
AWS
re:Invent
NOV. 28 – DEC. 2, 2022 | LAS VEGAS, NV

SVS316

Tuning AWS Lambda event source mappings

Andres Victoria (he/him)

Senior Solutions Architect
AWS

Hernan Garcia (he/him)

Senior Solutions Architect
AWS

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

What we are going to cover

Event source mappings (ESM)

- Amazon SQS, self-hosted Kafka/Confluent, Amazon Kinesis, Amazon DynamoDB

Scaling process

Practices to improve performance

Failure management

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS Lambda ESM for self-hosted Kafka and Confluent

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Kafka as an event source – A primer

- The AWS Lambda service (the consumer) internally polls for new messages from Kafka
- The consumer reads the messages from a topic in batches, from one or more Kafka partitions
- The consumer synchronously invokes a Lambda function (the target) and provides a batch of messages as an event payload
- The target processes the messages in the batch
- The consumer continues to poll batches of messages until there are no more messages in the topic

aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Lambda ESM for Kafka attributes

Attribute	Required	Default	Min–Max / Values
Batch size	No	100	1–10,000
Starting position	Yes	Latest | Trim horizon	NA
Batch window	No	500 ms	0–300
Consumer group ID	No	NA	NA

aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Some considerations

1. If you specify an existing **consumer group ID**, and there are other active pollers within that consumer group, Lambda doesn't receive all the messages in the Kafka topic
2. If you specify an existing **consumer group ID**, Lambda ignores the **Starting position** parameter for your ESM; it uses the committed offset of the consumer group
3. If you specify a **consumer group ID**, and Kafka cannot find an existing consumer group, then Lambda configures your event source with the specified **Starting position**

aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Lambda ESM for Kafka scaling behavior

After creating the Kafka event source, one consumer to process all partitions in the Kafka topic

In 1 min intervals, Lambda evaluates the consumer offset lag of all the partitions in the topic

When needed, Lambda adds or removes consumers from the topic. The scaling process of adding or removing consumers occurs within 3 minutes of evaluation

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Lambda ESM for Kafka failure modes

If the function times out, Lambda retries the whole batch until processing succeeds or messages expire

If the function returns an error



If a function encounters an unrecoverable error, the ESM is paused and consumer stops processing messages

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Keep in mind

- Maximum Lambda function concurrency = number of Kafka partitions
- Partitions are processed by one and only one consumer
- Messages in a partition are processed sequentially
- For low volumes of messages, data that take time to build up, or to reduce costs, use high values for Batch size and Batch window
- Monitor your Lambda function's `OffsetLag` metric
- Set appropriate timeouts per Lambda function
- Implement idempotency

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Let’s discuss

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Lambda ESM for Amazon Simple Queue Service (Amazon SQS)

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon SQS as an event source for Lambda

- Lambda service is responsible for polling messages from the queue
- You don’t need to include code for deleting messages from the queue



aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Lambda ESM for Amazon SQS – Attributes

Attribute	Required	Default	Min–Max / Values
Batch size	N	10	1–10,000
Batch window	N	0	0–300
`FunctionResponseTypes`	N		`ReportBatchItemFailures`

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Lambda scaling behavior with Amazon SQS

For standard queues

- Initially: Up to 5 batches
- If there are messages still available: up to 60 new processes reading batches per minute



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Lambda scaling behavior with Amazon SQS

For FIFO queues

- One batch at a time for each group





© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Failure management

- On failure, messages return to the queue after visibility timeout is reached and they are retried until
  - Successful processing
  - Retention period is reached



- Failure during the processing of a batch = all messages become visible again (default)

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Failure management

- Use a DLQ
  - Set `maxReceiveCount` >= 5
- Set `FunctionResponseTypes` to `ReportBatchItemFailures`
- Decrease message retention period in the queue



aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Keep in mind

- A large batch size can improve efficiency for workloads that are very fast or have a lot of overhead
- Be aware of Lambda payload maximum size (6 MB)
- Use a small batch size if processing every message is slow
- Configure visibility timeout **>6x** function timeout

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Let’s discuss

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Lambda ESM for Kinesis Data Streams

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Kinesis as an event source for Lambda

- Lambda service is responsible for polling messages from the stream
- By default, 1 instance of your function per shard



aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Lambda ESM for Kinesis – Attributes

Attribute	Required	Default	Min–Max / Values
Batch size	N	100	1–10,000
Split batch on error	N	false	
On-failure destination	N		
`FunctionResponseTypes`	N		`ReportBatchItemFailures`
Batch window	N	0	0–300
Maximum age of record	N	-1	-1 to 604,800
Retry attempts	N	-1	-1 to 10,000

aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Lambda ESM for Kinesis – Attributes

Attribute	Required	Default	Min–Max / Values
ParallelizationFactor	N	1	1–10
StartingPosition	Y		AT_TIMESTAMP TRIM_HORIZON LATEST

aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Lambda scaling behavior with Kinesis Data Streams

