

# 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



Amazon  
Kinesis Data  
Streams



Amazon  
Kinesis Video  
Streams



Amazon  
Kinesis Data  
Firehose

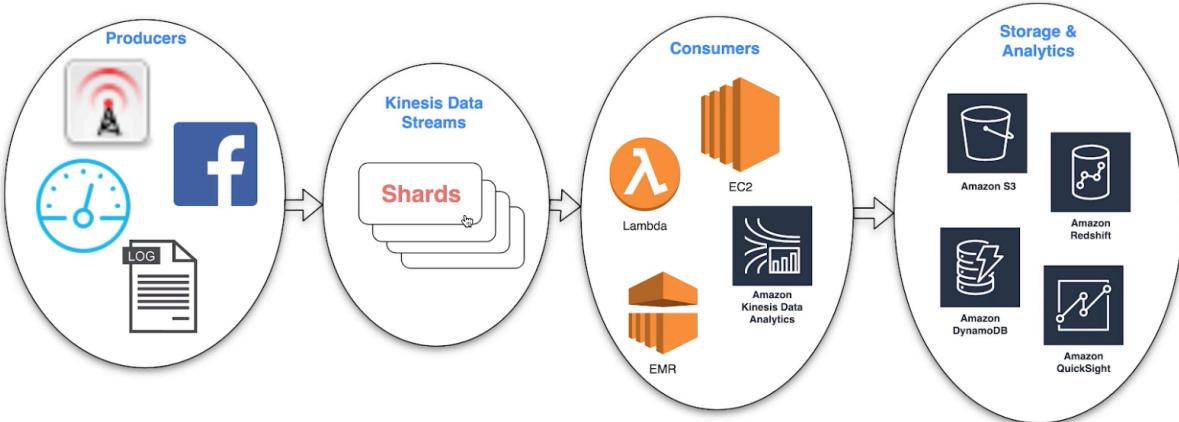


Amazon  
Kinesis Data  
Analytics



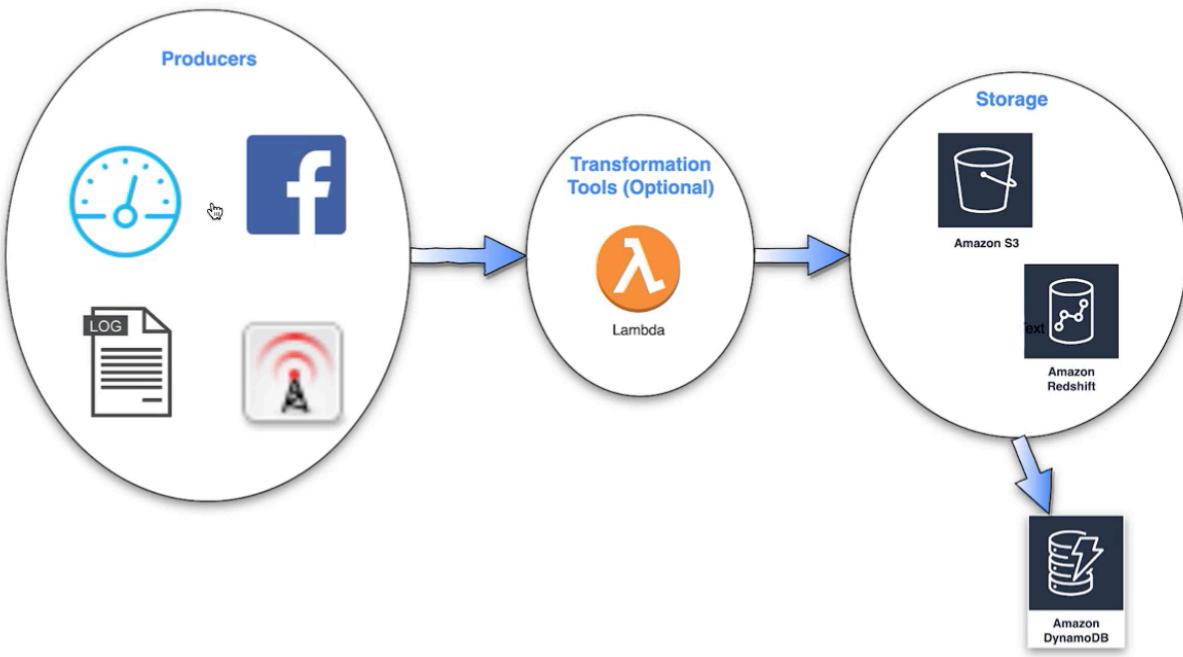
### Kinesis Data Streams

- 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



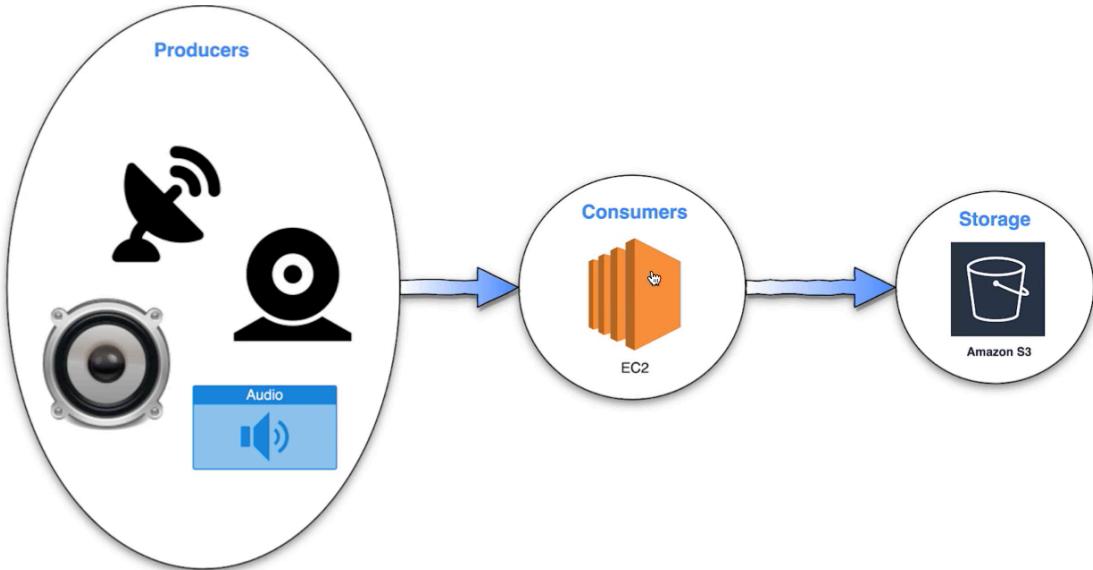
## Kinesis Data Firehose

- 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



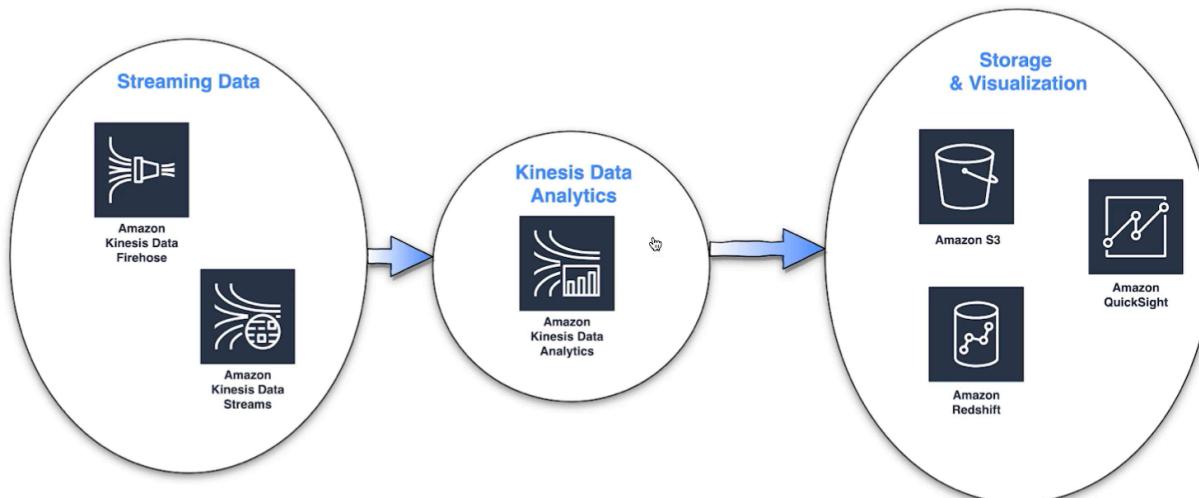
## Kinesis Video Streams

- 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



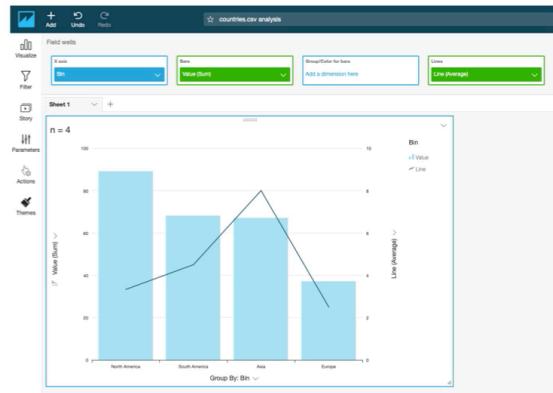
## Kinesis Data Analytics

- ❑ 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

