

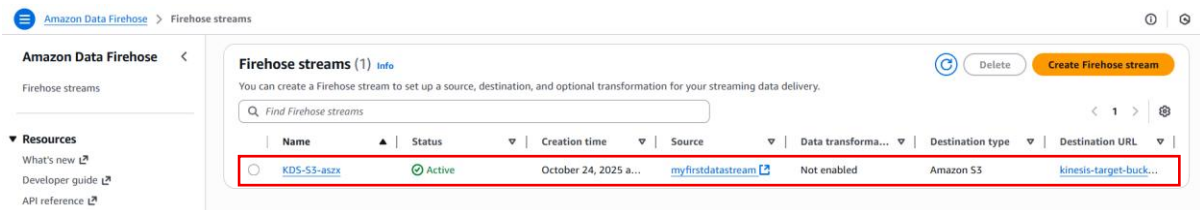
Data Firehose - Transformations with Lambda

To Begin with the Lab

Summary of the Lab

In this lab, you enable **data transformation in Amazon Kinesis Data Firehose** using **AWS Lambda**. You create a Lambda function from a blueprint to process and transform streaming data before it's delivered to S3. After connecting the Lambda function to Firehose, you send records to the Kinesis stream, verify transformed data in the S3 bucket, and confirm successful delivery.

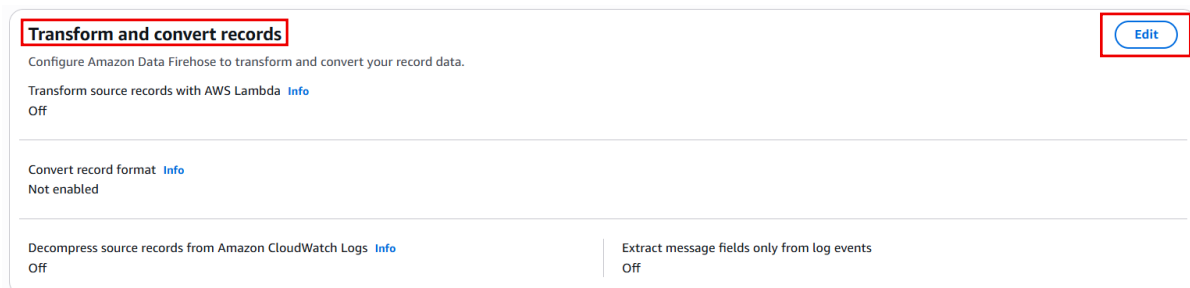
- **Prerequisites**
 - A **Kinesis Data Stream** already created (e.g., myfirstdatastream).
 - A **Kinesis Data Firehose** delivery stream connected to that Kinesis stream.
 - An **S3 bucket** configured as the Firehose destination.
- Go to the **AWS Management Console** → Search and open **Kinesis**.
- Choose your existing **Firehose stream**.



- Go to the **Configuration** tab of your Firehose stream.



- Scroll to **Transform and convert records**.
- Click **Edit**.



- Check the box **“Enable data transformation”**.

Edit transform and convert records

Transform and convert records - optional

Configure Amazon Data Firehose to transform and convert your record data.

Transform source records with AWS Lambda

To return records from AWS Lambda to Amazon Data Firehose after transformation, the Lambda function you invoke must be compliant with the required record transformation output model. Pricing may vary depending on usage charges.

☒ Turn on data transformation

- Under **AWS Lambda function**, click **Create new**.

AWS Lambda function

Choose a Lambda function or enter an ARN

Format: arn:aws:lambda:[Region]:[AccountId]:function:[FunctionName]

Version or alias

Choose a version or alias

[Browse](#) [Create function](#)

- In the Lambda creation window, choose **General Amazon Data Firehose Processing**.
- Select **Use blueprint**.

Create an AWS Lambda function using an AWS Lambda blueprint

Choose one of the AWS Lambda blueprints below to navigate to the AWS Lambda management console where you can create and configure your AWS Lambda function.

Lambda blueprints (5)

Lambda blueprint	Description
<input checked="" type="radio"/> General Amazon Data Firehose Processing	An Amazon Data Firehose stream processor that accesses the records in the input and returns them with a processing status. Use this processor for any custom transformation logic.
<input type="radio"/> Apache Log to JSON	An Amazon Data Firehose stream processor that converts input records from Apache Common Log format to JSON.
<input type="radio"/> Apache Log to CSV	An Amazon Data Firehose stream processor that converts input records from Apache Common Log format to CSV.
<input type="radio"/> Syslog to JSON	An Amazon Data Firehose stream processor that converts input records from RFC3164 Syslog format to JSON.
<input type="radio"/> Syslog to CSV	An Amazon Data Firehose stream processor that converts input records from RFC3164 Syslog format to CSV.
<input type="radio"/> Amazon Data Firehose Process Record Streams as source	An Amazon Data Firehose stream processor that accesses the Kinesis Data Streams records in the input and returns them with a processing status.

[Cancel](#) [Use blueprint](#)

- This will redirect you to lambda interface.
- Name the lambda function.
- In Blueprint Name, search for the firehose.
- Choose **Runtime: Python 3.x**.
- Under **Permissions**, choose:
 - ☐ **Use an existing role** → select the role created earlier (e.g., LambdaKinesisRole) *(This role should have permissions for Kinesis and S3 access)*.
- Review the pre-loaded code — it will **decode, process, and re-encode** records.
- Click **Create function**.

