

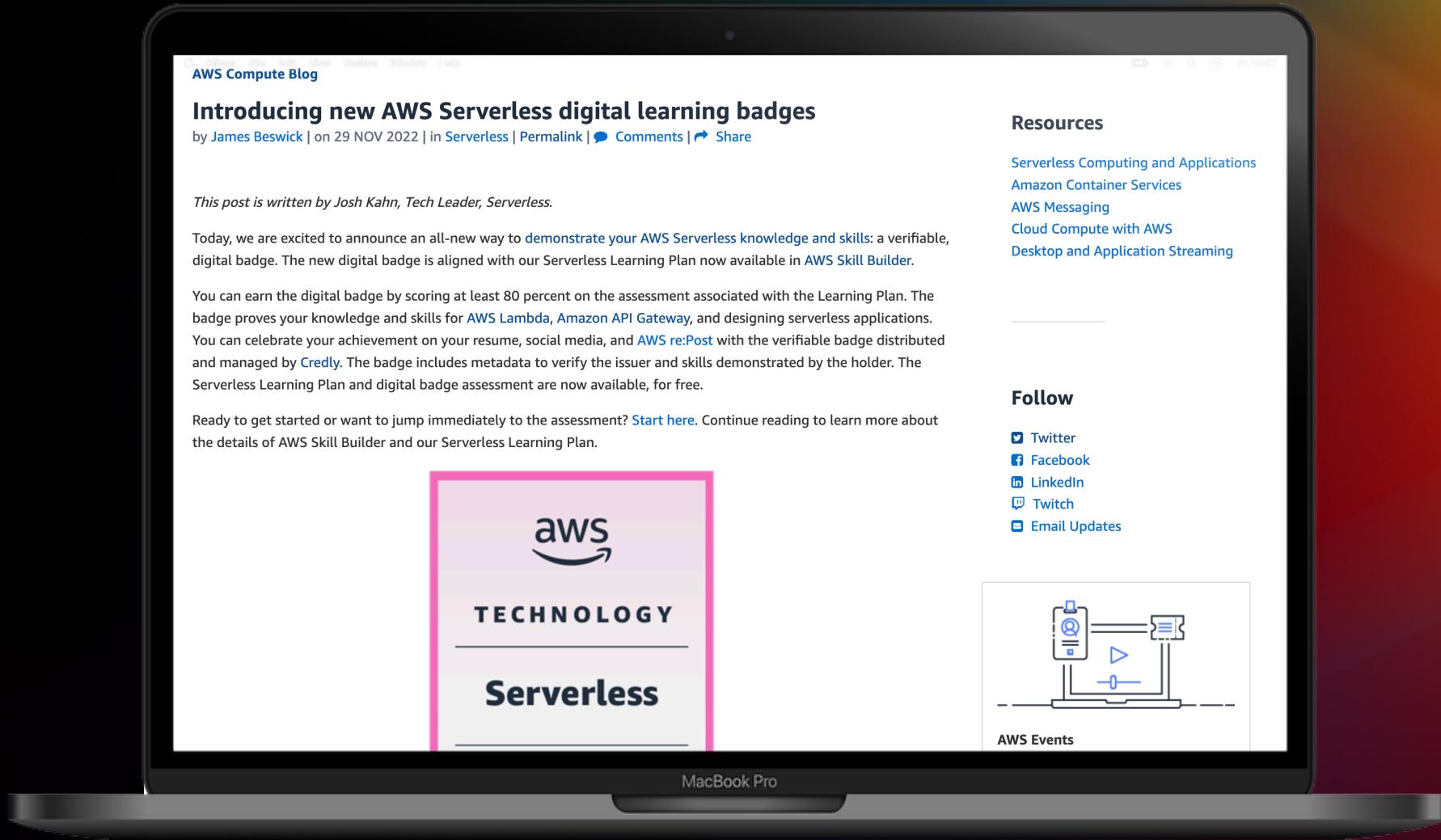
# What's new in AWS Serverless

2023



**Anton Aleksandrov**  
Principal Solutions Architect  
AWS Serverless and Event-Driven Architectures

# Serverless Digital Learning Badges - get yours NOW!

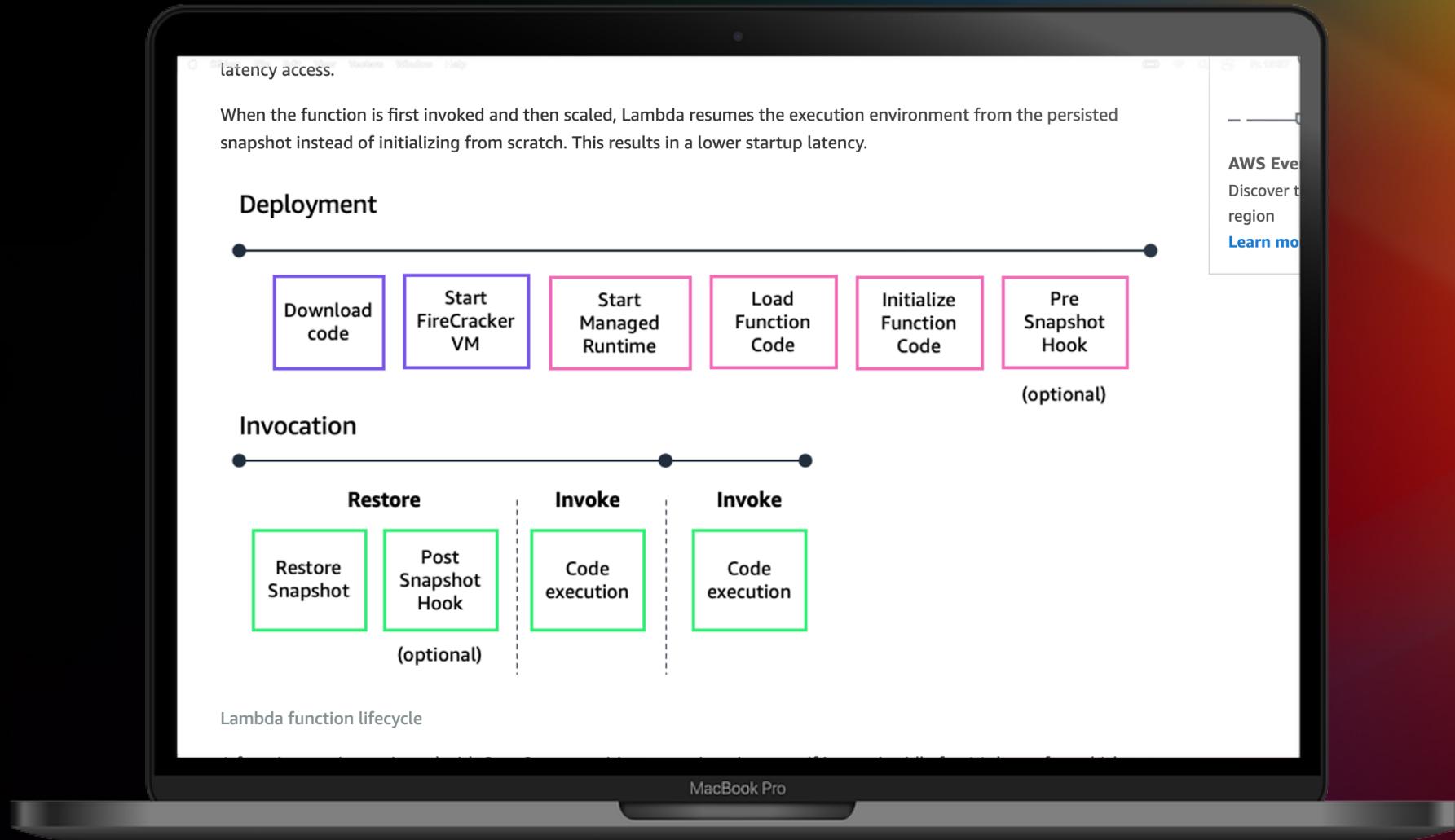


<https://aws.amazon.com/blogs/compute/introducing-new-aws-serverless-digital-learning-badges/>



© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# Lambda SnapStart 😊



<https://aws.amazon.com/blogs/aws/new-accelerate-your-lambda-functions-with-lambda-snapstart/>



© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

<https://youtu.be/ZbnAithBNYY>

# Node.js 18 and .NET 7 AOT support



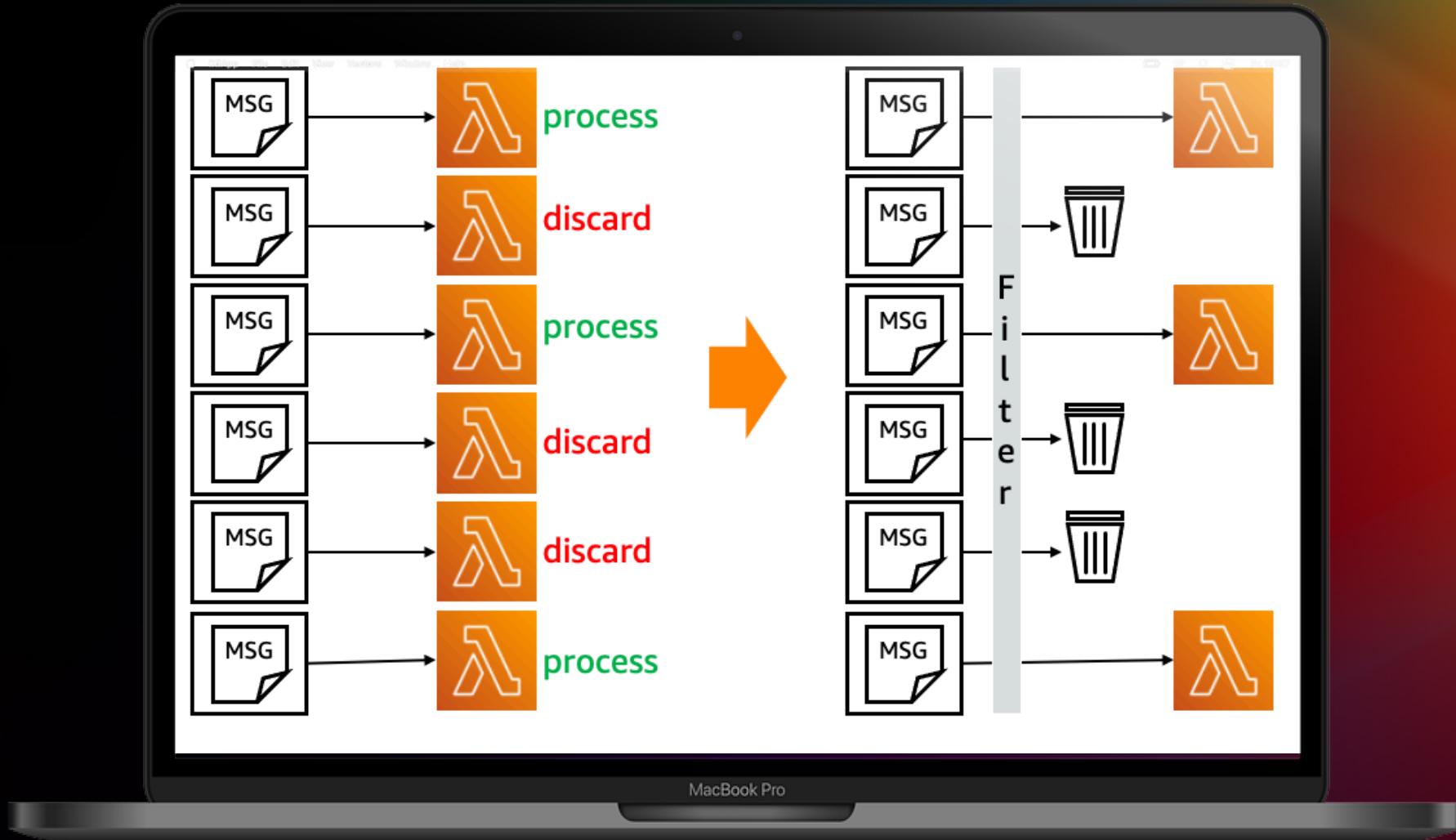
<https://aws.amazon.com/about-aws/whats-new/2022/11/aws-lambda-support-node-js-18/>