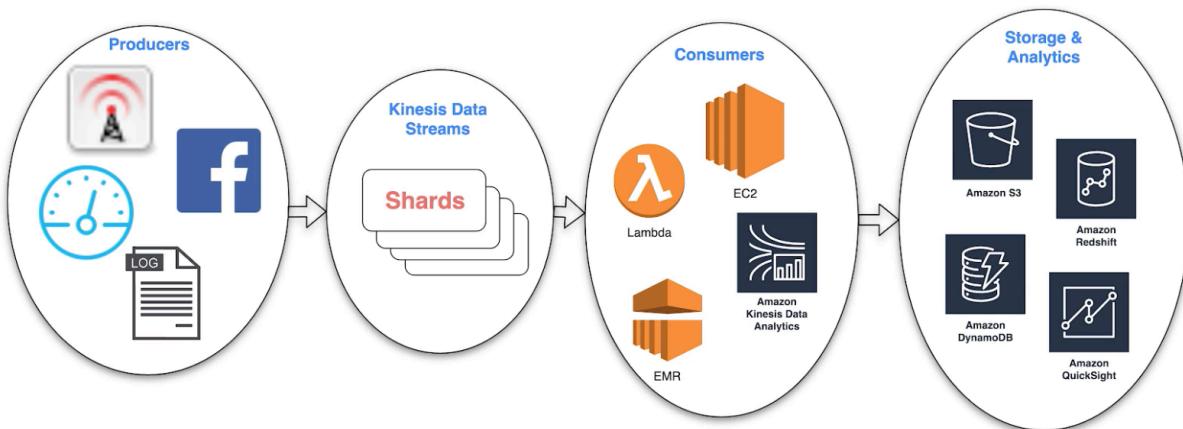
- ❑ Technique using static and interactive visuals to represent large amounts of data.
- ❑ Visualizes patterns, trends and correlations that may be difficult to discern
- ❑ Data visualization helps monetize data as a product



## 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                    Financial transactions
  - Database event streams                IoT events
  - Social media feeds                       Application logs
- Enables real-time analytics
  - Real-time dashboards                    Dynamic pricing
  - Real-time anomaly detection           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

