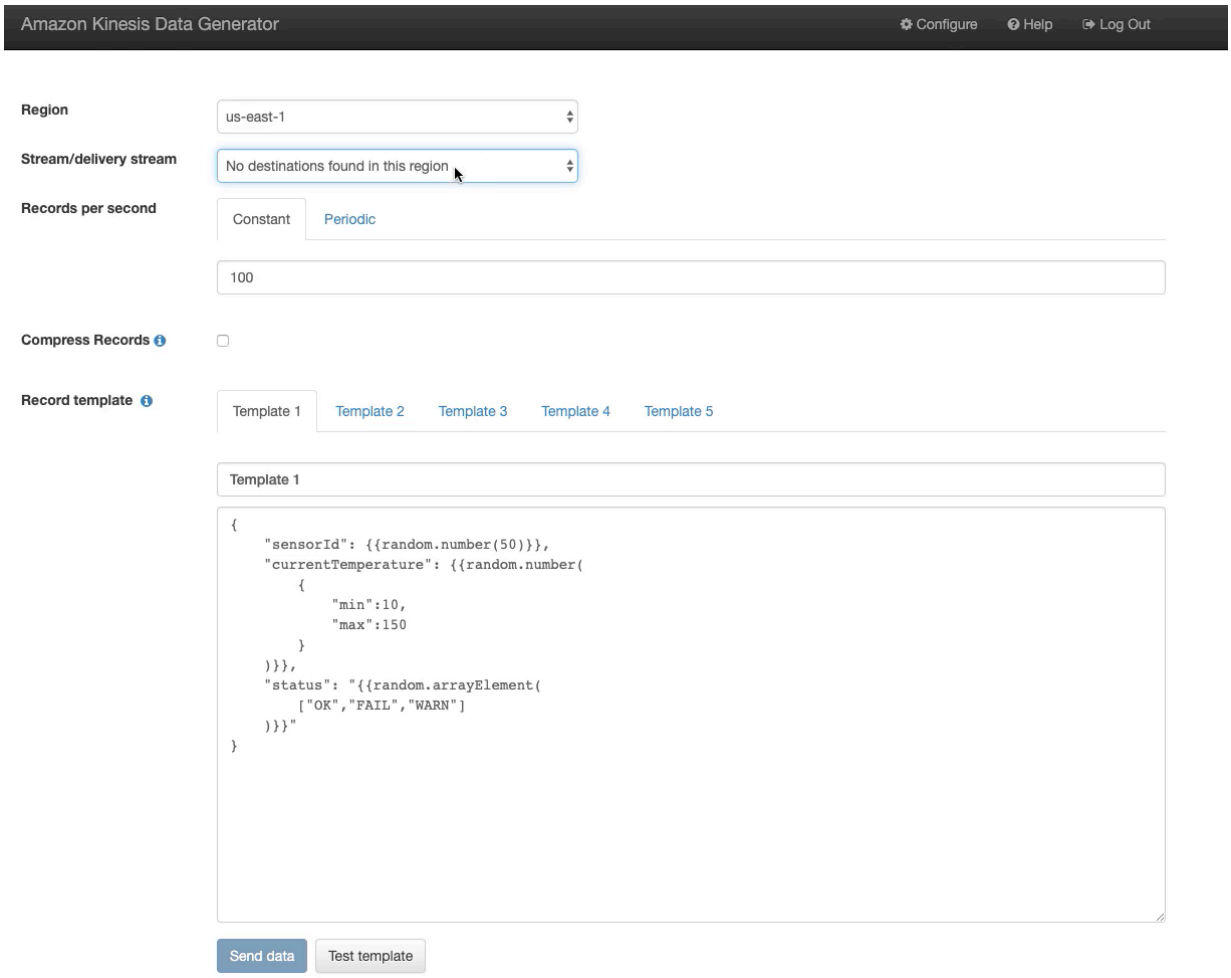


Create the stack and log in to KDG



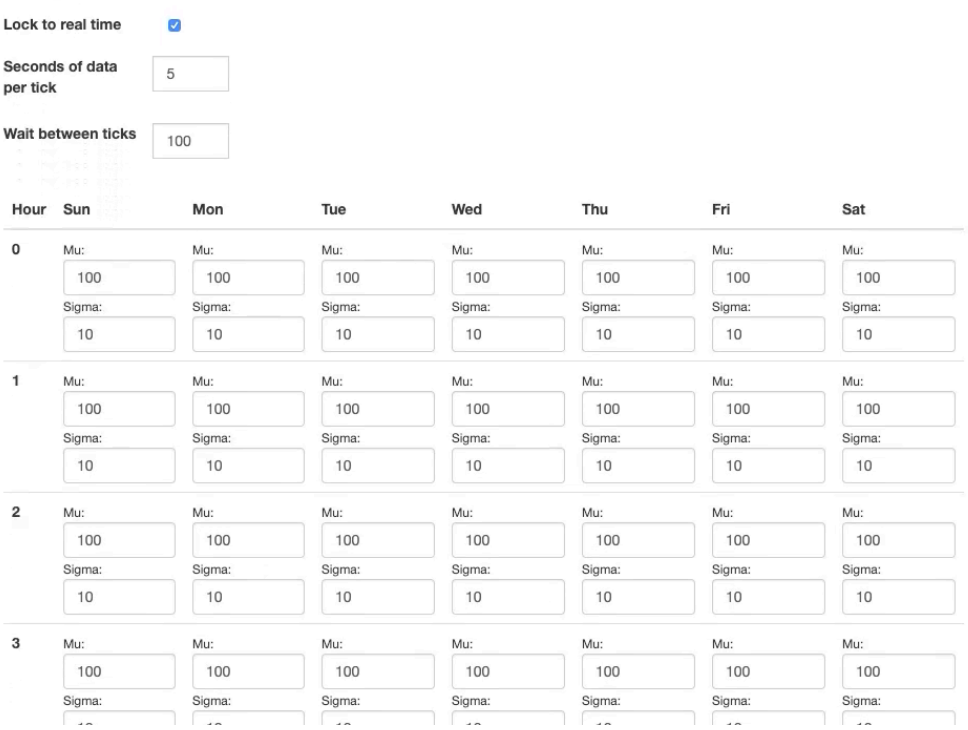
We will use a sensor template to generate data

```
{
    "sensorId": {{random.number(50)}},
    "currentTemperature": {{random.number(
        {
            "min":10,
            "max":150
        }
    )}},
    "status": "{{random.arrayElement(
        ["OK","FAIL","WARN"]
    )}}"
}
```

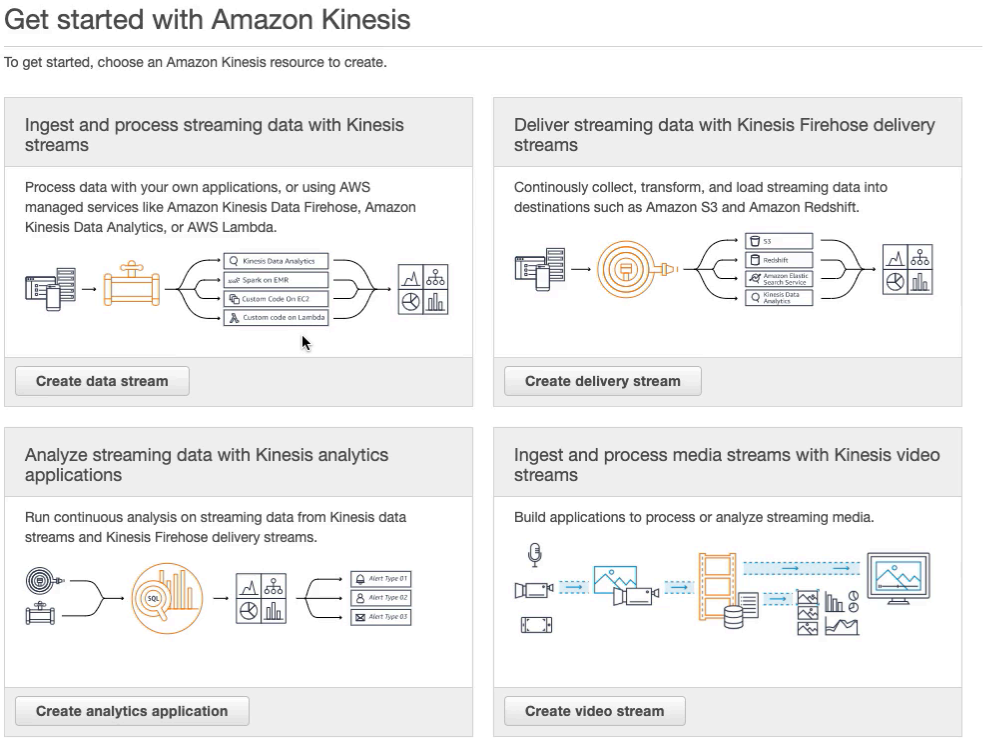


Sample Records

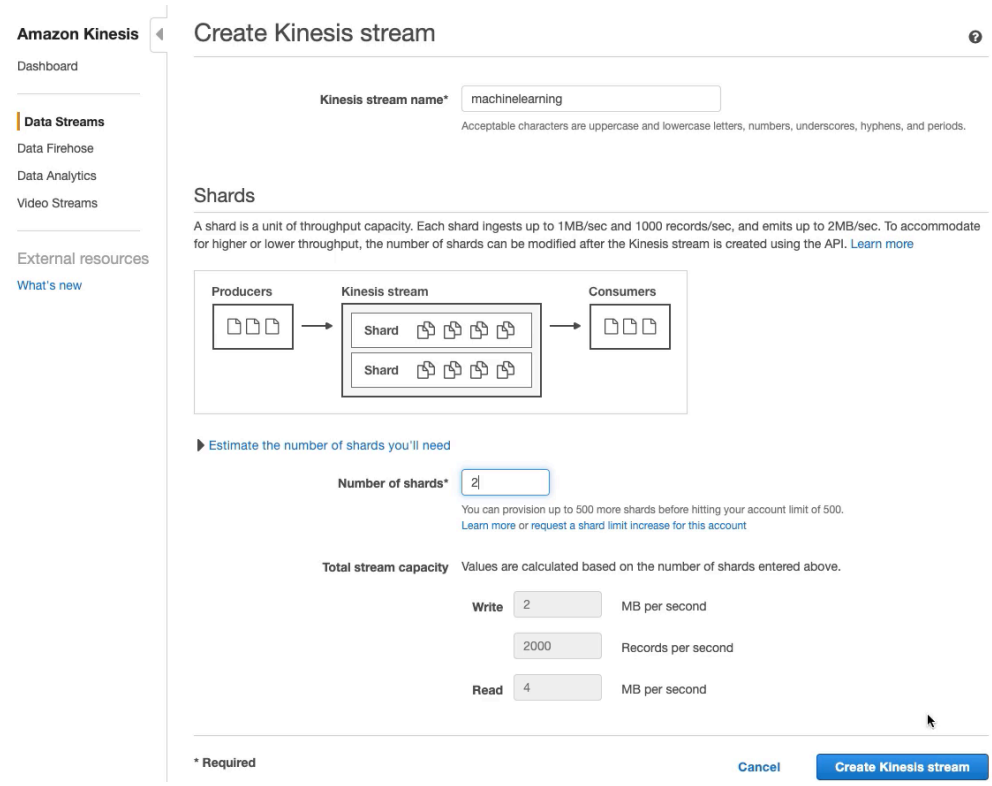

```
{    "sensorId": 14,    "currentTemperature": 28,    "status": "FAIL"}
{    "sensorId": 46,    "currentTemperature": 11,    "status": "WARN"}
{    "sensorId": 44,    "currentTemperature": 94,    "status": "OK"}
{    "sensorId": 24,    "currentTemperature": 81,    "status": "WARN"}
{    "sensorId": 11,    "currentTemperature": 110,    "status": "WARN"}
```