<https://aws.amazon.com/blogs/compute/building-serverless-net-applications-on-aws-lambda-using-net-7/>



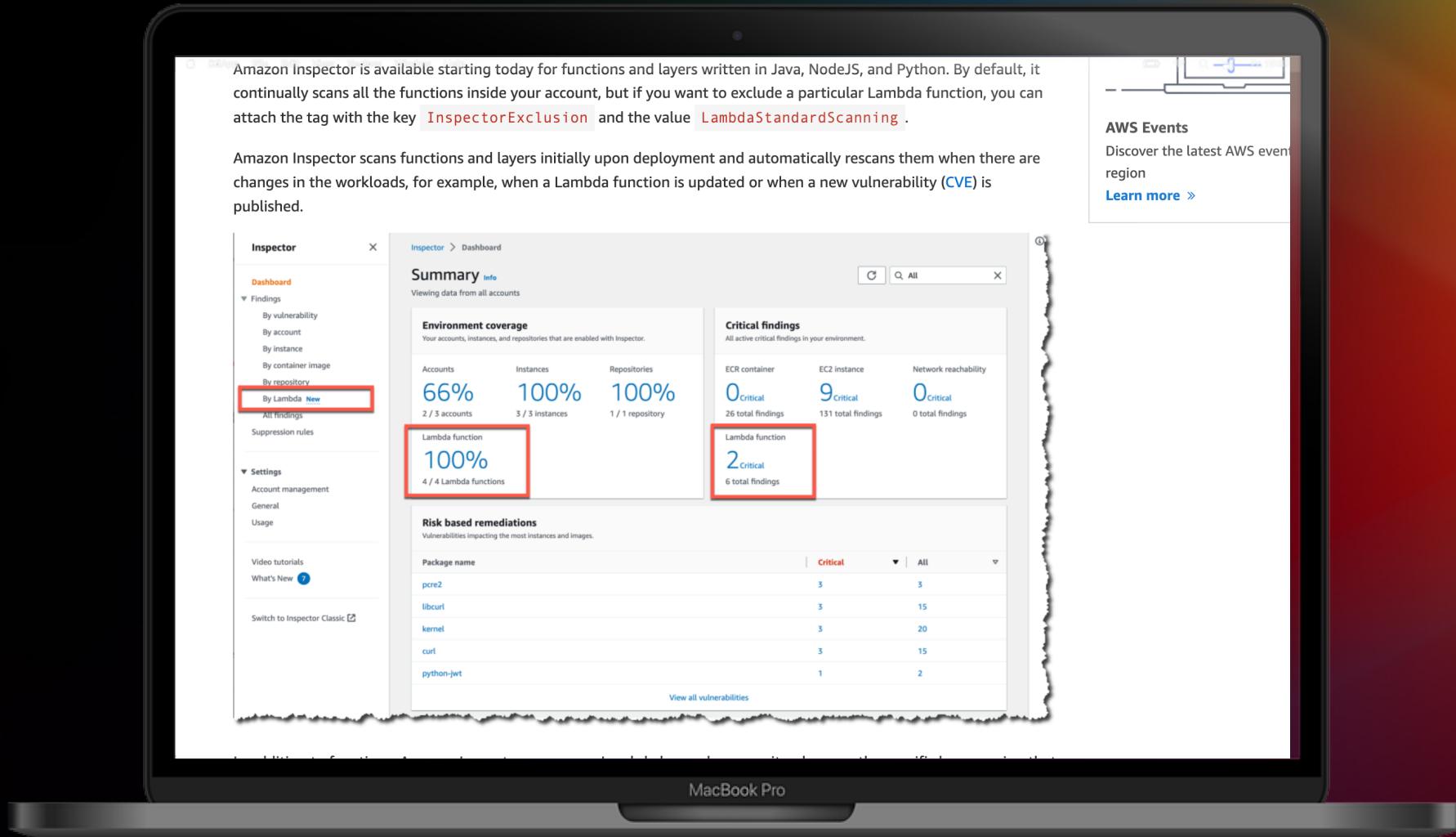
© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# Lambda event filtering for additional event sources



