

# The one mistake everyone makes when using Kinesis with Lambda

Yan Cui · Follow  
Published in theburningmonk.com · 5 min read · Dec 11, 2023

9 Q Bookmarks Refresh Share More



AWS Kinesis and Lambda are a great combo for processing large amounts of data in real-time. However, there's a common oversight that many developers make when integrating these two services.

There are established best practices for configuring Lambda's [EventSourceMapping](#) [1] for Kinesis:

- Configure an `onFailure` destination for failed records.
- Enable `BisectBatchOnFunctionError`.
- Override `MaximumRetryAttempts`. Choose a value that gives failed messages a reasonable number of retries before sending them to the `onFailure` destination.

These are great advice for building a resilient data processing pipeline and guarding against poison messages. But one crucial detail is missing: **the payload captured in the `OnFailure` destination does not contain the event payload.**

This is what will be captured in the `onFailure` destination.

```
{
  "requestContext": {
    "requestId": "cf6fa2a6-48e2-49a6-bc23-7ed2b154b176",
    "functionArn": "arn:aws:lambda:us-east-1:xxx:function:kinesis-processor",
    "condition": "RetryAttemptsExhausted",
    "approximateInvokeCount": 4
  },
  "responseContext": {
    "statusCode": 200,
    "executedVersion": "$LATEST",
    "functionError": "Unhandled"
  },
  "version": "1.0",
  "timestamp": "2023-12-10T22:06:23.446Z",
  "KinesisBatchInfo": {
    "shardId": "shardId-000000000000",
    "startSequenceNumber": "4964723877401875844914379084747495239034204468389137",
    "endSequenceNumber": "49647238774018758449143790941123017485931487851477729",
    "approximateArrivalOfFirstRecord": "2023-12-10T22:05:59.286Z",
    "approximateArrivalOfLastRecord": "2023-12-10T22:06:19.196Z",
    "batchSize": 3,
    "streamArn": "arn:aws:kinesis:us-east-1:xxx:stream/MyKinesisStream"
  }
}
```

The `KinesisBatchInfo` contains the shard ID, sequence numbers and batch size. The actual data payload — the heart of the event that you need for processing and analysis — is not included.