Basic information

Blueprint name

Process records sent to an Amazon Data Firehose stream

An Amazon Data Firehose stream processor that accesses the records in the input and returns them with a processing status. Use this processor for any custom transformation logic.

python3.12

Function name

Enter a name that describes the purpose of your function.

firehosetransformfunction

Function name must be 1 to 64 characters, must be unique to the Region, and can't include spaces. Valid characters are a-z, A-Z, 0-9, hyphens (-), and underscores (_).

Runtime

python3.12

Architecture

x86_64

Execution role

Choose a role that defines the permissions of your function. To create a custom role, go to the [IAM console](#).

☐ Create a new role with basic Lambda permissions

☒ Use an existing role

☐ Create a new role from AWS policy templates

Existing role

Choose an existing role that you've created to be used with this Lambda function. The role must have permission to upload logs to Amazon CloudWatch Logs.

LambdaKinesisRole

[View the LambdaKinesisRole role](#) on the IAM console.

- Return to your **Firehose stream** configuration.
- In the **Transformation** section, select the Lambda function you just created.

AWS Lambda function

arn:aws:lambda:us-east-1:463646775279:function:firehosetransformfunction:\$LATEST

Version or alias: \$LATEST

[Browse](#) [Create function](#)

Format: arn:aws:lambda:[Region]:[AccountId]:function:[FunctionName]

AWS Lambda function timeout
The current timeout of the specified AWS Lambda function is 3 seconds. To reduce the risk of the AWS Lambda function timing out before data transformation is complete, increase the timeout to 1 minute or longer in the Advanced settings section of your AWS Lambda configuration. [Go to AWS Lambda configuration.](#)

transform and convert records

Choose an AWS Lambda function

AWS Lambda function timeout
The current timeout of the specified AWS Lambda function is 3 seconds. To reduce the risk of the AWS Lambda function timing out before data transformation is complete, increase the timeout to 1 minute or longer in the Advanced settings section of your AWS Lambda configuration. [Go to AWS Lambda configuration.](#)

AWS Lambda functions (1/1)

Find AWS Lambda functions

Function name	Description	Runtime	Timeout
firehosetransformfunction	An Amazon Data Firehose stream processor that accesses the records in...	python3.12	3 seconds

[Cancel](#) [Choose](#)

- Open **AWS CloudShell** or your **AWS CLI**.
- Use the **put-record** command to send records into your Kinesis data stream

```
{
  "ShardId": "shardId-000000000001",
  "SequenceNumber": "49668313584833940833244021898261289758589924286289354778"
}

{
  "ShardId": "shardId-000000000001",
  "SequenceNumber": "49668313584833940833244021738267802161721041262737358866"
}

$ aws kinesis put-record --stream-name myfirstdatastream --partition-key "PartitionKey2" --data $(echo -n "Data Entry 2" | base64)

{
  "ShardId": "shardId-000000000001",
  "SequenceNumber": "49668313584833940833244021739875673581888498339974479898"
}

$ aws kinesis put-record --stream-name myfirstdatastream --partition-key "PartitionKey" --data $(echo -n "Data Entry 1" | base64)

{
  "ShardId": "shardId-000000000001",
  "SequenceNumber": "49668313584833940833244021898261289758589924286289354778"
}

$ aws kinesis put-record --stream-name myfirstdatastream --partition-key "PartitionKey" --data $(echo -n "Data Entry 1" | base64)

{
  "ShardId": "shardId-000000000001",
  "SequenceNumber": "49668313584833940833244021892462663676108164070868254738"
}

$ aws kinesis put-record --stream-name myfirstdatastream --partition-key "PartitionKey2" --data $(echo -n "Data Entry 2" | base64)

{
  "ShardId": "shardId-000000000001",
  "SequenceNumber": "49668313584833940833244021896742261077543951624206024722"
}

$ aws kinesis put-record --stream-name myfirstdatastream --partition-key "PartitionKey2" --data $(echo -n "Data Entry 2" | base64)
```

- Wait a few minutes for Firehose to process and deliver the transformed data.
- Go to the **S3 Console** → Open your **destination bucket**.
- Locate the new folder (organized by **year/month/day**).
- Download a file and open it to confirm the **transformed data**.

Amazon S3 > Buckets > [kinesis-target-bucket01](#) > [2025/](#) > [10/](#) > [24/](#) > [14/](#)

Amazon S3

General purpose buckets

Directory buckets

Table buckets

Vector buckets

Access Grants

Access Points (General Purpose Buckets, FSx file systems)

Access Points (Directory Buckets)

Object Lambda Access Points

Multi-Region Access Points

Batch Operations

IAM Access Analyzer for S3

Block Public Access settings for this account

Objects (2)

[Copy S3 URI](#) [Copy URL](#) [Download](#) [Open](#) [Delete](#) [Actions](#) [Create folder](#) [Upload](#)

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

Find objects by prefix

Name	Type	Last modified	Size	Storage class
KDS-S3-aszx-2-2025-10-24-14-16-30-c768e417-c5a6-4f03-9a2f-6331dc4c26d7	-	October 24, 2025, 19:51:31 (UTC+05:30)	72.0 B	Standard
KDS-S3-aszx-2-2025-10-24-14-22-29-e75cf129-a549-4d38-b77a-dafe5a398169	-	October 24, 2025, 19:58:32 (UTC+05:30)	48.0 B	Standard