<https://aws.amazon.com/about-aws/whats-new/2022/10/aws-lambda-event-filtering-amazon-msk-kafka-mq-apache-activemq-amazon-mq-rabbit-mq/>

# Amazon Inspector Scans Lambda Functions Vulnerabilities



<https://aws.amazon.com/blogs/aws/amazon-inspector-now-scans-aws-lambda-functions-for-vulnerabilities/>



© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# X-Ray trace propagation from SQS to Lambda

Amazon SQS and AWS X-Ray

[PDF](#) | [RSS](#)

AWS X-Ray integrates with Amazon Simple Queue Service (Amazon SQS) to trace messages that are passed through an Amazon SQS queue. If a service traces requests by using the X-Ray SDK, Amazon SQS can send the tracing header and continue to propagate the original trace from the sender to the consumer with a consistent trace ID. Trace continuity enables users to track, analyze, and debug throughout downstream services.

AWS X-Ray supports tracing event-driven applications using Amazon SQS and AWS Lambda. Use the CloudWatch console to see a connected view of each request as it's queued with Amazon SQS and processed by a downstream Lambda function. Traces from upstream message producers are automatically linked to traces from downstream Lambda consumer nodes, creating an end-to-end view of the application. For more information, see [tracing event-driven applications](#).

```
graph LR; PF((ProducerFunction)) --> SQS((https://...MySQSQueue  
SQS Queue)); SQS -.-> CF((ConsumerFunction  
Lambda Context));
```

Amazon SQS supports the following tracing header instrumentation:

- Default HTTP Header – The X-Ray SDK automatically populates the trace header as an HTTP header when you call Amazon SQS through the AWS SDK. The default trace header is carried by `X-Amzn-Trace-Id` and corresponds to all messages included in a `SendMessage` or `SendMessageBatch` request. To learn more about the default HTTP header, see [Tracing header](#).
- AWSTraceHeader System Attribute – The `AWSTraceHeader` is a message system attribute reserved by Amazon SQS to carry the X-Ray trace header with messages in the queue. `AWSTraceHeader` is available for use even when auto-instrumentation through the X-Ray SDK is not, for example when building a tracing SDK for a new language. When both header instrumentations are set, the message system attribute overrides the HTTP trace header.

When running on Amazon EC2, Amazon SQS supports processing one message at a time. This applies when running on an on-premises host, and when using container services, such as AWS Fargate, Amazon ECS, or AWS App Mesh.

The trace header is excluded from both Amazon SQS message size and message attribute quotas. Enabling X-Ray tracing will not exceed your Amazon SQS quotas. To learn more about AWS quotas, see [Amazon SQS Quotas](#).