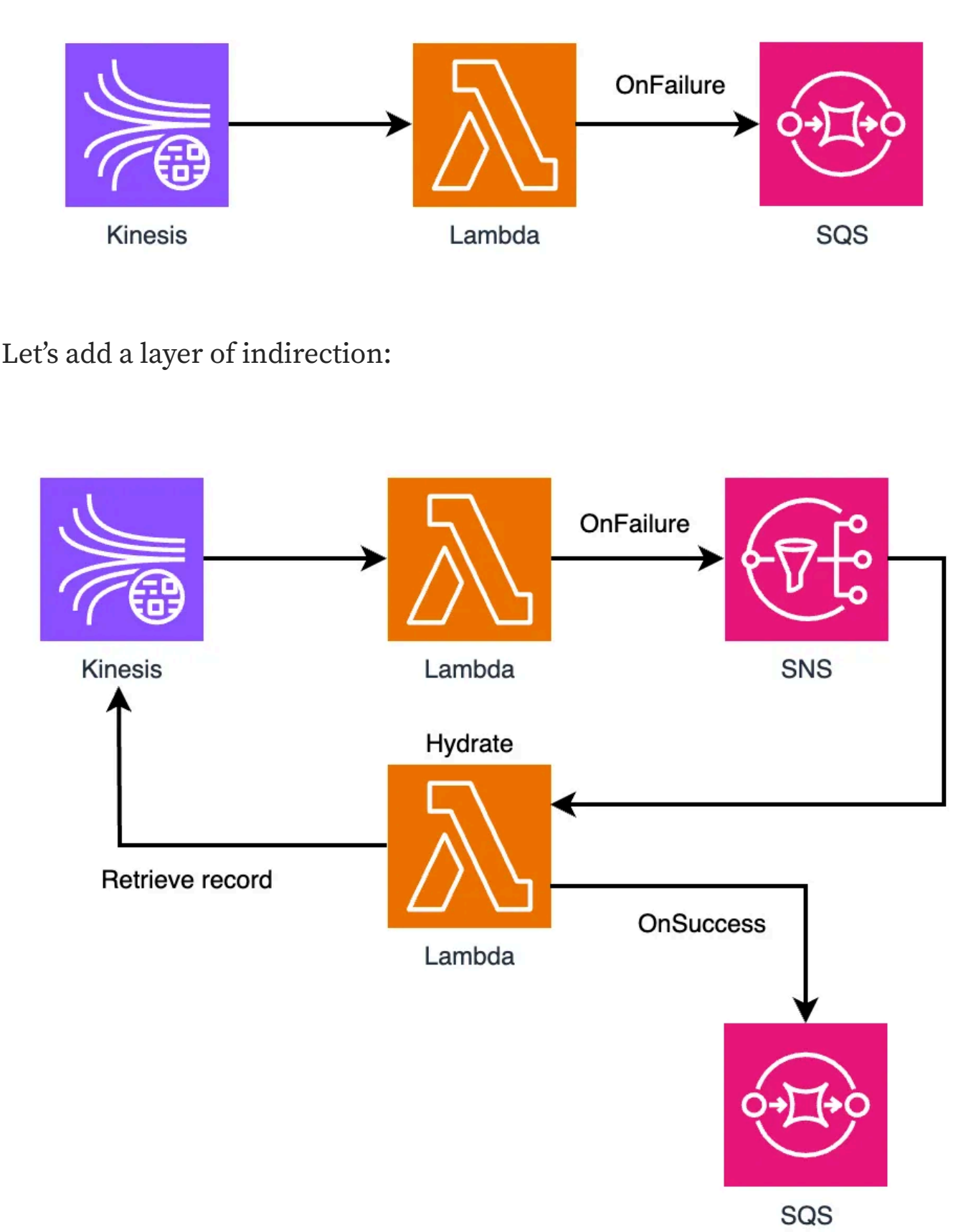
This omission can lead to significant issues:

1. **Difficulty in debugging:** Without the actual event data, pinpointing the cause of the failure becomes a complex task.
2. **Data loss:** If you are not able to fetch the record before it expires from the stream, then the data is lost forever. Default retention for a Kinesis stream is 24 hours. If the error occurs on a Saturday, likely, your developers won't see it until they are back in the office on Monday. By then, the record is no longer available in the stream.
3. **Increased complexity:** Developers need to implement additional mechanisms to retrieve and store the original event data.

Most people don't realise this and many have been caught off-guard when they encounter a failed message for the first time!

To handle this situation, you need to hydrate the event data yourself.

Instead of using an SQS queue as the `onFailure` destination directly:



- For the Kinesis function, use an SNS topic as the `onFailure` destination.
- Subscribe another Lambda function to this topic. This `hydrate` function retrieves the record from the Kinesis stream.
- Because the `hydrate` function is invoked asynchronously by SNS, we can configure an `onSuccess` destination to write its return value to the original SQS DLQ. This way, we don't have to maintain the code to write to SQS. However, the downside is that the message captured in the SQS queue is **more verbose** (see below). All we care about is the `responsePayload` where we enriched the original `onFailure` message with the `KinesisBatch` array.

```
{
  "version": "1.0",
  "timestamp": "2023-12-10T22:43:49.863Z",
  "requestContext": {
    "requestId": "2944cfad-d4c4-4b28-ad45-bfdd7f4526ef",
```



```

    "functionArn": "arn:aws:lambda:us-east-1:374852340823:function:hydrate-kines",
    "condition": "Success",
    "approximateInvokeCount": 1
  },
  "requestPayload": {
    "Records": [
      {
        "EventSource": "aws:sns",
        "EventVersion": "1.0",
        "EventSubscriptionArn": "arn:aws:sns:us-east-1:374852340823:hydrate-kines",
        "Sns": {
          "Type": "Notification",
          "MessageId": "002d4204-3a47-5737-b1b5-1ff47b22a32b",
          "TopicArn": "arn:aws:sns:us-east-1:374852340823:hydrate-kines-dlq-de",
          "Subject": null,
          "Message": "{\"requestContext\":{\"requestId\":\"5577e27b-a674-48cc-9f\",",
          "Timestamp": "2023-12-10T22:43:49.506Z",
          "SignatureVersion": "1",
          "Signature": "MauQRhErEYexthVBzNasLk2LSkJLWBgdrZjIubK4pyIMs7XSGM4dTl",
          "SigningCertUrl": "https://sns.us-east-1.amazonaws.com/SimpleNotificat",
          "UnsubscribeUrl": "https://sns.us-east-1.amazonaws.com/?Action=Unsubsc",
          "MessageAttributes": {}
        }
      }
    ]
  },
  "responseContext": {
    "statusCode": 200,
    "executedVersion": "$LATEST"
  },
  "responsePayload": {
    "requestContext": {
      "requestId": "5577e27b-a674-48cc-9fcb-aa6cf675e939",
      "functionArn": "arn:aws:lambda:us-east-1:374852340823:function:hydrate-kin",
      "condition": "RetryAttemptsExhausted",
      "approximateInvokeCount": 4
    },
    "responseContext": {
      "statusCode": 200,
      "executedVersion": "$LATEST",
      "functionError": "Unhandled"
    },
    "version": "1.0",
    "timestamp": "2023-12-10T22:43:49.459Z",
    "KinesisBatchInfo": {
      "shardId": "shardId-000000000000",
      "startSequenceNumber": "49647238774018758449143791361301066723689190929215",
      "endSequenceNumber": "4964723877401875844914379136130469350114803674988525",
      "approximateArrivalOfFirstRecord": "2023-12-10T22:42:59.398Z",
      "approximateArrivalOfLastRecord": "2023-12-10T22:43:20.392Z",
      "batchSize": 4,
      "streamArn": "arn:aws:kinesis:us-east-1:374852340823:stream/hydrate-kines"
    },
    "KinesisBatch": [
      "Hello, theburningmonk!",
      "Hello, theburningmonk!",
      "Hello, theburningmonk!",
      "Hello, theburningmonk!"
    ]
  }
}
```

If you're interested in trying this out yourself, then please take a look at [this Github repo](#) [2]. In particular, look at the `hydrate` function to see how to retrieve the records from the Kinesis stream.

It's worth noting that [this approach also works for DynamoDB streams](#). In the GitHub repo linked above, you will find working examples for both Kinesis and DynamoDB streams.

Next steps

I hope this helps you avoid a very common mistake people make when working with Kinesis and Lambda.

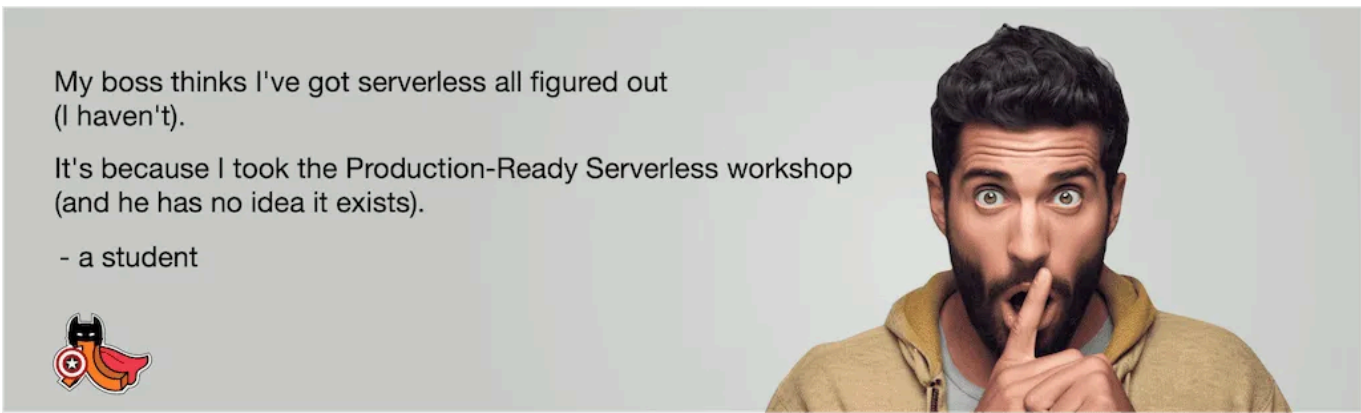
To learn more about building production-ready serverless applications, check out my [upcoming workshop](#) [3]. I'm working on some new lessons to incorporate the latest changes from re:Invent 2023. And I'm adding support for CDK as well, due to popular demand!