- By default, 1 synchronous invocation per batch per shard
- Up to 10 concurrent invocations per shard with `ParallelizationFactor`

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Lambda ESM for Kinesis failure modes

- If Lambda throttles or returns an error without invoking the function, Lambda retries until
  - Records expire
  - Max age reached
- If the function is invoked but returns an error, Lambda retries until
  - Records expire
  - Max age reached
  - Max retries reached



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Keep in mind

- Be aware that failed messages block the processing of new messages
- Enabling `BisectBatchOnFunctionError`
- Configure failed-event destination



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Keep in mind

- Lambda processes records in each shard in order
- If records are processed more than once, they might be processed out of order
- Maximum Lambda concurrency = **# of shards * Parallelization factor**
- Parallelization factor will only work if there is a good sample of partition key values
- Be aware that with default settings, a bad record can block processing on the affected shard for up to one week (**stalled shard**)
- There are several attributes that allow you to do a better handling of failed messages

aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

DynamoDB ESM specifics

- Polling frequency: 4 times per second
- Number of consumers: Max 2 Lambda functions per stream
- DynamoDB scales up to the number of shards
- Retention period: 24 hours

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Let’s discuss

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Lambda ESM for DynamoDB

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

DynamoDB as an event source – A primer

- The AWS Lambda service (the consumer) internally polls shards in the DynamoDB stream **four times per second**
- The consumer reads the messages from a stream in batches and **synchronously** invokes a Lambda function (the target) with a batch of messages as an event payload
- The consumer continues to poll batches of messages until there are no more messages in the stream
- Messages are **delivered exactly once** and **strictly ordered by key**

aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Lambda ESM for DynamoDB – Attributes

Attribute	Required	Default	Min–Max / Values
Batch size	N	100	1–10,000
Split batch on error	N	FALSE	TRUE/FALSE
On-failure destination	N		Amazon SQS or Amazon SNS
Batch window	N	0	0–300
Maximum age of record	N	-1	-1 to 604,800
Retry attempts	N	-1	-1 to 10,000

aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Lambda ESM for DynamoDB – Attributes

Attribute	Required	Default	Min–Max / Values
Concurrent batches per shard	N	1	1–10

Starting position	Y		`AT_TIMESTAMP`
			`TRIM_HORIZON`
			`LATEST`

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Some considerations

1. You can subscribe up to **two Lambda functions** to the same DynamoDB stream
2. There is not direct control of the number of shards
3. When a DynamoDB table's ability to serve a set amount of traffic doubles, so does the number of shards in the stream

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Lambda ESM for DynamoDB failure modes

If **Lambda throttles the function or returns an error without invoking the function**, Lambda retries until the records expire or exceed the maximum age that you configure on the event source mapping

If **the function receives the records but returns an error**, Lambda retries until the records in the batch expire, exceed the maximum age, or reach the configured retry quota

- You can also split the failed batch

If error handling fails, **Lambda discards the records and continues processing batches from the stream**

aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Let’s discuss

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Keep in mind

- DynamoDB on-demand mode has the ability to instantly scale; make sure your Lambda function can handle that scaling
- Visibility is key: watch `IteratorAge` and Lambda concurrency
- If using a parallelization factor > 1, implement idempotency in your Lambda function
- Data retention time is 24 hours, act in accordance; you can use on-failure destination

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Keep in mind

- For low-throughput streams, set Batch window to higher values
- If subscribing more than 1 Lambda function to the stream, consider adding Amazon Kinesis Data Streams
- With ESM default values, a “poison pill” will block processing of the shard; configure number of retries and a maximum record age

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

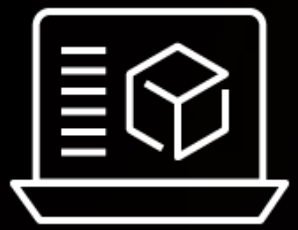
Related sessions

ID	Title	Type	Schedule
SVS402-R	Serverless stream processing with AWS Lambda, Amazon Kinesis & Kafka	Chalk Talk	Wednesday, 10:45 AM
SVS305-R	Understanding AWS Lambda concurrency	Chalk Talk	Wednesday, 3:15 PM
SVS305-R1	Understanding AWS Lambda concurrency	Chalk Talk	Friday, 10:45 AM
SVS314-R1	AWS Lambda performance tuning: Best practices and guidance	Chalk Talk	Wednesday, 12:15 PM
SVS314-R	AWS Lambda performance tuning: Best practices and guidance	Chalk Talk	Thursday, 4:15 PM

aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Continue your AWS Serverless learning

Learn at your own pace



Expand your serverless skills with our learning plan on **AWS Skill Builder**

Increase your knowledge



Use our **Ramp-Up Guides** to build your serverless knowledge

Earn AWS Serverless badge



Demonstrate your knowledge by achieving **digital badges**

https://s12d.com/serverless-learning

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Thank you!

Andres Victoria
andvic@amazon.com

Hernan Garcia
hernagar@amazon.nl



Please complete the session survey in the **mobile app**

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.