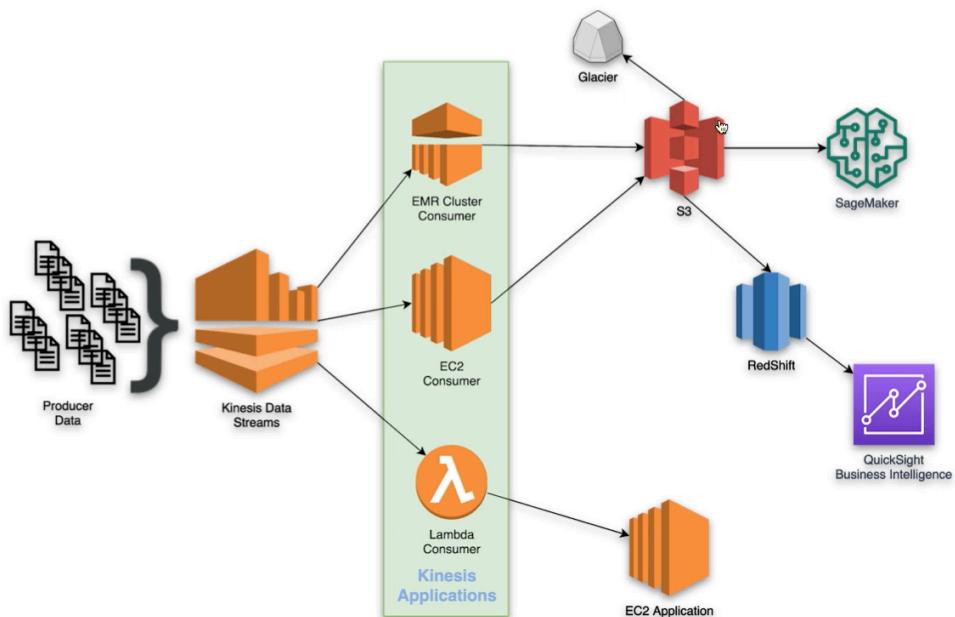
# AWS Machine Learning Kinesis Data Streams

## 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

The screenshot shows a blog post from the AWS Big Data Blog. The title is "Test Your Streaming Data Solution with the New Amazon Kinesis Data Generator". The post is by Allan MacInnis and was published on May 10, 2017. It discusses the challenges of testing streaming data solutions and introduces the Amazon Kinesis Data Generator (KDG) to simplify the process. The post includes several related links at the bottom.

**Resources**

- Amazon Athena
- Amazon EMR
- AWS Glue
- Amazon DynamoDB
- Amazon Kinesis
- Amazon QuickSight
- Amazon Redshift

**Follow**

- Twitter
- Facebook

Create the stack and log in to KDG

The KDG makes it simple to send test data to your Amazon Kinesis stream or Amazon Kinesis Firehose delivery stream. Sign in to get started. If you haven't configured an Amazon Cognito user, choose [Help](#).

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"]  
    )}}"  
}
```

Amazon Kinesis Data Generator

Configure Help Log Out

Region: us-east-1

Stream/delivery stream: No destinations found in this region

Records per second: Constant 100

Compress Records:

Record template: Template 2

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

Send data Test template

## 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" }
```

Lock to real time

Seconds of data per tick

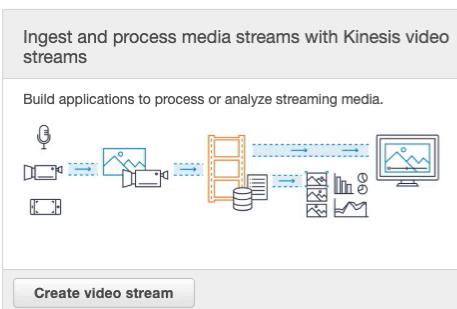
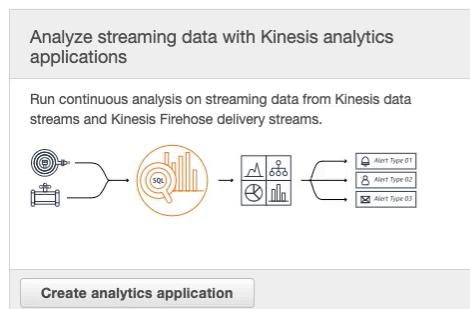
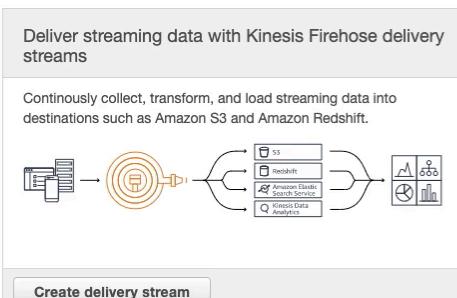
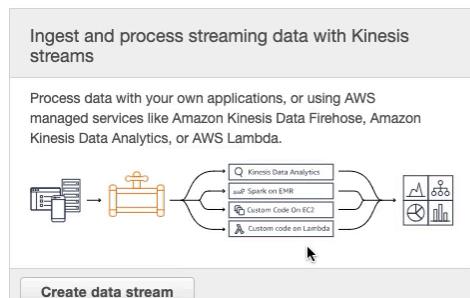
Wait between ticks

Hour	Sun	Mon	Tue	Wed	Thu	Fri	Sat
0	Mu: <input type="text" value="100"/> Sigma: <input type="text" value="10"/>						
1	Mu: <input type="text" value="100"/> Sigma: <input type="text" value="10"/>						
2	Mu: <input type="text" value="100"/> Sigma: <input type="text" value="10"/>						
3	Mu: <input type="text" value="100"/> Sigma: <input type="text" value="10"/>						

## Now create a Data stream

### Get started with Amazon Kinesis

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



With 2 shards

**Create Kinesis stream**

Kinesis stream name\* machinelearning  
Acceptable characters are uppercase and lowercase letters, numbers, underscores, hyphens, and periods.