MacBook Pro

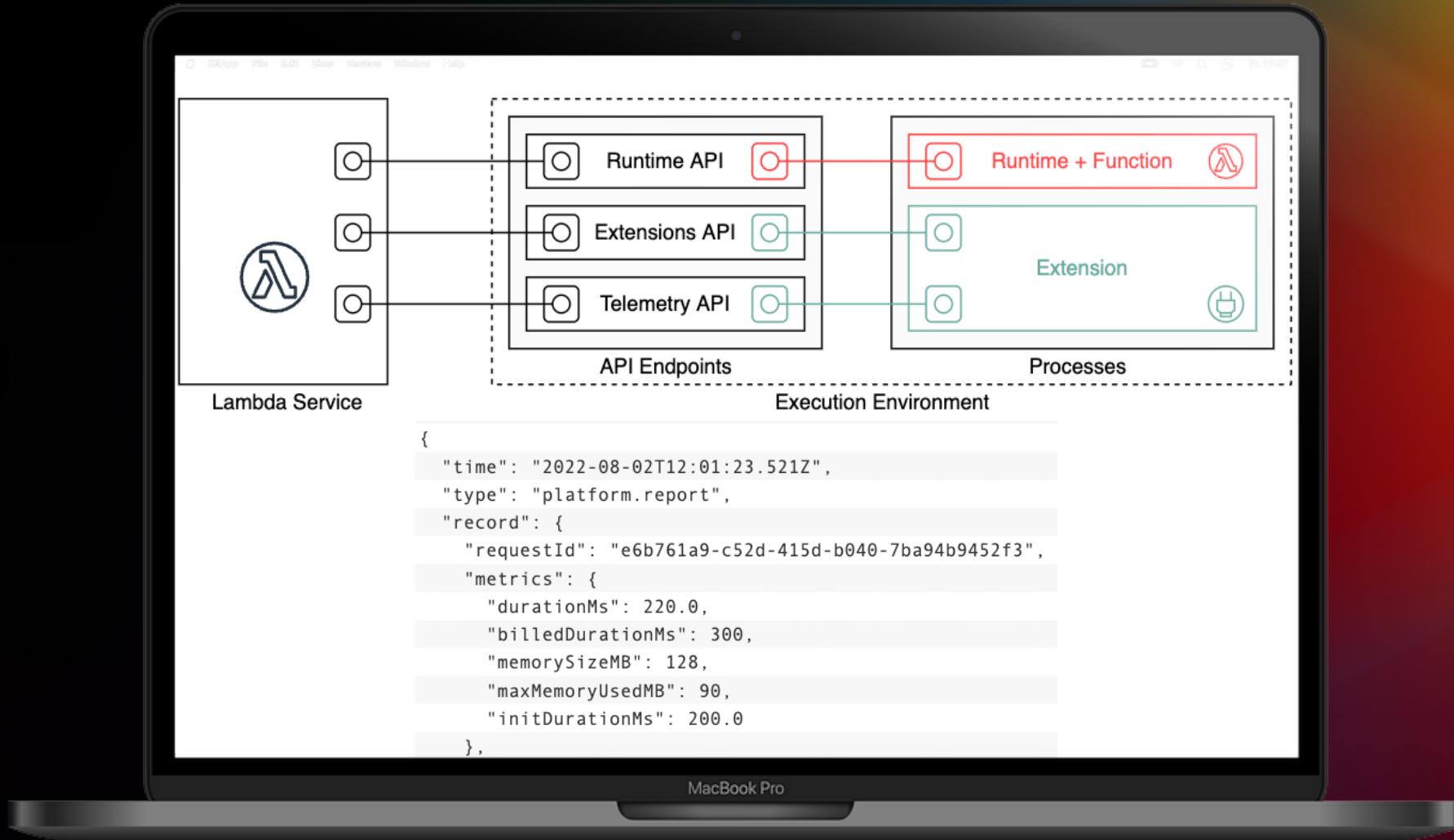


<https://aws.amazon.com/about-aws/whats-new/2022/11/aws-x-ray-trace-linking-event-driven-applications-amazon-sqs-lambda/>



© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

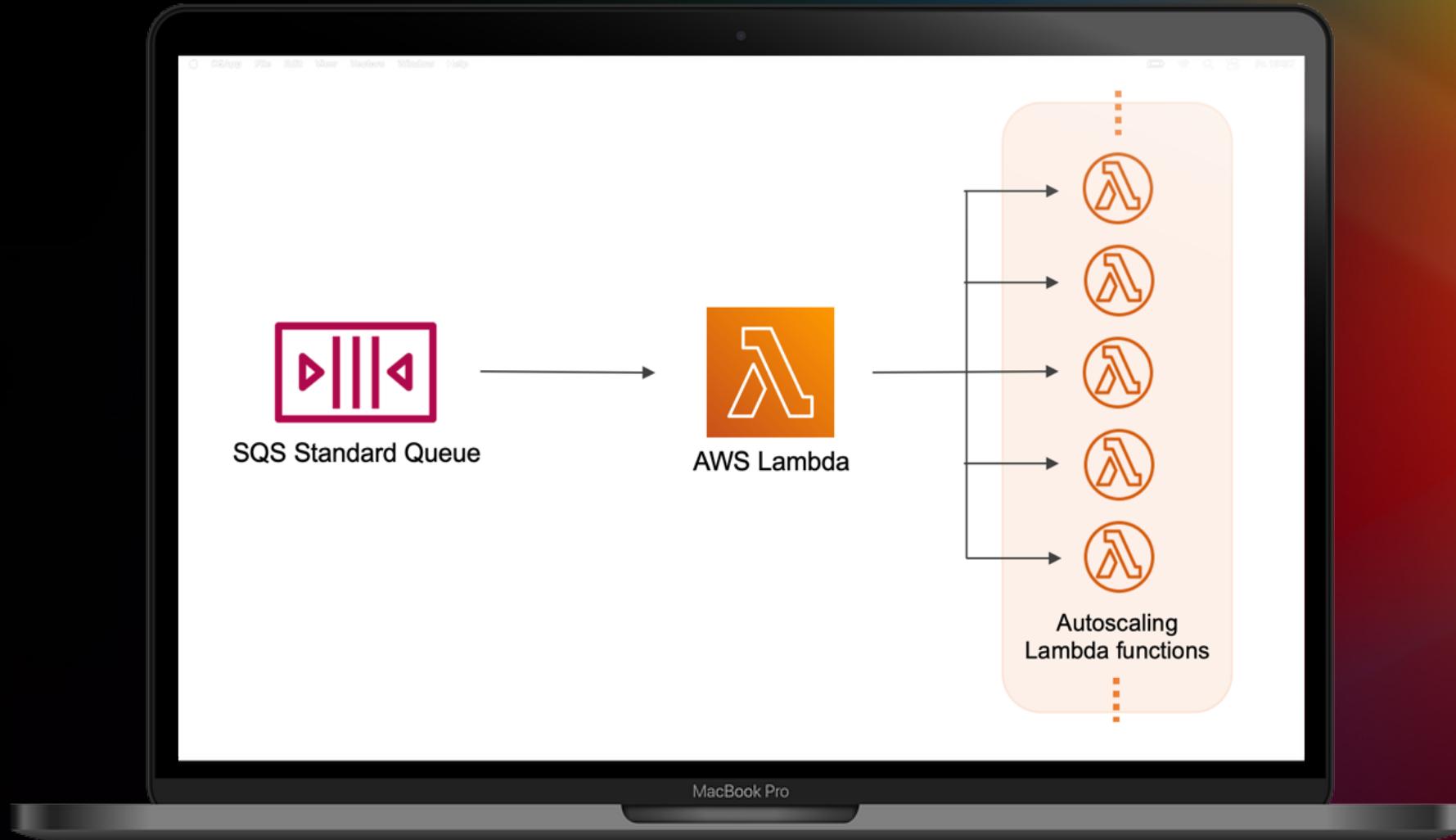
# Telemetry API



<https://aws.amazon.com/blogs/compute/introducing-the-aws-lambda-telemetry-api/>

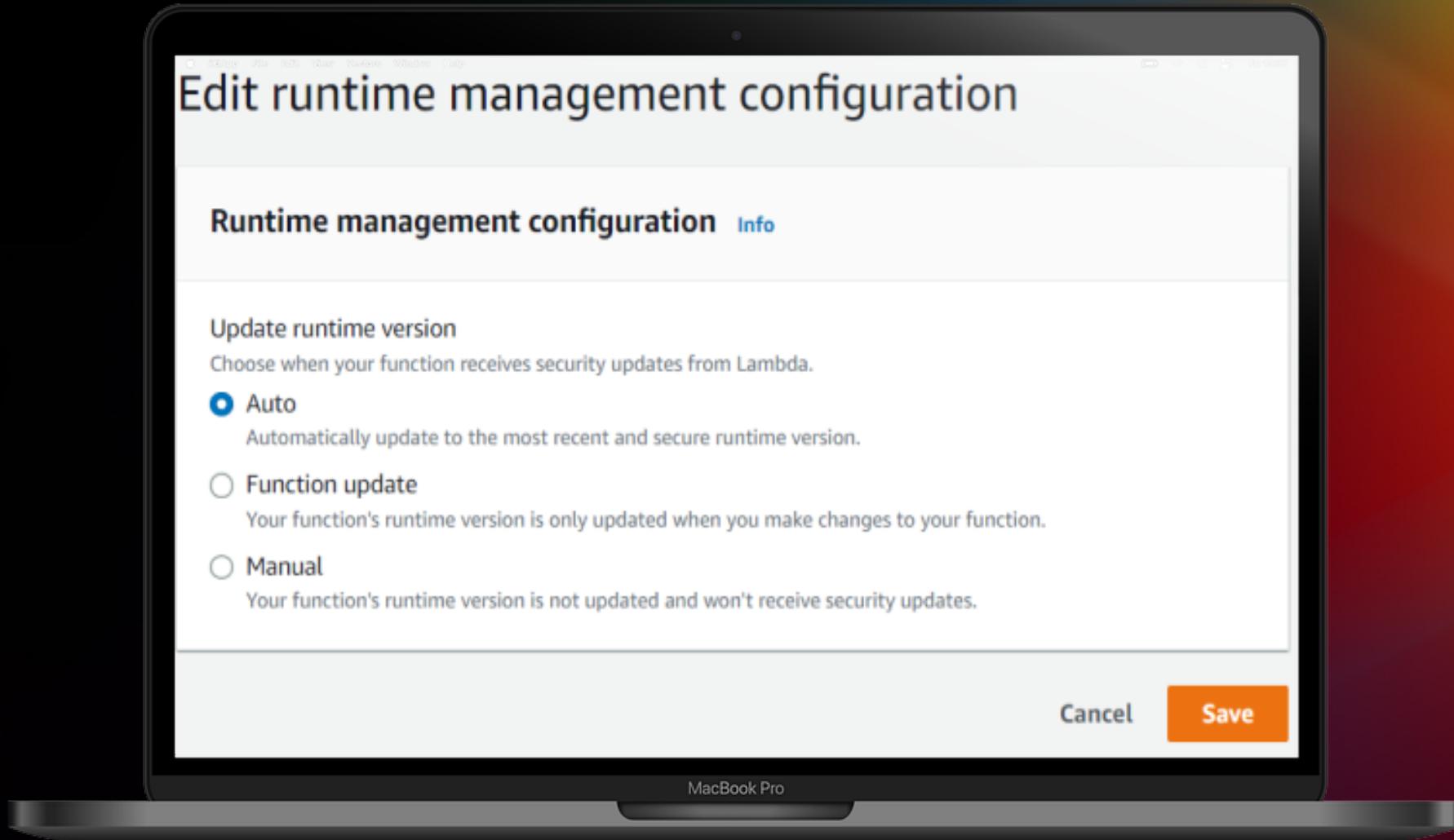


# Concurrency controls for SQS Lambda Event Source



<https://aws.amazon.com/blogs/compute/introducing-maximum-concurrency-of-aws-lambda-functions-when-using-amazon-sqs-as-an-event-source/>