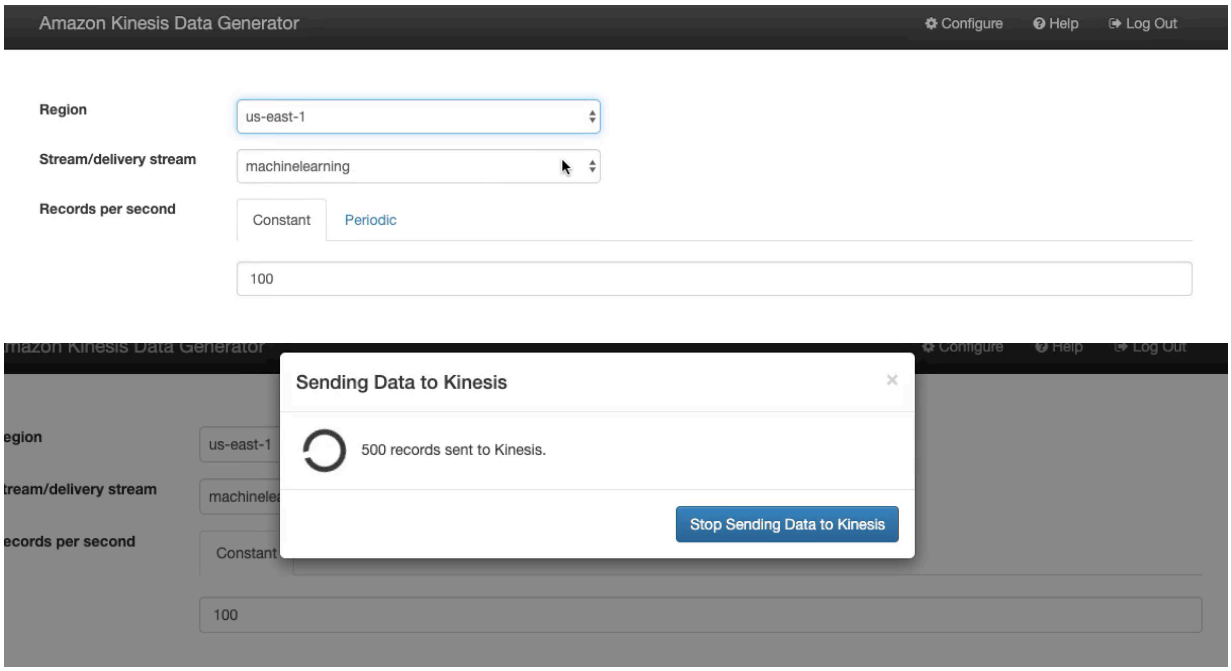
Now create a Data stream



With 2 shards



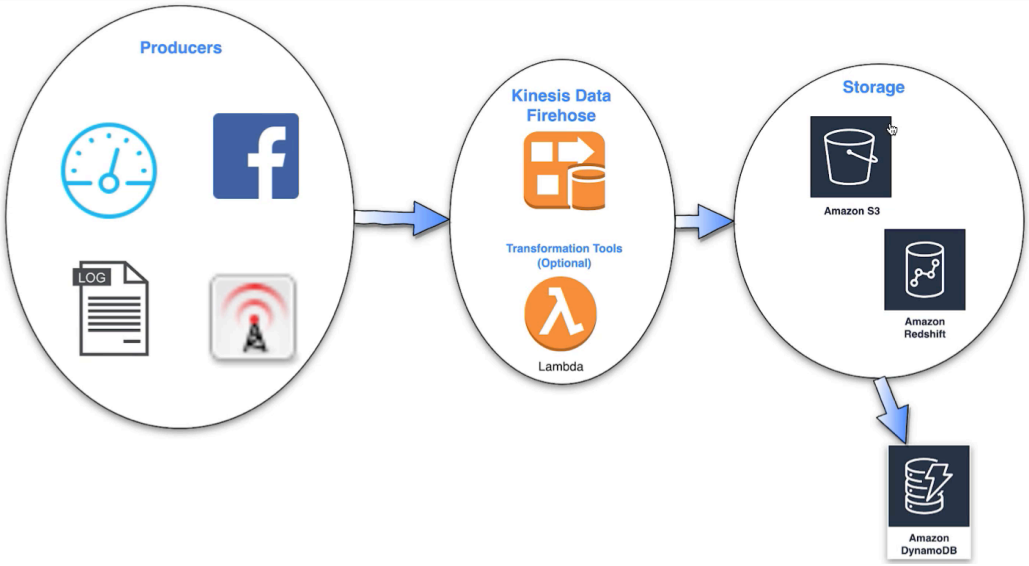
Now we can plug the data stream into KDG and send data



In next lab, we will be consuming that dataset with Kinesis Firehose

AWS Machine Learning Exploratory Data Analysis

Kinesis Data Firehose



- ❏ Fully managed service that automatically scales to match data throughput
- ❏ Capture, transform, and load streaming data into S3, Redshift, Elasticsearch, and Splunk
- ❏ Batch, compress, transform, and encrypt data before loading it onto your destination
  - ❏ Minimizes the storage used at destination and increases security
- ❏ Automatically convert incoming data to Apache Parquet/ORC before delivering to S3
- ❏ Near real-time analytics with existing business intelligence tools
- ❏ Requires no ongoing administration

Kinesis Firehose Key concepts

Key Concepts

- Kinesis Data Delivery Stream
  - Underlying entity of Kinesis Data Firehose
  - Create a delivery stream via the AWS console then send it data
- Elastic scaling handles variations in data throughput
  - Automatically scale to handle gigabytes per second input rate
  - Maintain data latency at specified levels
  - Requires no intervention or maintenance
- Transform your data using Lambda
  - Automatically apply Lambda function to every input data record
  - Pre-built Lambda blueprints for converting common data sources such as Apache/system logs to JSON and CSV formats

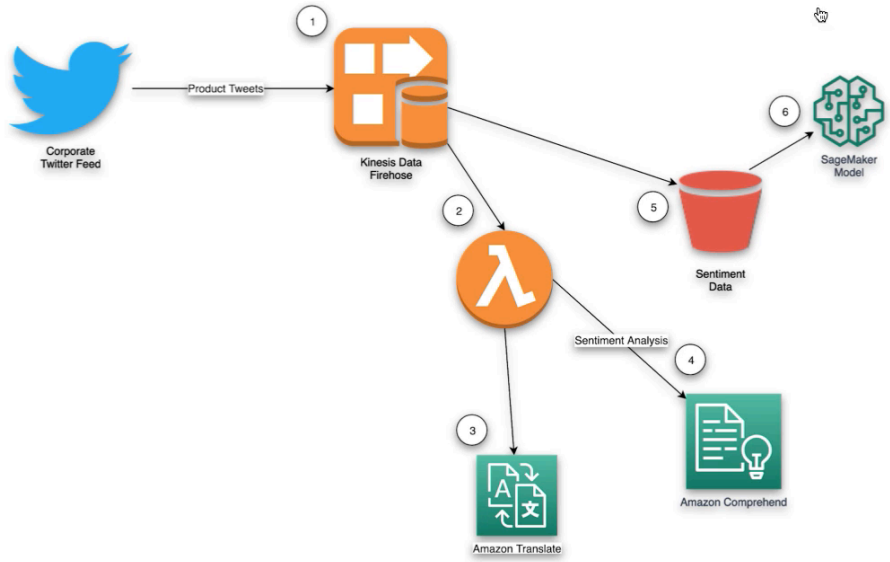
Putting Data Into Delivery Streams

- Send data to the delivery stream
  - Call the Firehose API from your data producer application
  - Run the Linux agent on your data source
- Control the delivery pace to your delivery stream
  - Specify a batch size or batch interval to control how quickly data is delivered
  - Compress the data stream using GZip or Snappy

Kinesis Firehose - Architecture example

Tweets get produced to Firehose
Use lambda function to transform these tweets to Translate (3) and Comprehend (4) for sentiment analysis
Then we send the data to S3 and use it into SageMaker to forecast next best pb to hold a large marketing campaign based on sentiment of the tweets

Social Media Stream



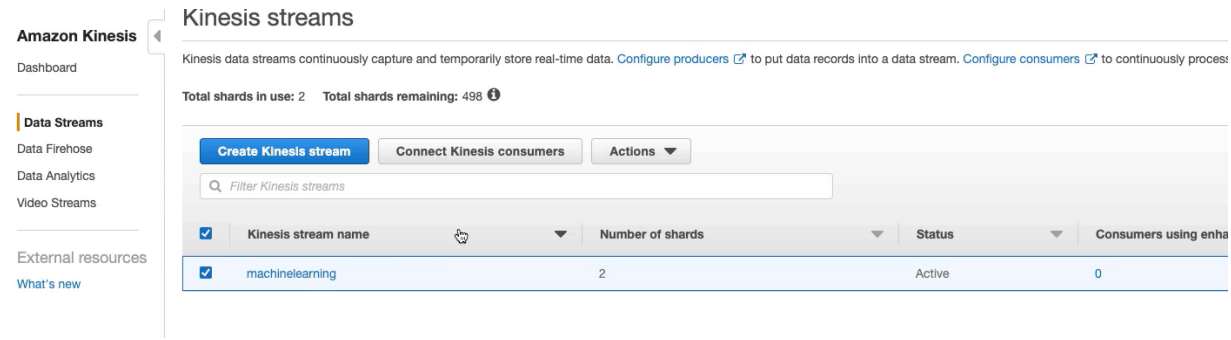
Additional Key Points

- ❏ Automatic encryption using AWS Key Management System (KMS) encryption key
- ❏ Track metrics of your stream
  - ❏ Volume of data submitted to the stream
  - ❏ Volume of data sent to destination
  - ❏ Time from source to destination
  - ❏ Delivery success rate
- ❏ Pay-as-you-go pricing
  - ❏ Pay only for the volume of data you transmit through the service