**Shards**  
A shard is a unit of throughput capacity. Each shard ingests up to 1MB/sec and 1000 records/sec, and emits up to 2MB/sec. To accommodate for higher or lower throughput, the number of shards can be modified after the Kinesis stream is created using the API. [Learn more](#)

▶ Estimate the number of shards you'll need

Number of shards\* 2  
You can provision up to 500 more shards before hitting your account limit of 500.  
[Learn more](#) or request a shard limit increase for this account

Total stream capacity Values are calculated based on the number of shards entered above.

Write	2	MB per second
	2000	Records per second
Read	4	MB per second

\* Required      Cancel      **Create Kinesis stream**

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

Amazon Kinesis Data Generator

Configure Help Log Out

Region us-east-1

Stream/delivery stream machinelearning

Records per second Constant 100

Sending Data to Kinesis

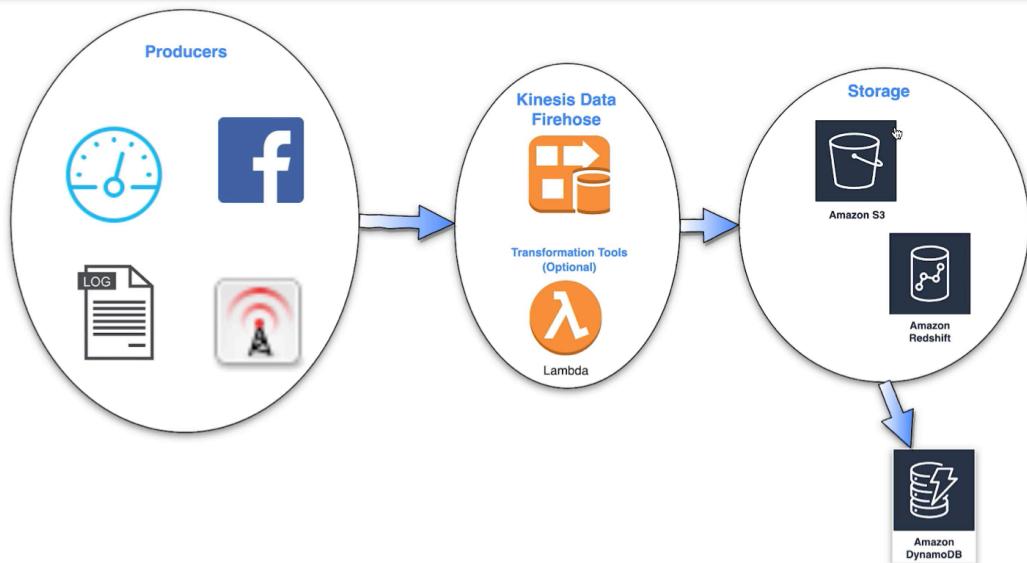
500 records sent to Kinesis.

Stop Sending Data to Kinesis

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

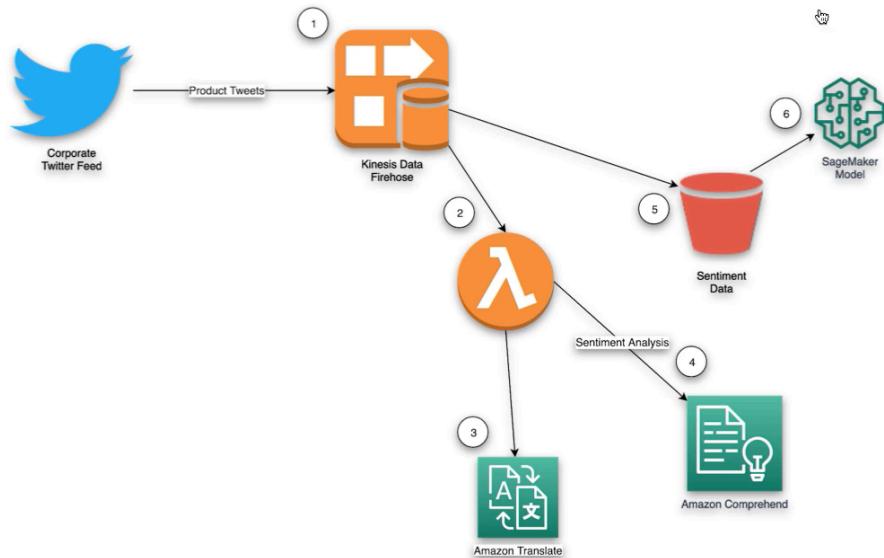
Kinesis Firehose - Architecture example

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:

**Kinesis streams**

Kinesis data streams continuously capture and temporarily store real-time data. [Configure producers](#) to put data records into a data stream. [Configure consumers](#) to continuously process data.

Total shards in use: 2 Total shards remaining: 498 ⓘ

Create Kinesis stream	Connect Kinesis consumers	Actions
<input type="text"/> Filter Kinesis streams		
<input checked="" type="checkbox"/> Kinesis stream name	<input type="button"/>	Number of shards
<input checked="" type="checkbox"/> machinelearning	2	Status Active
		Consumers using enhanced consumer API 0

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

**Connect Kinesis consumers**

Connect your Kinesis data stream to other Kinesis consumers

Deliver records with a Kinesis Data Firehose delivery stream

Continuously collect, transform, and load streaming data to destinations such as Amazon S3 and Amazon Redshift.  
[Learn more](#)

Connect to delivery stream

Analyze records with a Kinesis Data Analytics application

Run continuous analysis on data stream records. [Learn more](#)

Connect to analytics application

Close

## 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](#)

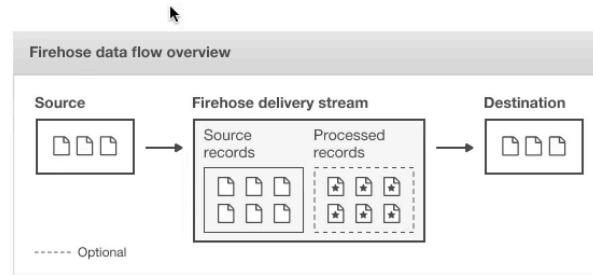
Delivery stream name

machinelearningfirehose

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.