Hope to see you there.


Links

- [1] [EventSourceMapping documentation](#)
- [2] [Git Hub repo for a working demo](#)
- [3] [Production-Ready Serverless workshop](#)



Originally published at <https://theburningmonk.com> on December 11, 2023.

- AWS
- AWS Lambda
- Kinesis
- Cloud
- Architecture



**Written by Yan Cui**

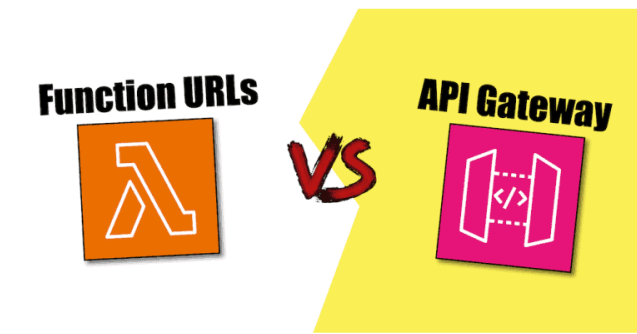
5.8K Followers

Editor for theburningmonk.com

AWS Serverless Hero. Follow me to learn practical tips and best practices for AWS and Serverless.

Follow

More from Yan Cui and theburningmonk.com



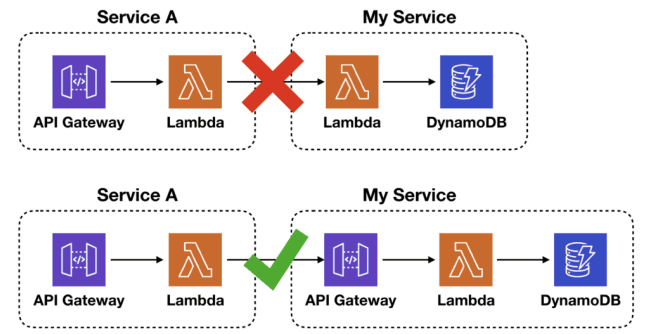
**Function URLs vs API Gateway**

Yan Cui in theburningmonk.com

**When to use API Gateway vs. Lambda Function URLs**

"Lambdalth" is a monolithic approach to building serverless applications where a...


5 min read · Mar 3, 2024



**Are Lambda-to-Lambda calls really so bad?**

If you utter the words "I call a Lambda function from another Lambda function" you...

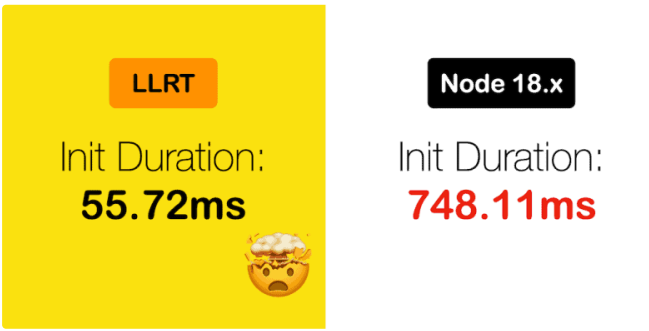
8 min read · Jul 12, 2020



**How to secure CI/CD roles without burning production to the ground**

By now, most of us have moved away from using IAM users for CI/CD pipelines. Instead...

6 min read · Feb 16, 2024



**First impressions of the fastest JavaScript runtime for Lambda**

I thought Lambda needed a specialised runtime. One that works well with its...

5 min read · Feb 27, 2024