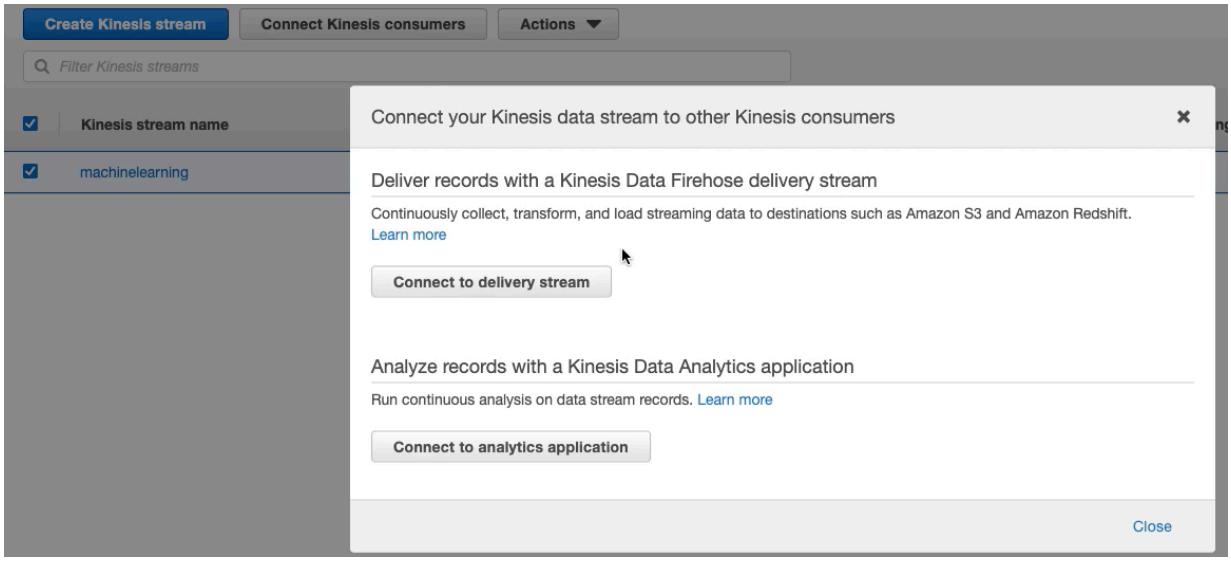
Kinesis Firehose - Lab

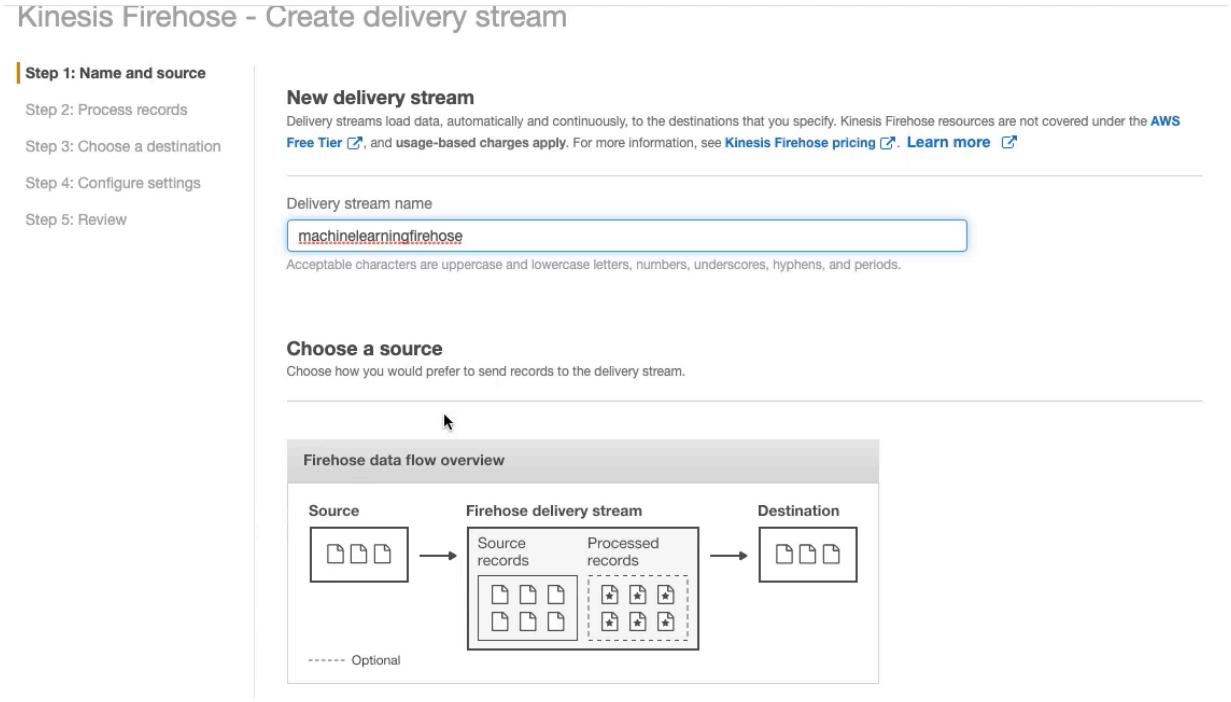
Using data from the previous lab, where data is produced to Kinesis Data Stream

In our Kinesis Data Stream:

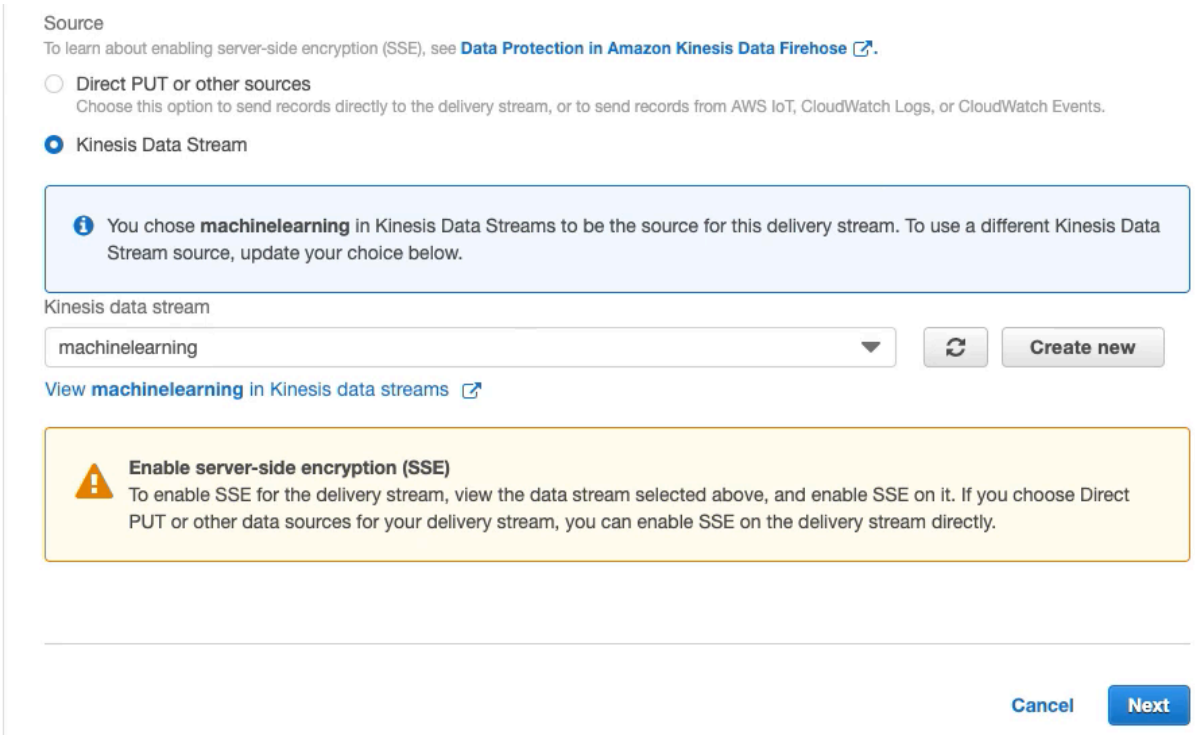


We click on "Connect Kinesis consumers" and add a FireHose consumer

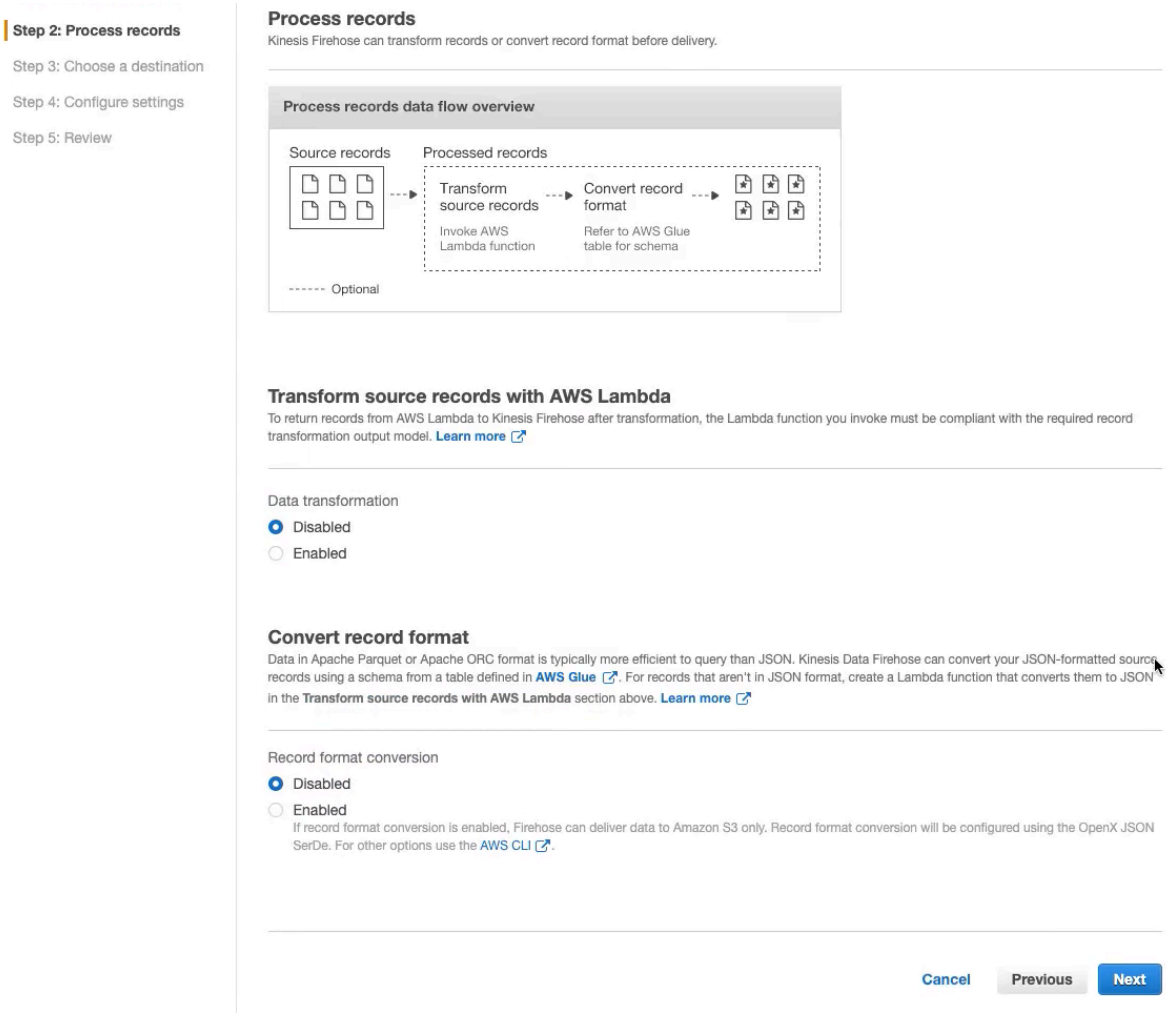




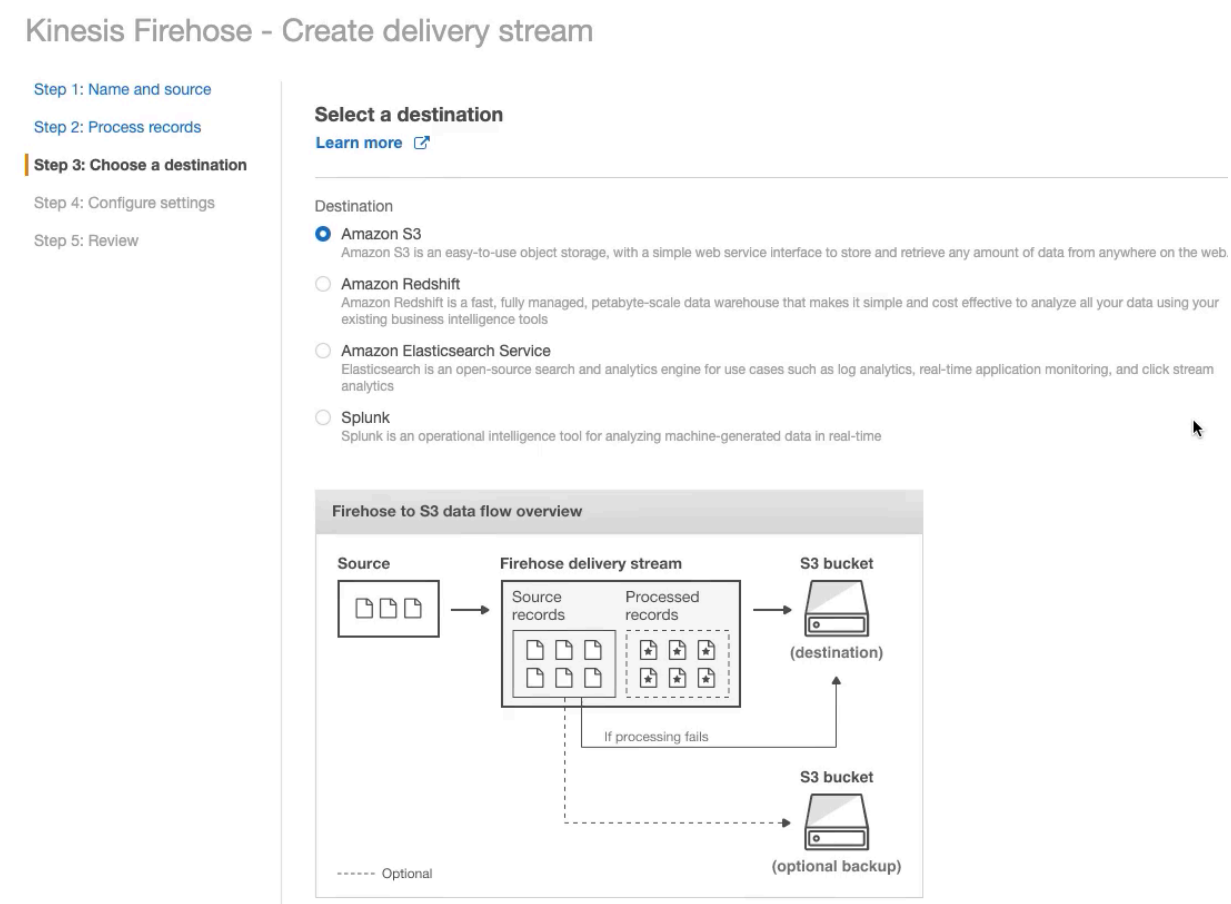
Use Kinesis Data Stream as the source (pre-selected)

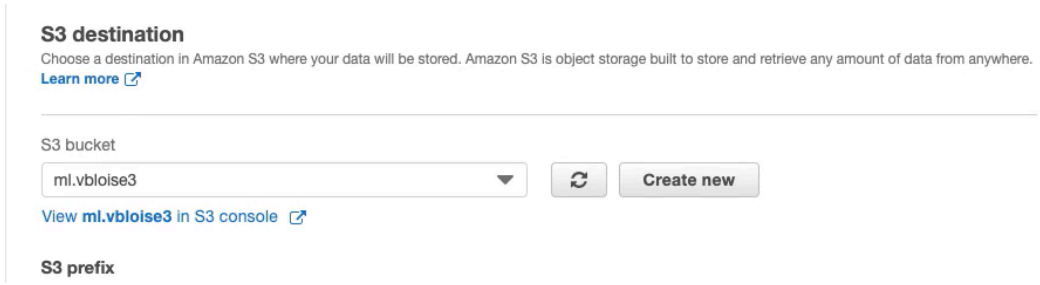


We are not going to transform the data with a Lambda, nor convert it so we will keep both options disabled

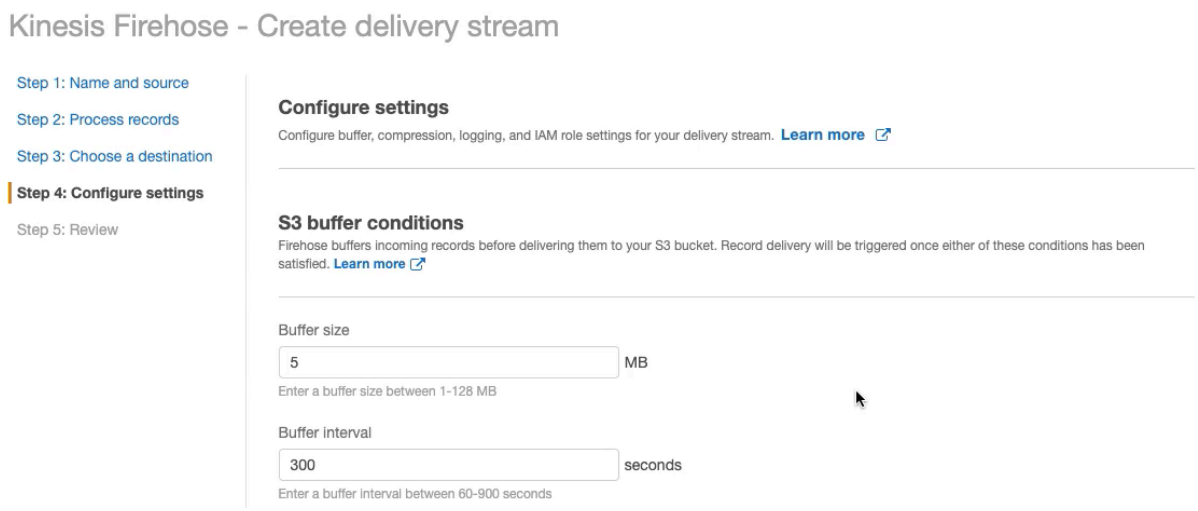


Use S3 as final destination

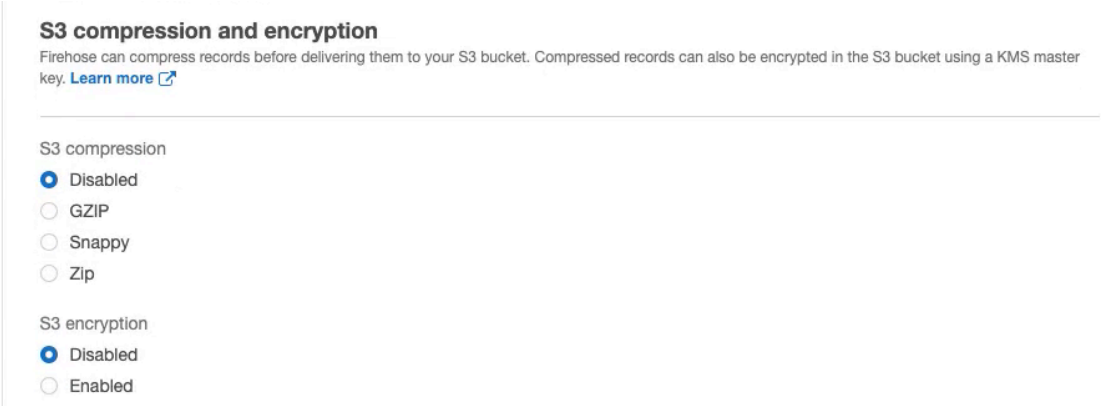




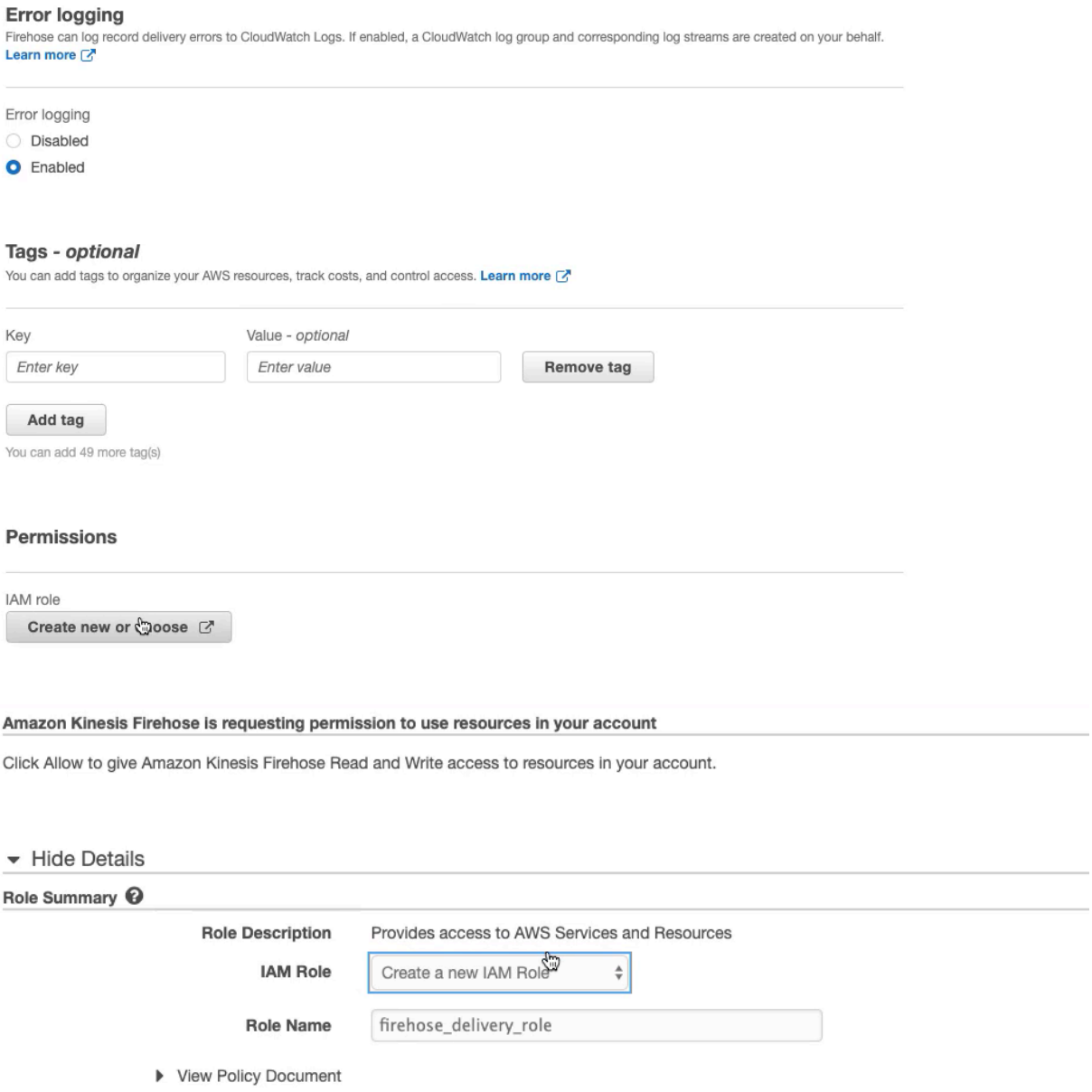
Ih the “Configure Settings”, we keep the buffer size default