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

### Source

To learn about enabling server-side encryption (SSE), see [Data Protection in Amazon Kinesis Data Firehose](#).

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

**Info** You chose **machinelearning** in Kinesis Data Streams to be the source for this delivery stream. To use a different Kinesis Data Stream source, update your choice below.

Kinesis data stream

machinelearning



Create new

[View machinelearning in Kinesis data streams](#)



**Enable server-side encryption (SSE)**

To enable SSE for the delivery stream, view the data stream selected above, and enable SSE on it. If you choose Direct PUT or other data sources for your delivery stream, you can enable SSE on the delivery stream directly.

[Cancel](#)

[Next](#)

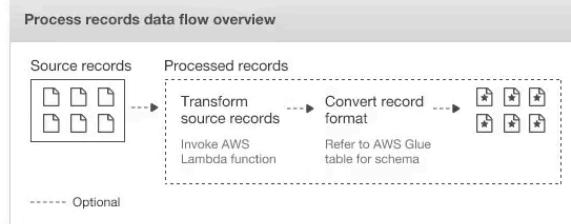
We are not going to transform the data with a Lambda, nor convert it so we will keep both options disabled

## 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](#)

##### 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 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](#).

[Cancel](#)

[Previous](#)

[Next](#)

## Use S3 as final 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

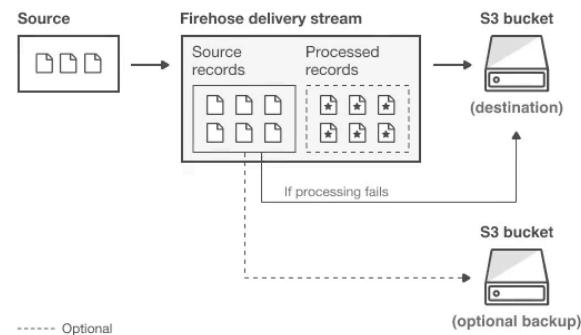
Elasticsearch is an open-source search and analytics engine for use cases such as log analytics, real-time application monitoring, and click stream analytics

##### Splunk

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



### Firehose to S3 data flow overview



### 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



[Create new](#)

[View ml.vbloise3 in S3 console](#)

S3 prefix

In the "Configure Settings", we keep the buffer size default

## 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

5 MB

Enter a buffer size between 1-128 MB



Buffer interval

300 seconds

Enter a buffer interval between 60-900 seconds

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

#### S3 compression and encryption

Firehose can compress records before delivering them to your S3 bucket. Compressed records can also be encrypted in the S3 bucket using a KMS master key. [Learn more](#)

S3 compression

- Disabled
- GZIP
- Snappy
- Zip

S3 encryption

- Disabled
- Enabled

We keep Error logging to CloudWatch enabled (by default)

And we create an IAM role to allow Firehose to retrieve records from the stream and write records to S3

## Error logging

Firehose can log record delivery errors to CloudWatch Logs. If enabled, a CloudWatch log group and corresponding log streams are created on your behalf.  
[Learn more](#)

### Error logging

- Disabled
- Enabled

## Tags - optional

You can add tags to organize your AWS resources, track costs, and control access. [Learn more](#)

Key	Value - optional	
<input type="text" value="Enter key"/>	<input type="text" value="Enter value"/>	<a href="#">Remove tag</a>

[Add tag](#)

You can add 49 more tag(s)

## Permissions

### IAM role

[Create new or choose](#)

### Amazon Kinesis Firehose is requesting permission to use resources in your account

Click Allow to give Amazon Kinesis Firehose Read and Write access to resources in your account.

[▼ 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](#)

Extending policy documents:

▼ 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

 Hide Policy Document

Edit

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Sid": "",  
      "Effect": "Allow",  
      "Action": [  
        "glue:GetTable",  
        "glue:GetTableVersion",  
        "glue:GetTableVersions"  
      ]  
    }  
  ]  
}
```

```
"Resource": [  
  "arn:aws:s3:::ml.vb|pise3",  
  "arn:aws:s3:::ml.vb|pise3/*",  
  "arn:aws:s3:::%FIREHOSE_BUCKET_NAME%",  
  "arn:aws:s3:::%FIREHOSE_BUCKET_NAME%/*"  
]
```

Click on Allow

**Cancel** **Allow** 

Now checking the role:

Identity and Access Management (IAM)

Role ARN: arn:aws:iam::001178231653:role/firehose\_delivery\_role\_2

Role description: Edit

Instance Profile ARNs: /

Path: /

Creation time: 2020-01-11 15:33 EST

Last activity: Not accessed in the tracking period

Maximum CLI/API session duration: 1 hour Edit

Permissions Trust relationships Tags Access Advisor Revoke sessions

Permissions policies (1 policy applied)

Attach policies

Policy name: oneClick\_firehose\_delivery\_role\_1578774758791

Policy type: Inline policy

## Edit the policy

Policy name: oneClick\_firehose\_delivery\_role\_1578774758791

Policy summary { } JSON Edit policy