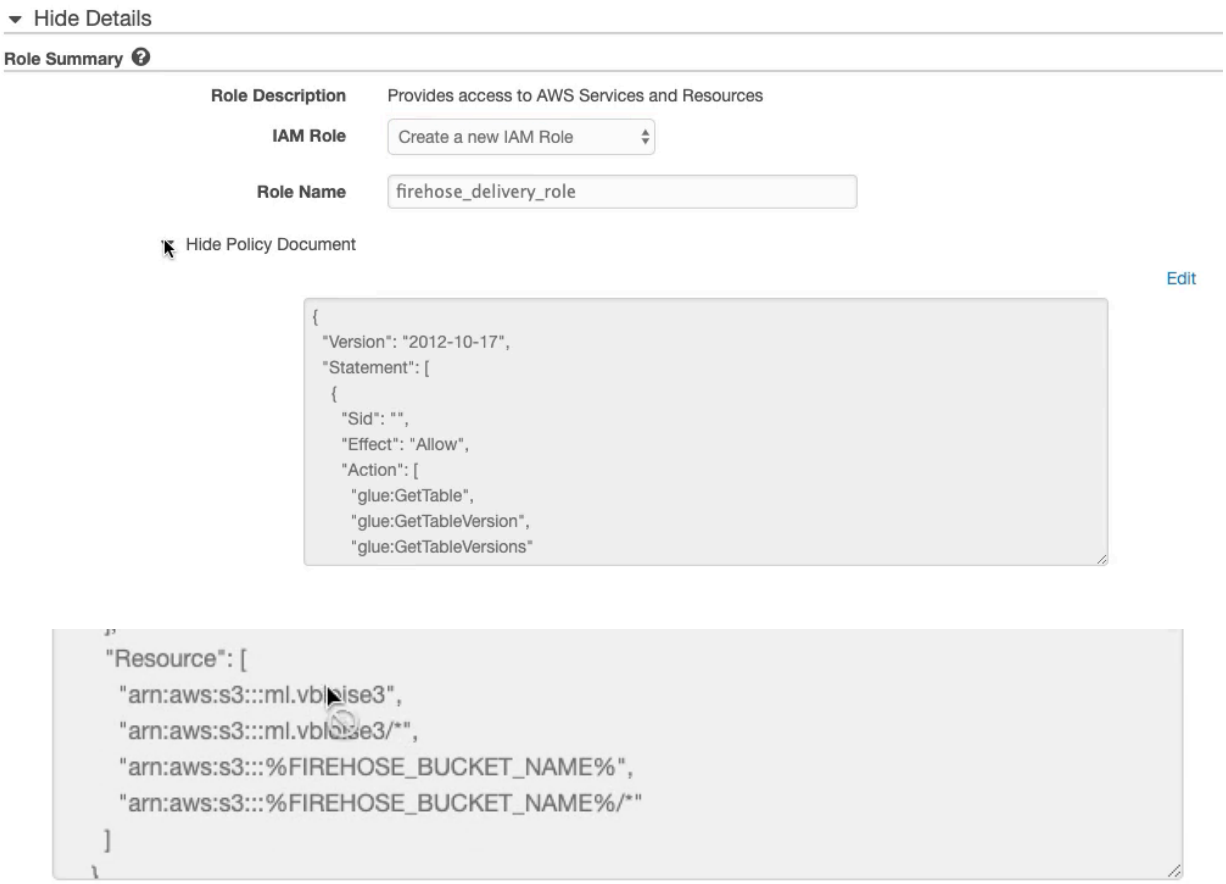
We don’t need to compress or encrypt - but we could



We keep Error logging to CloudWatch enabled (by default)
And w create an IAM role to allow Firehose to retrieve records from the steam and write records to S3

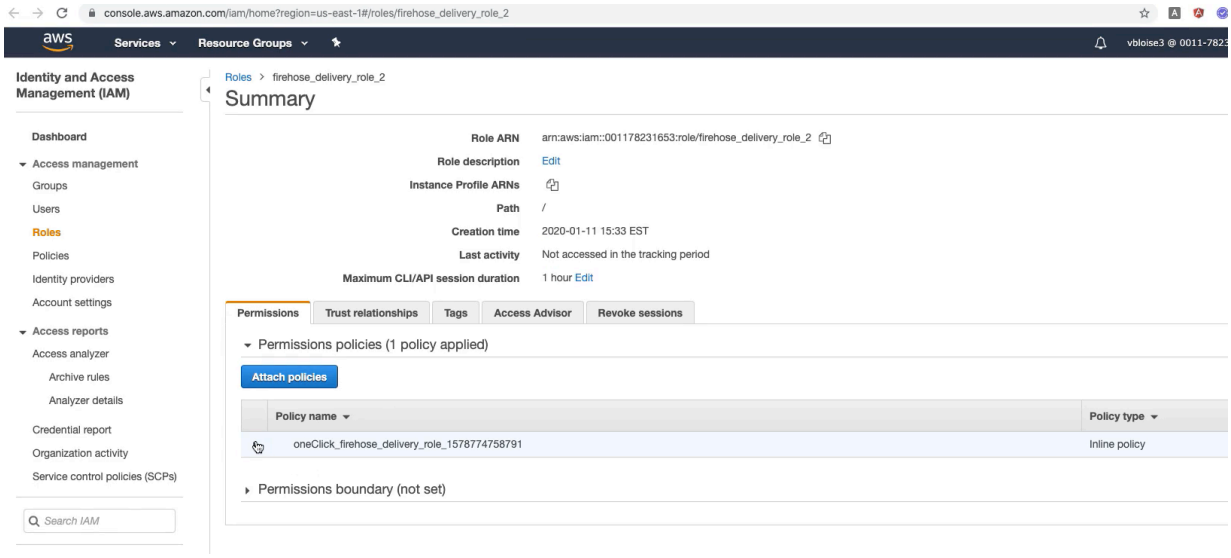


Extending policy documents:

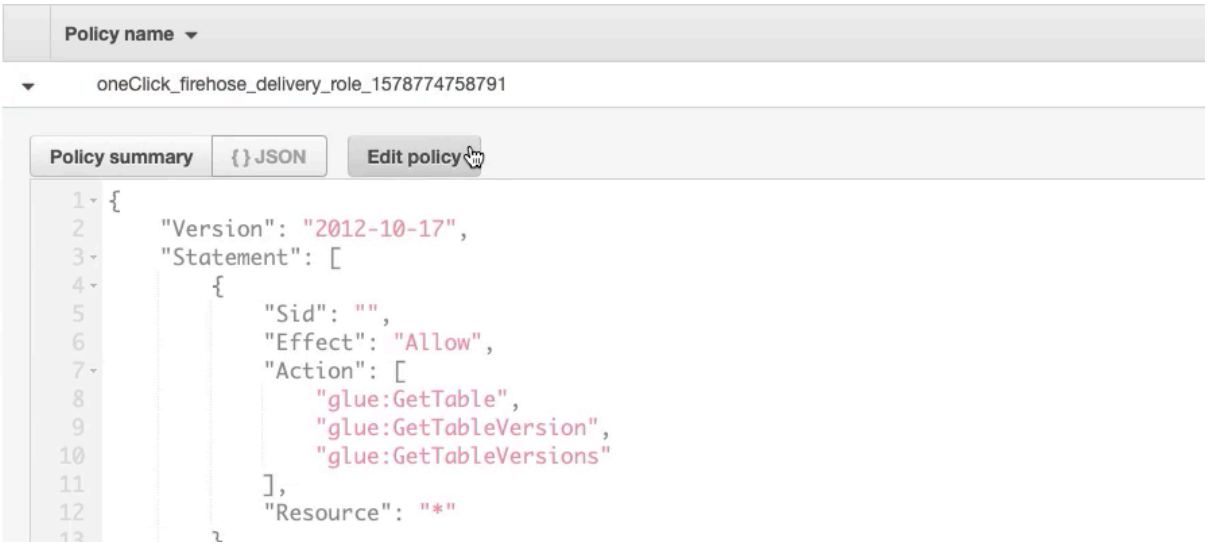


Click on Allow

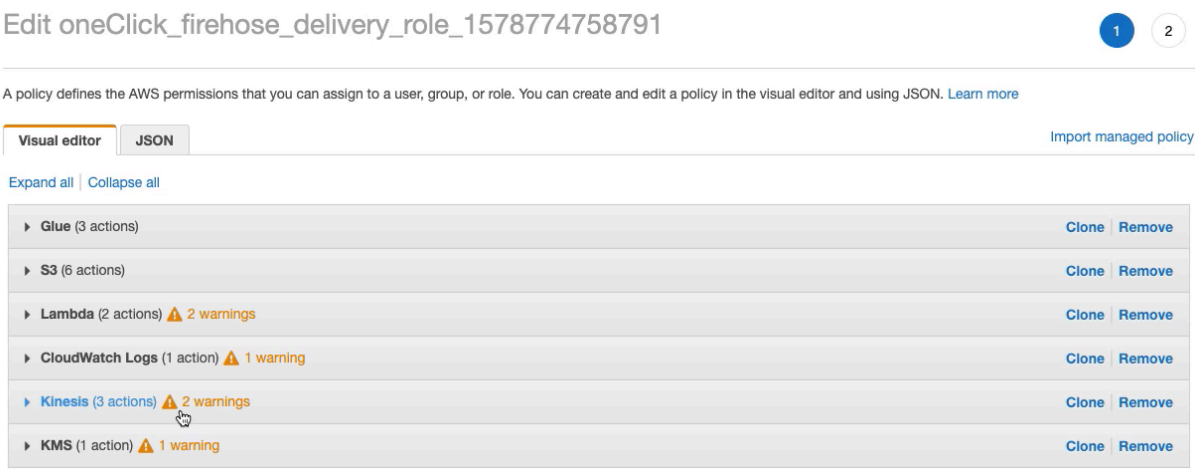
Now checking the role:



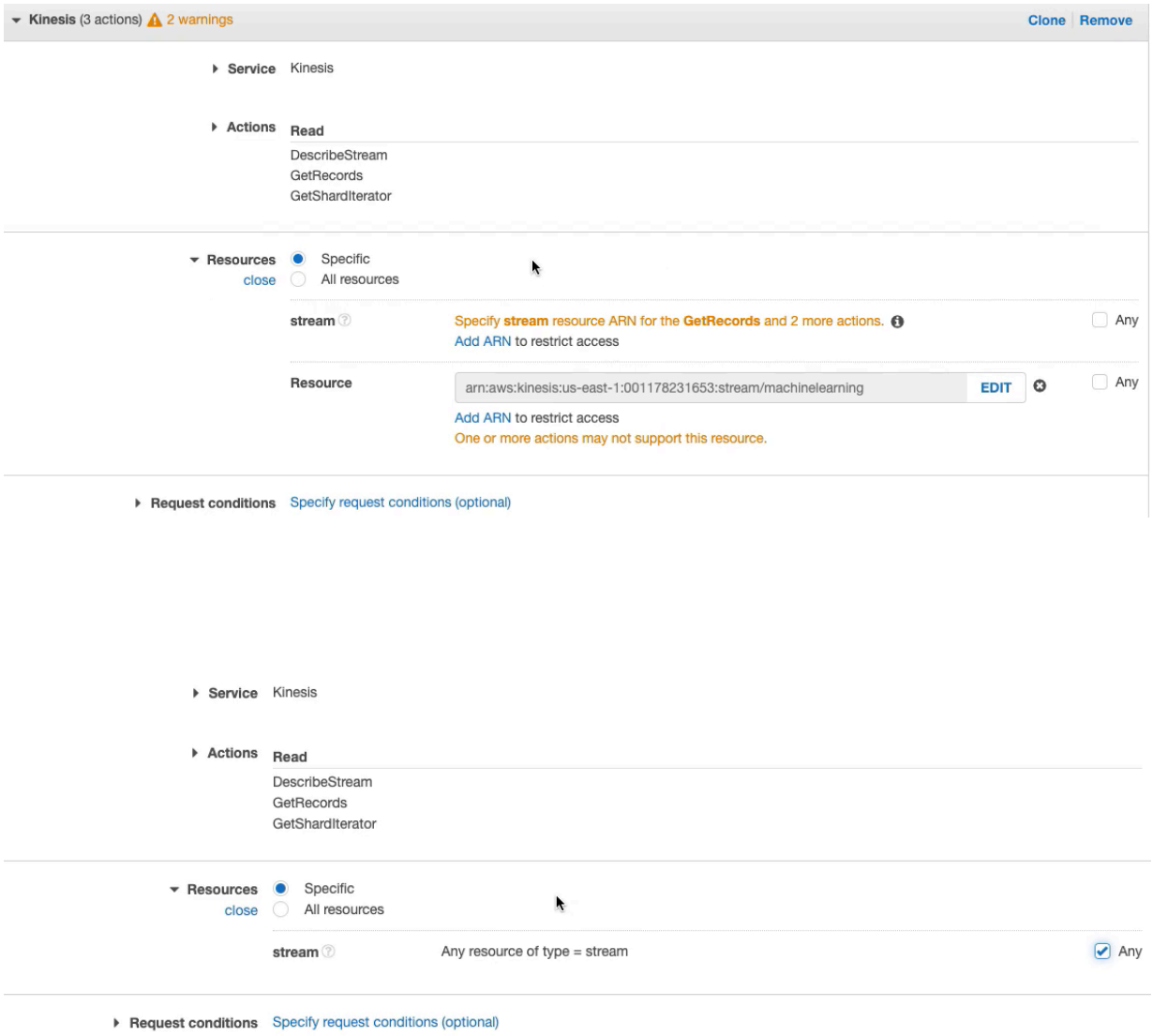
Edit the policy



We can see 2 warnings for Kinesis

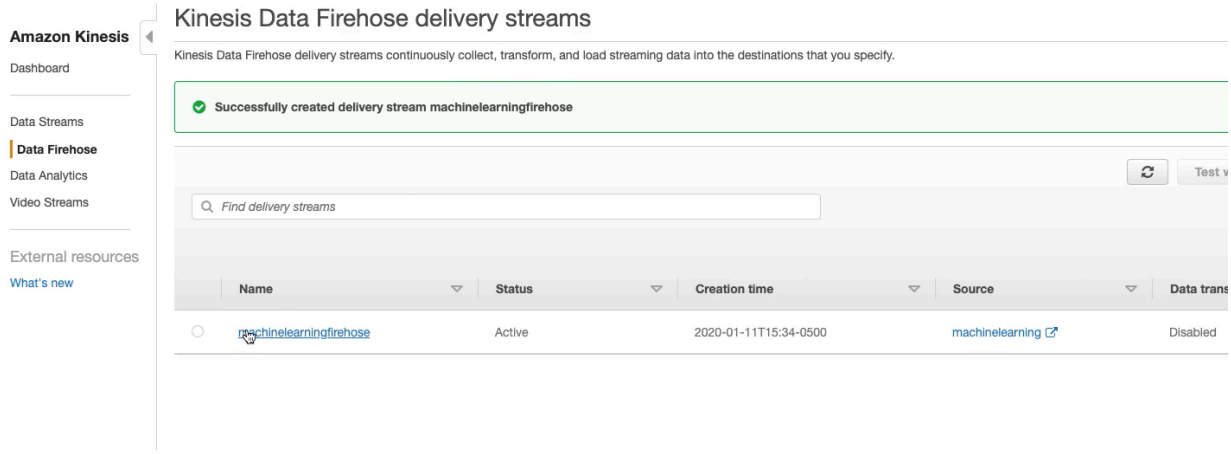


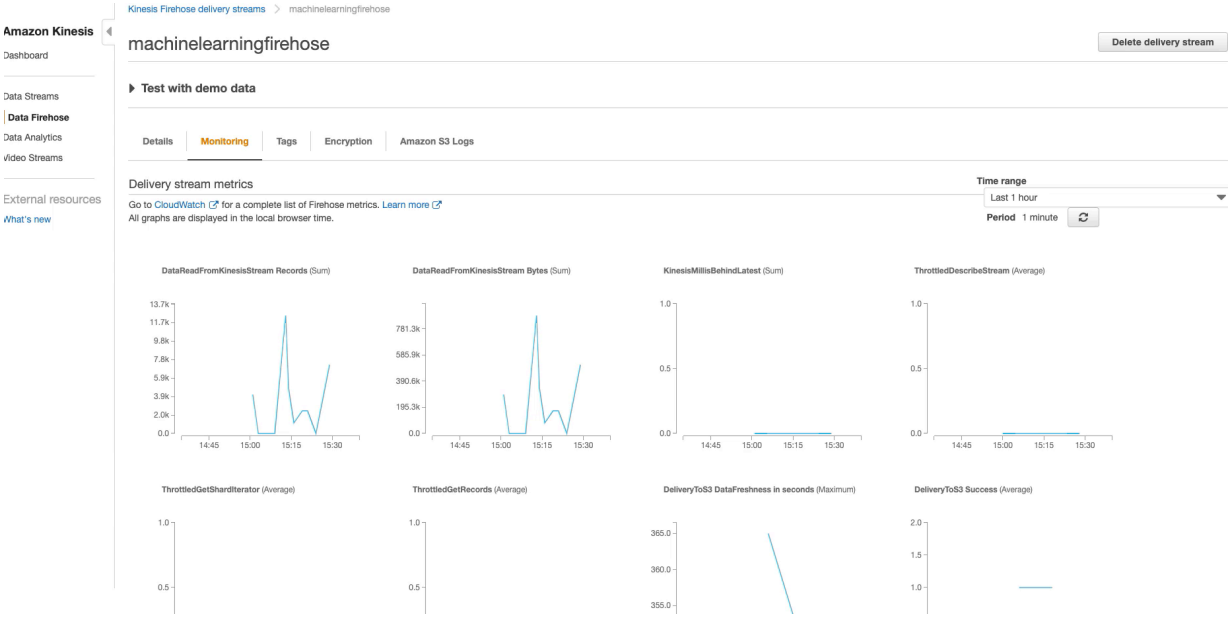
We need to fix the Kinesis Warnings:

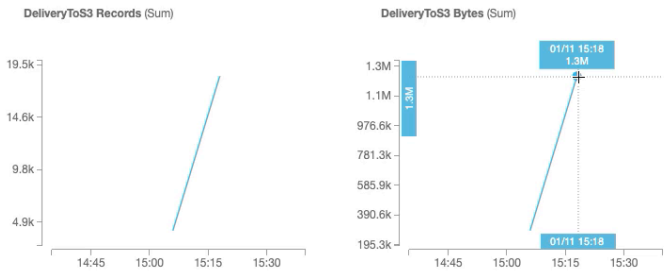


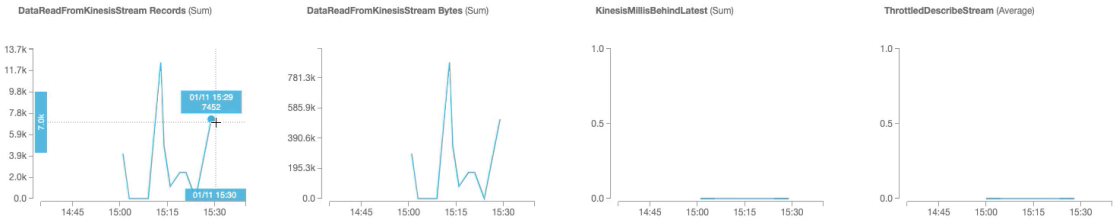
Now any resource of type stream can use this

Now Firehose is set up



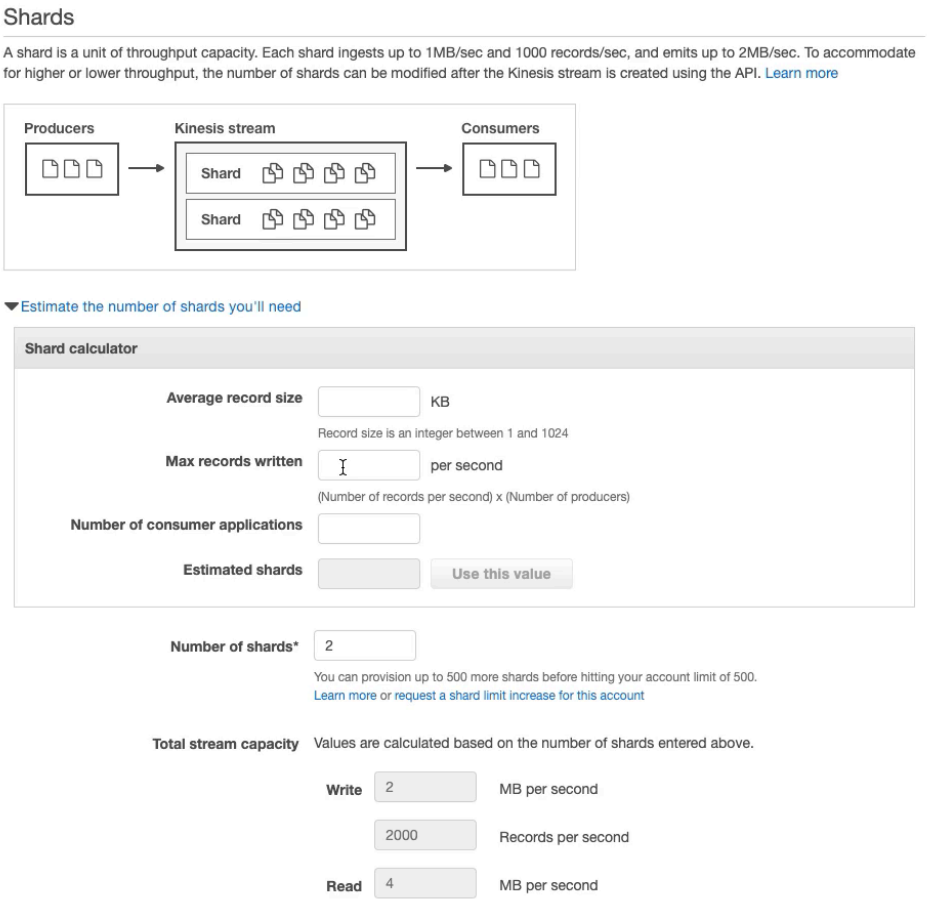




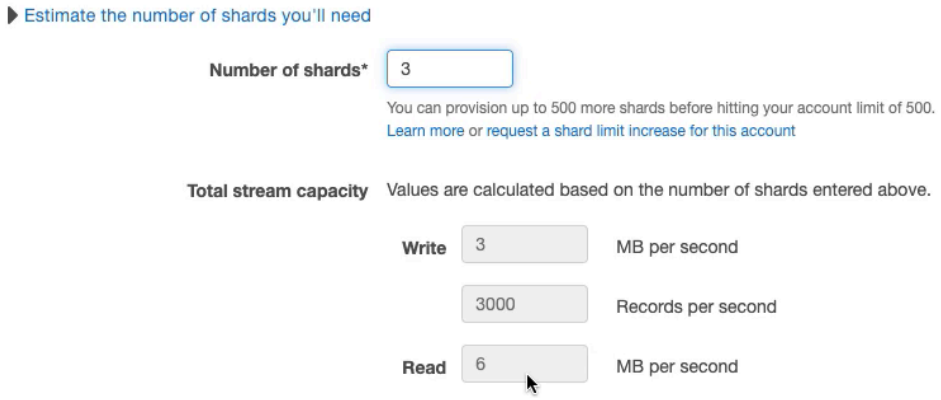


Things to remember for the exam with Kinesis Data Stream and Firehose:

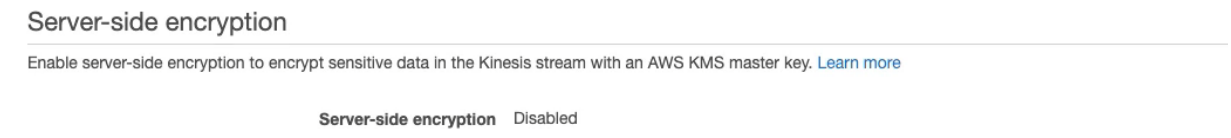
We can estimate record shards



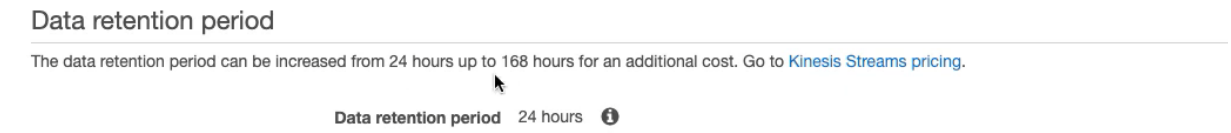
Or we can set a particular # of shards (1000 records/shard or 1Mb/shard)



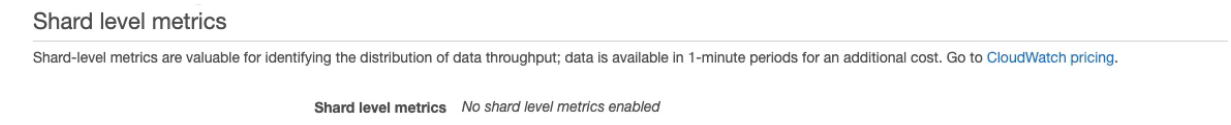
We can encrypt the data



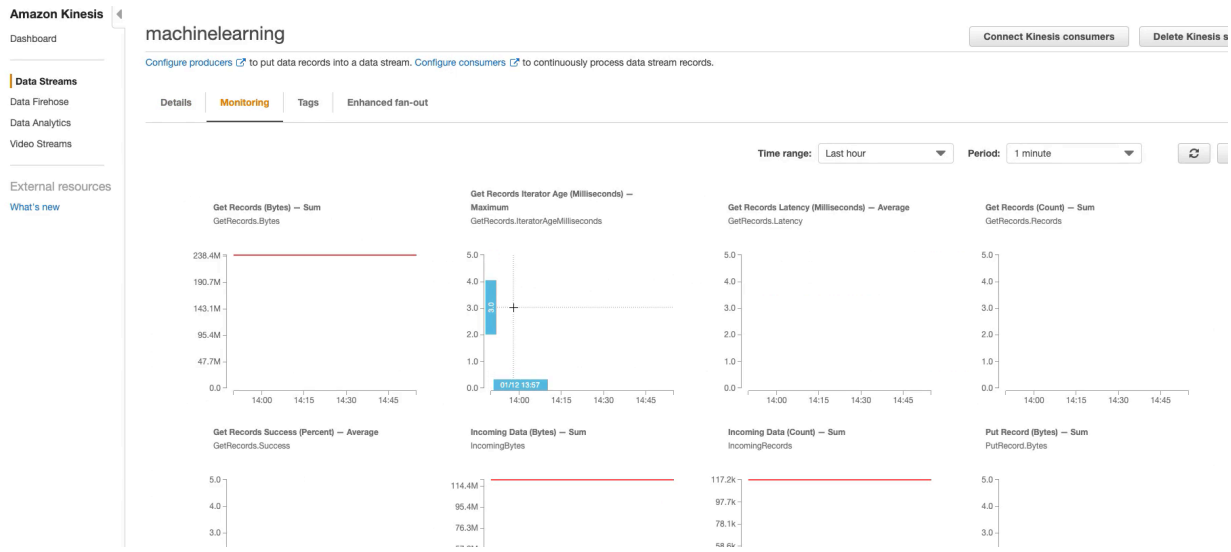
Data Retention period is 24 hours by default but can be extended to 168 hours (7 days)



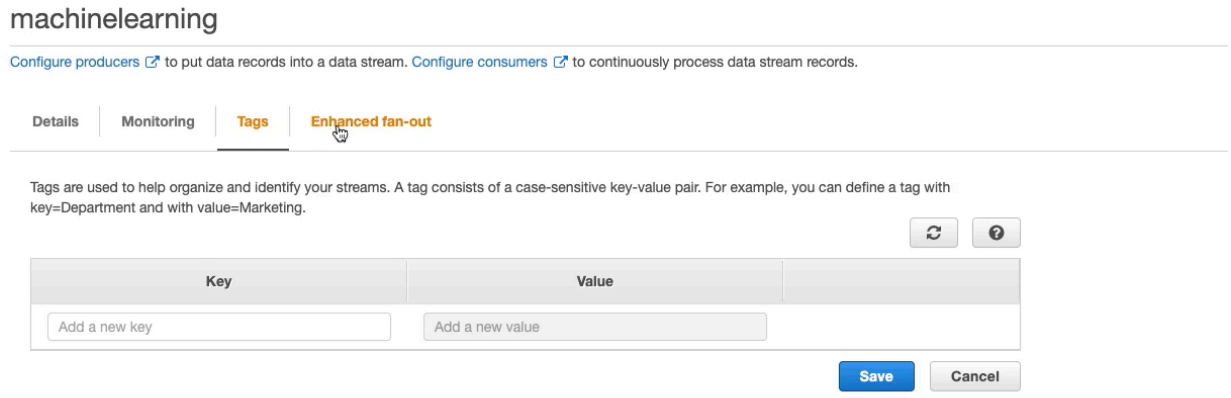
We can retrieve shard level metrics in cloudwatch



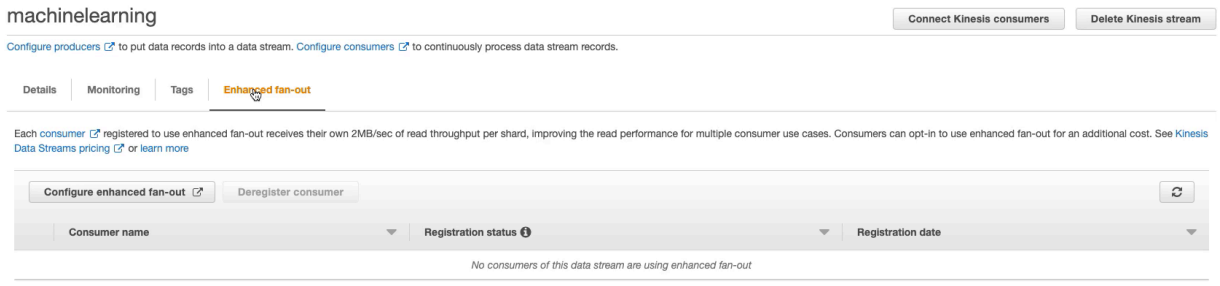
We can monitor our stream



We can tag out stream

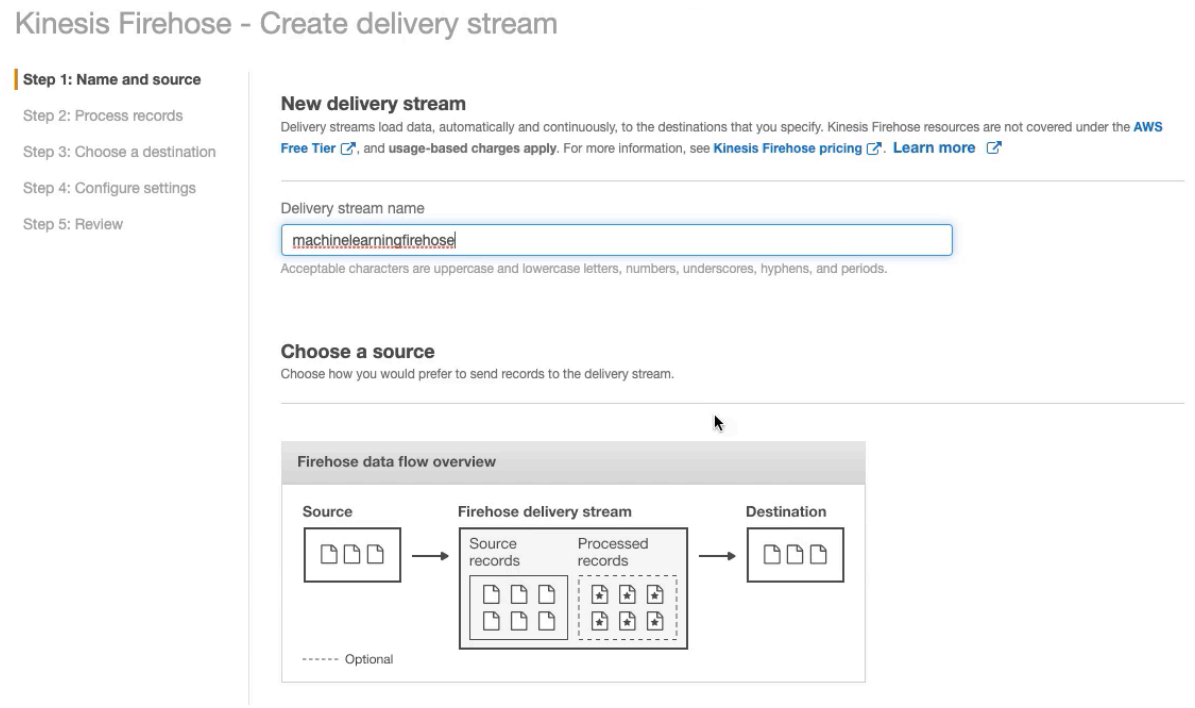


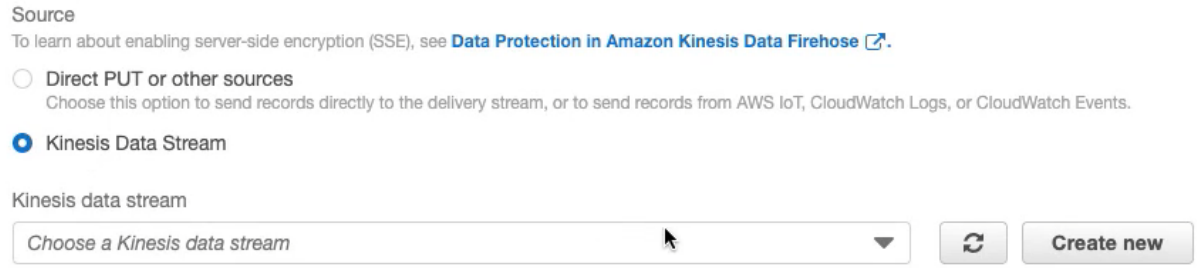
And we can use the advanced option of Fan out