```
1 {  
2   "Version": "2012-10-17",  
3   "Statement": [  
4     {  
5       "Sid": "",  
6       "Effect": "Allow",  
7       "Action": [  
8         "glue:GetTable",  
9         "glue:GetTableVersion",  
10        "glue:GetTableVersions"  
11      ],  
12      "Resource": "*"  
13    }  
14  ]  
15}
```

We can see 2 warnings for Kinesis

## Edit oneClick\_firehose\_delivery\_role\_1578774758791

1 2

A policy defines the AWS permissions that you can assign to a user, group, or role. You can create and edit a policy in the visual editor and using JSON. [Learn more](#)

Visual editor    [JSON](#)    [Import managed policy](#)

[Expand all](#) | [Collapse all](#)

▶ Glue (3 actions)	<a href="#">Clone</a> <a href="#">Remove</a>
▶ S3 (6 actions)	<a href="#">Clone</a> <a href="#">Remove</a>
▶ Lambda (2 actions) <span style="color: orange;">⚠ 2 warnings</span>	<a href="#">Clone</a> <a href="#">Remove</a>
▶ CloudWatch Logs (1 action) <span style="color: orange;">⚠ 1 warning</span>	<a href="#">Clone</a> <a href="#">Remove</a>
▶ Kinesis (3 actions) <span style="color: orange;">⚠ 2 warnings</span>	<a href="#">Clone</a> <a href="#">Remove</a>
▶ KMS (1 action) <span style="color: orange;">⚠ 1 warning</span>	<a href="#">Clone</a> <a href="#">Remove</a>

We need to fix the Kinesis Warnings:

▼ Kinesis (3 actions) ⚠ 2 warnings    [Clone](#) | [Remove](#)

▶ Service Kinesis

▶ Actions Read

DescribeStream  
GetRecords  
GetShardIterator

▼ Resources  Specific [close](#)  All resources

stream Specify stream resource ARN for the GetRecords and 2 more actions. [i](#)  Any  
[Add ARN](#) to restrict access

Resource  [EDIT](#) [✖](#)  Any  
[Add ARN](#) to restrict access  
One or more actions may not support this resource.

▶ Request conditions [Specify request conditions \(optional\)](#)

▶ Service Kinesis

▶ Actions Read

DescribeStream  
GetRecords  
GetShardIterator

▼ Resources  Specific [close](#)  All resources

stream Any resource of type = stream  Any

▶ Request conditions [Specify request conditions \(optional\)](#)

Now any resource of type stream can use this

Now Firehose is set up

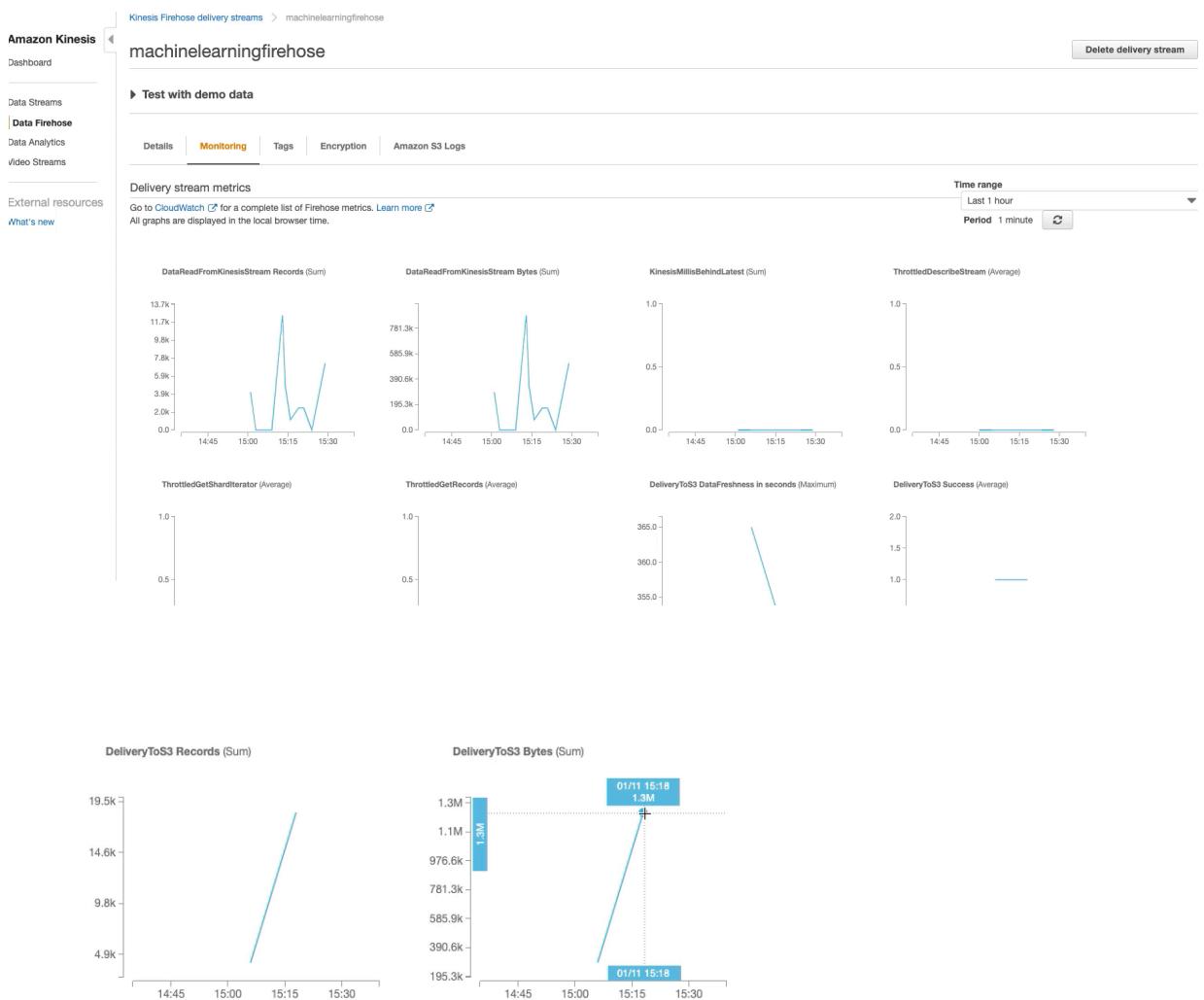
Kinesis Data Firehose delivery streams

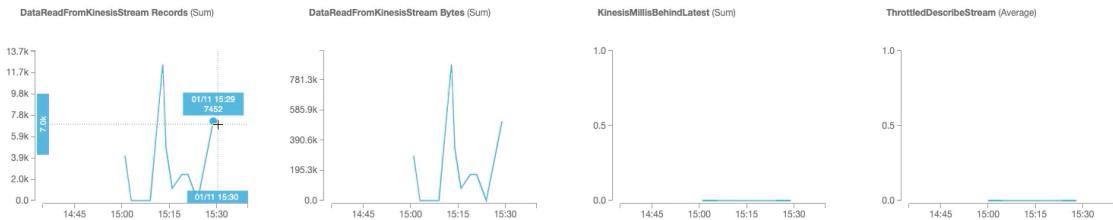
Kinesis Data Firehose delivery streams continuously collect, transform, and load streaming data into the destinations that you specify.

Successfully created delivery stream machinelearningfirehose

Find delivery streams

Name	Status	Creation time	Source	Data transformation
machinelearningfirehose	Active	2020-01-11T15:34:0500	machinelearning	Disabled



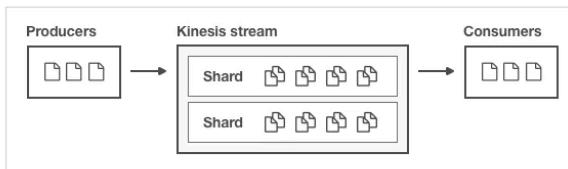


## Things to remember for the exam with Kinesis Data Stream and Firehose:

We can estimate record shards

### Shards

A shard is a unit of throughput capacity. Each shard ingests up to 1MB/sec and 1000 records/sec, and emits up to 2MB/sec. To accommodate for higher or lower throughput, the number of shards can be modified after the Kinesis stream is created using the API. [Learn more](#)



▼ Estimate the number of shards you'll need

**Shard calculator**

Average record size	<input type="text"/> KB
Record size is an integer between 1 and 1024	
Max records written	<input type="text"/> per second
(Number of records per second) x (Number of producers)	
Number of consumer applications	<input type="text"/>
Estimated shards	<input type="button" value="Use this value"/>

**Number of shards\***

You can provision up to 500 more shards before hitting your account limit of 500.  
[Learn more](#) or [request a shard limit increase for this account](#)

**Total stream capacity** Values are calculated based on the number of shards entered above.

**Write**  MB per second

**2000** Records per second

**Read**  MB per second

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

► Estimate the number of shards you'll need

Number of shards\*  You can provision up to 500 more shards before hitting your account limit of 500.  
[Learn more](#) or [request a shard limit increase for this account](#)

Total stream capacity Values are calculated based on the number of shards entered above.

Write  MB per second  
3000 Records per second  
Read  MB per second

We can encrypt the data

#### Server-side encryption

Enable server-side encryption to encrypt sensitive data in the Kinesis stream with an AWS KMS master key. [Learn more](#)

Server-side encryption Disabled

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

#### Data retention period

The data retention period can be increased from 24 hours up to 168 hours for an additional cost. Go to [Kinesis Streams pricing](#).

Data retention period 24 hours 

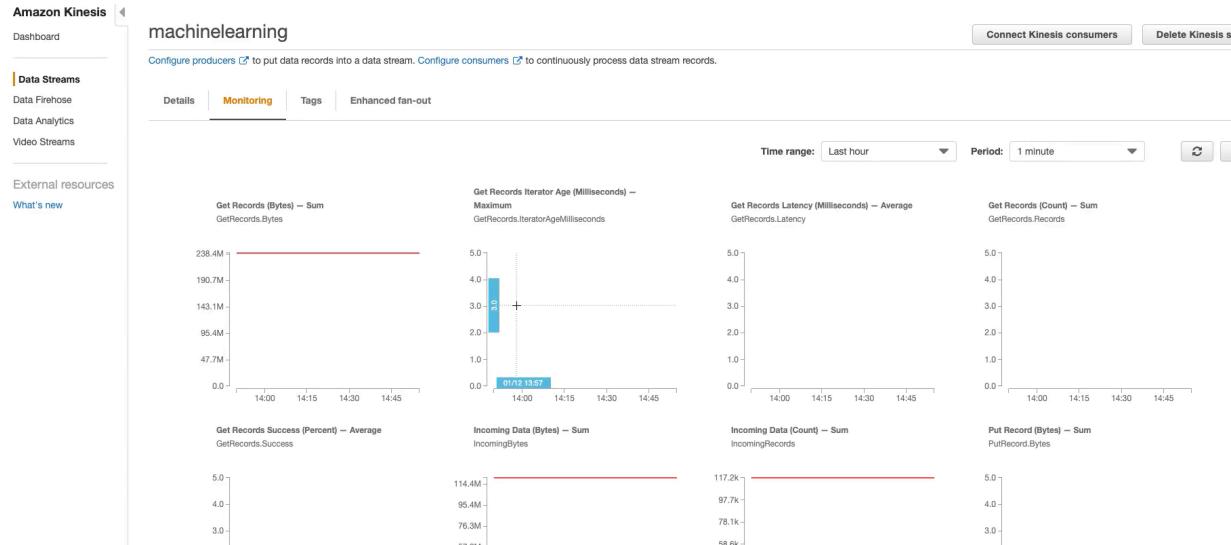
We can retrieve shard level metrics in cloudwatch

#### Shard level metrics

Shard-level metrics are valuable for identifying the distribution of data throughput; data is available in 1-minute periods for an additional cost. Go to [CloudWatch pricing](#).

Shard level metrics No shard level metrics enabled

We can monitor our stream



We can tag out stream

## machinelearning

Configure producers to put data records into a data stream. Configure consumers to continuously process data stream records.

Details Monitoring Tags Enhanced fan-out

Tags are used to help organize and identify your streams. A tag consists of a case-sensitive key-value pair. For example, you can define a tag with key=Department and with value=Marketing.

Key	Value
Add a new key	Add a new value

**Save** **Cancel**

And we can use the advanced option of Fan out

## machinelearning

Connect Kinesis consumers Delete Kinesis stream

Configure producers to put data records into a data stream. Configure consumers to continuously process data stream records.

Details Monitoring Tags Enhanced fan-out

Each consumer registered to use enhanced fan-out receives their own 2MB/sec of read throughput per shard, improving the read performance for multiple consumer use cases. Consumers can opt-in to use enhanced fan-out for an additional cost. See Kinesis Data Streams pricing or learn more

Consumer name	Registration status	Registration date
No consumers of this data stream are using enhanced fan-out		

Advanced FAN OUT:

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

## Kinesis Firehose interface and tips

### 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](#)

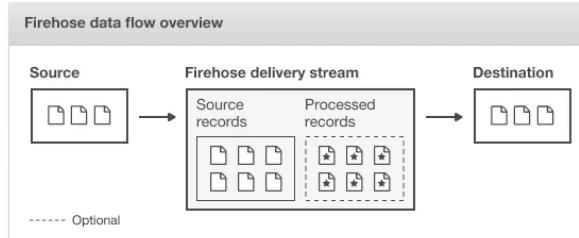
Delivery stream name

machinelearningfirehose

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.



#### Source

To learn about enabling server-side encryption (SSE), see [Data Protection in Amazon Kinesis Data Firehose](#).

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

#### Kinesis data stream

Choose a Kinesis data stream



Create new

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

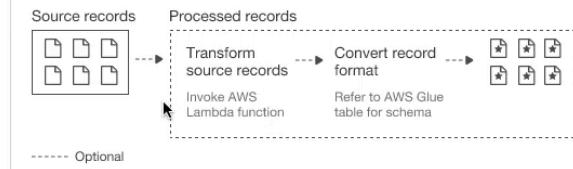
Step 4: Configure settings

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

**Firehose to S3 data flow overview**