Advanced FAN OUT:
From: 2Mb/s for all consumers by default
To: 2Mb/s for each consumer

Kinesis Firehose interface and tips





We can Transform the records by invoking a Lambda
And/or convert a record format using an AWS glue table for schema

Kinesis Firehose - Create delivery stream

Step 1: Name and source

Step 2: Process records

Step 3: Choose a destination

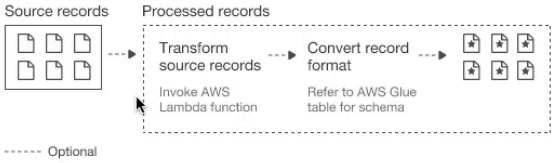
Step 4: Configure settings

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**

Data 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

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

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

AWS Glue table

Choose a table

AWS Glue table version

Choose a version

Finally we can write to S3:

Kinesis Firehose - Create delivery stream

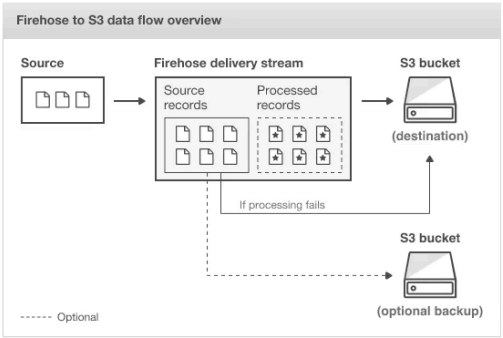
Step 1: Name and source
Step 2: Process records
Step 3: Choose a destination
Step 4: Configure settings
Step 5: Review

Select a destination

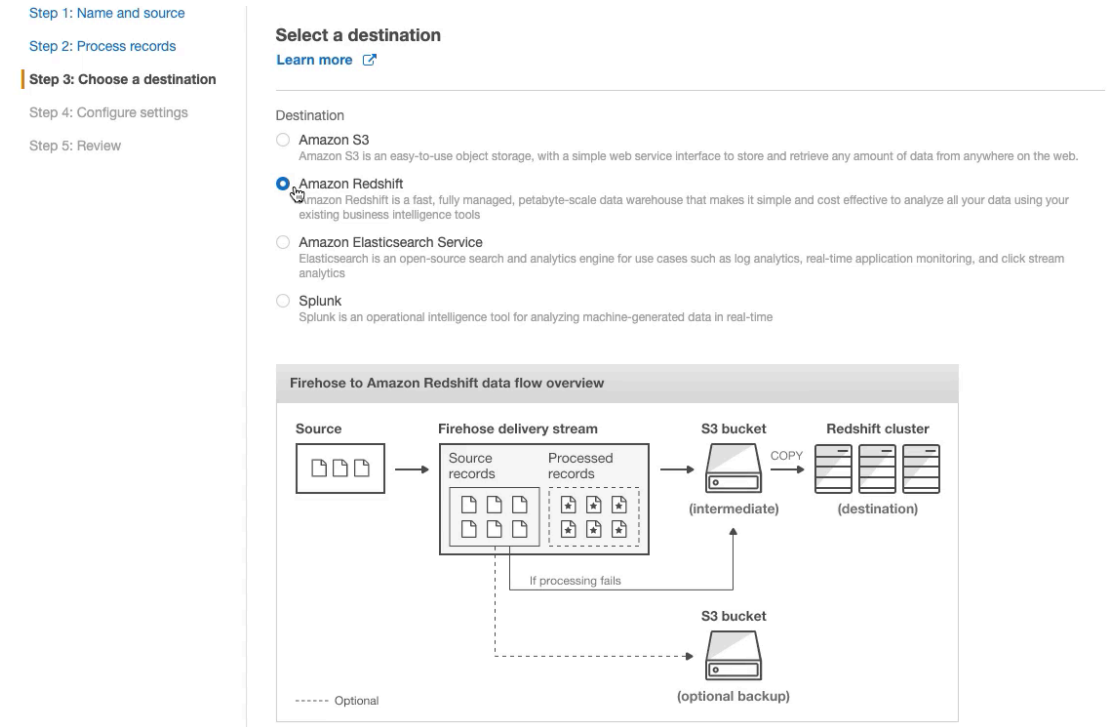
Learn more

Destination

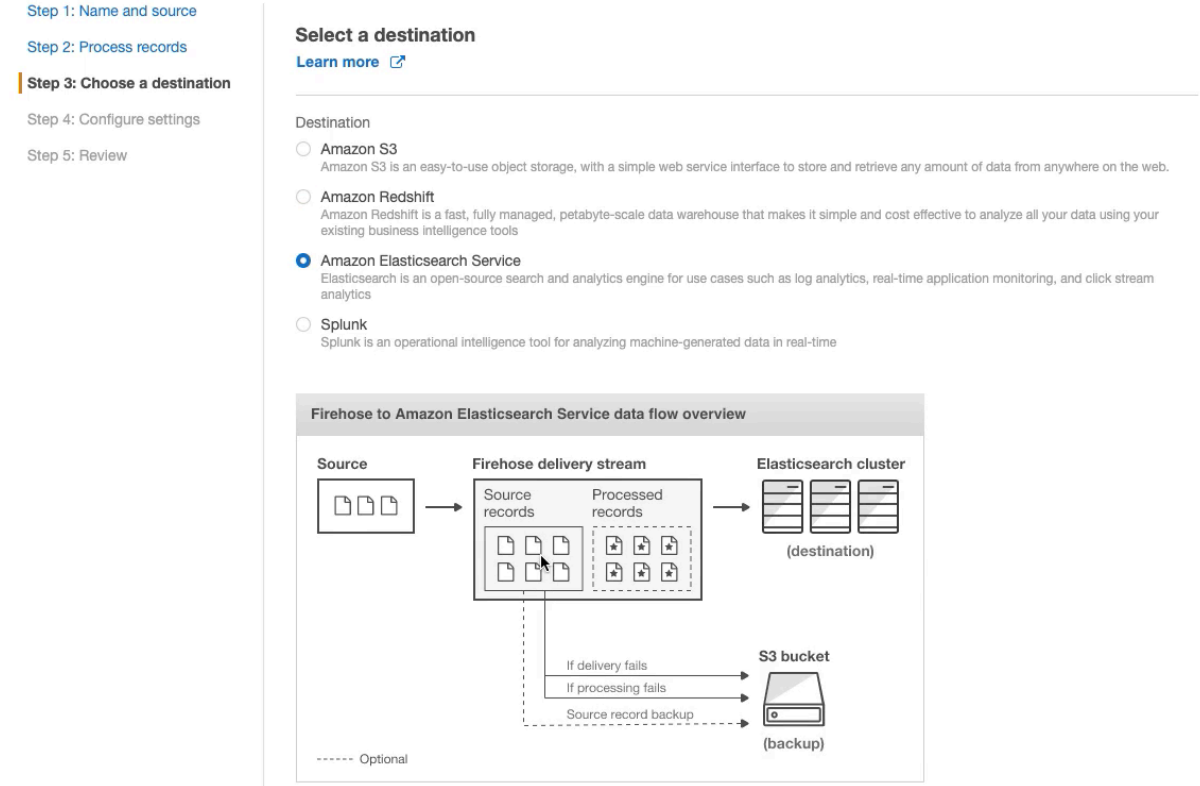
- Amazon S3
  Amazon S3 is an easy-to-use object storage, with a simple web service interface to store and retrieve any amount of data from anywhere on the web.
- Amazon Redshift
  Amazon Redshift is a fast, fully managed, petabyte-scale data warehouse that makes it simple and cost effective to analyze 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



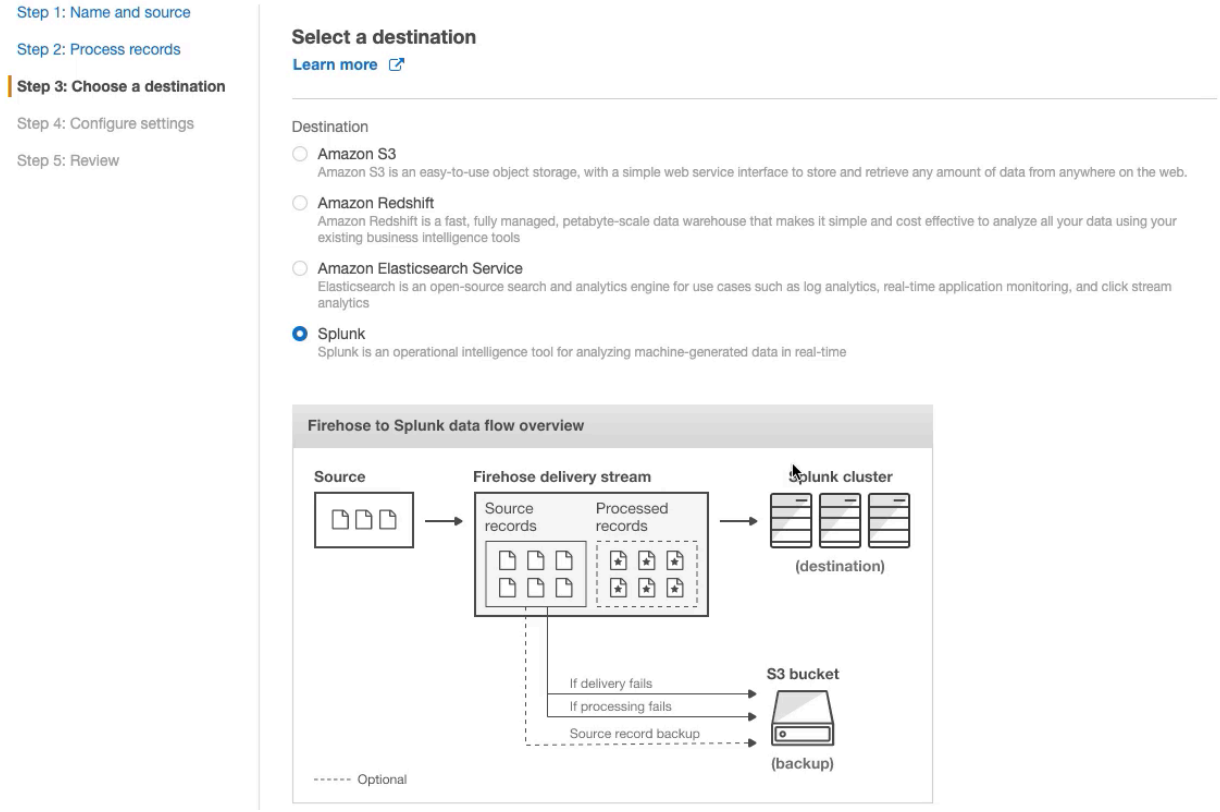
Or Redshift



Or Elastic Search



Or Splunk



With both labs, we have built the below pipeline