```

graph LR
    Source[Source] --> Firehose[Firehose delivery stream]
    subgraph Firehose [Firehose delivery stream]
        SR[Source records]
        PR[Processed records]
        SR --> PR
    end
    PR --> S3Dest[S3 bucket (destination)]
    S3Dest --> S3Backup[S3 bucket (optional backup)]
    S3Backup -.-> Firehose
    PR -. "If processing fails" .-> S3Backup
  
```

The diagram illustrates the data flow for a Firehose delivery stream. It starts with a 'Source' icon (containing three document icons) pointing to a central 'Firehose delivery stream' box. This box contains two sections: 'Source records' (containing four document icons) and 'Processed records' (containing eight document icons). An arrow points from 'Source records' to 'Processed records'. From 'Processed records', an arrow points to an 'S3 bucket' icon labeled '(destination)'. A dashed arrow labeled 'Optional' points from the bottom of the 'Processed records' section to another 'S3 bucket' icon labeled '(optional backup)'. A dashed arrow labeled 'If processing fails' points from the 'Processed records' section to the '(optional backup)' S3 bucket.

## Or Redshift

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

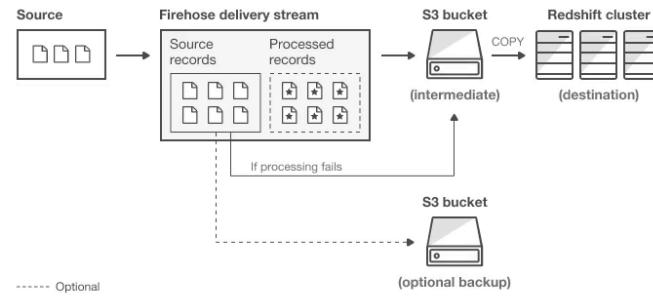
Amazon Elasticsearch Service

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

#### Firehose to Amazon Redshift data flow overview



## Or Elastic Search

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

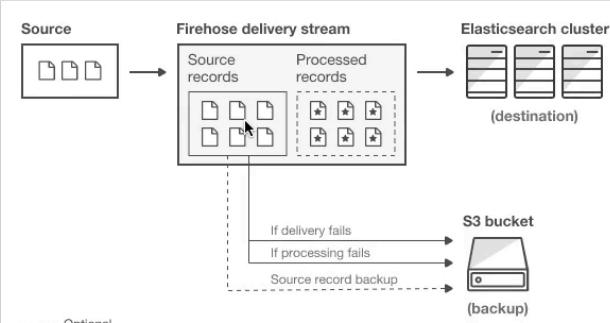
Amazon Elasticsearch Service

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

#### Firehose to Amazon Elasticsearch Service data flow overview



## Or Splunk

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

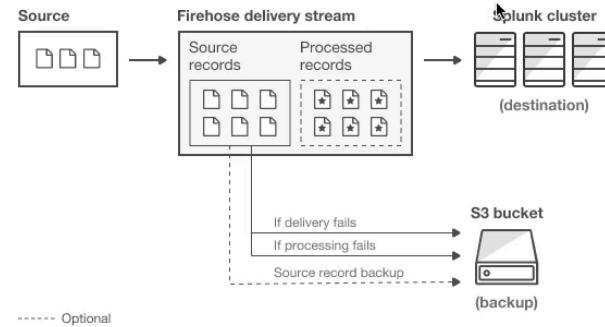
##### Amazon Elasticsearch Service

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

#### Firehose to Splunk data flow overview



With both labs, we have built the below pipeline