# Lambda Runtime Version Management

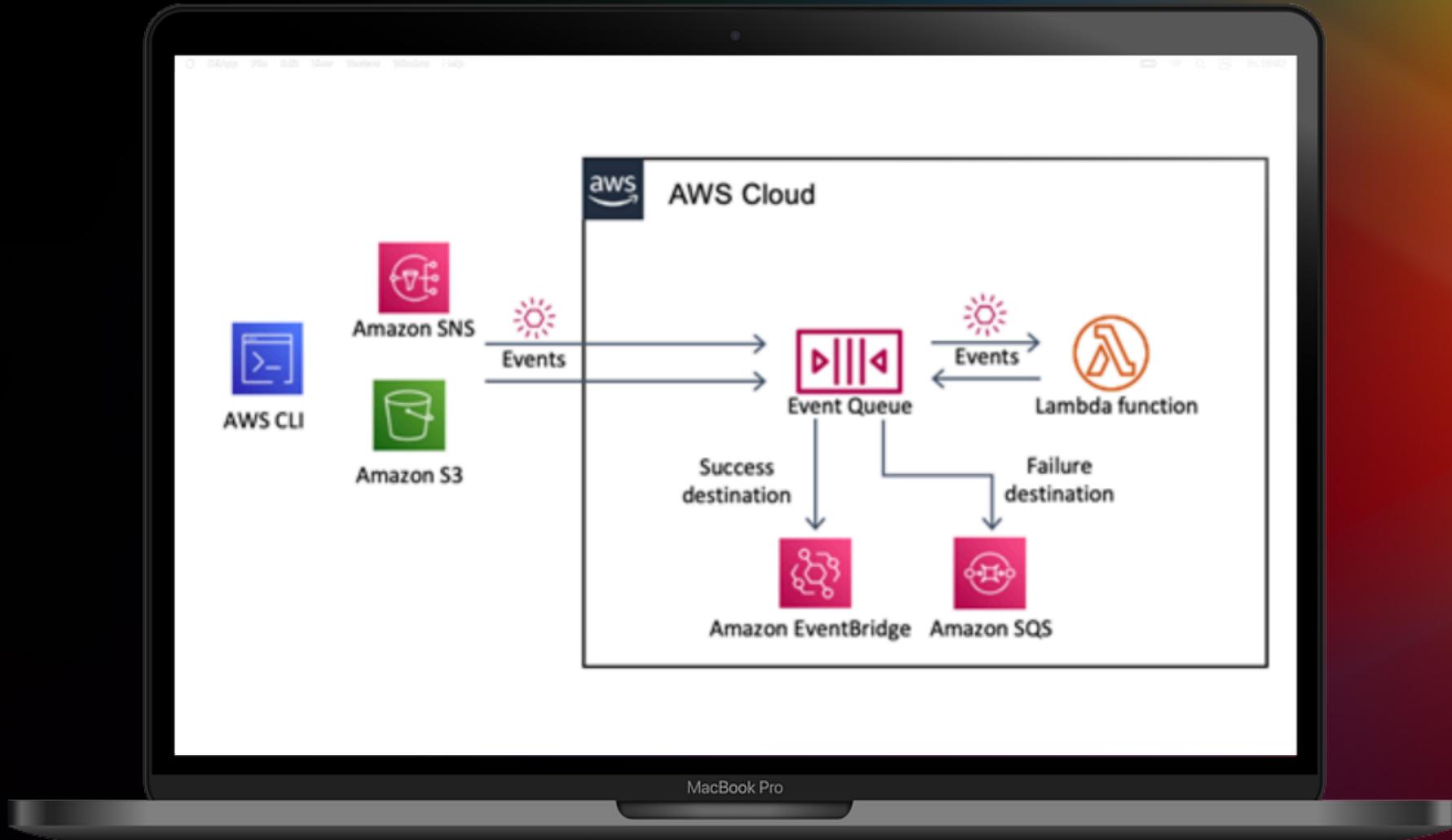


<https://aws.amazon.com/blogs/compute/introducing-aws-lambda-runtime-management-controls/>



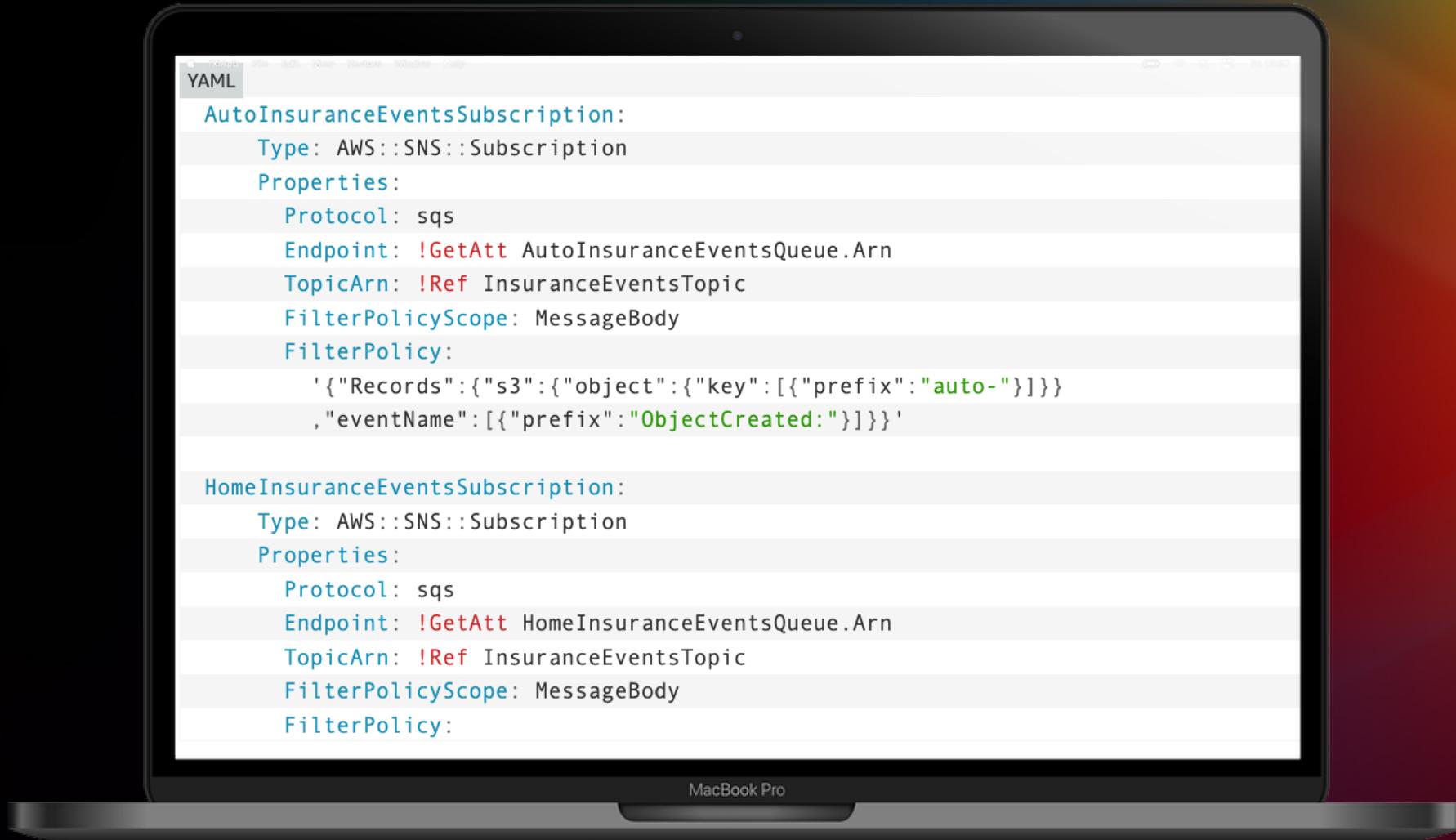
© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# Lambda Async Event Invoke Metrics



<https://aws.amazon.com/blogs/compute/introducing-new-asynchronous-invocation-metrics-for-aws-lambda/>

# Payload-based message filtering for Amazon SNS

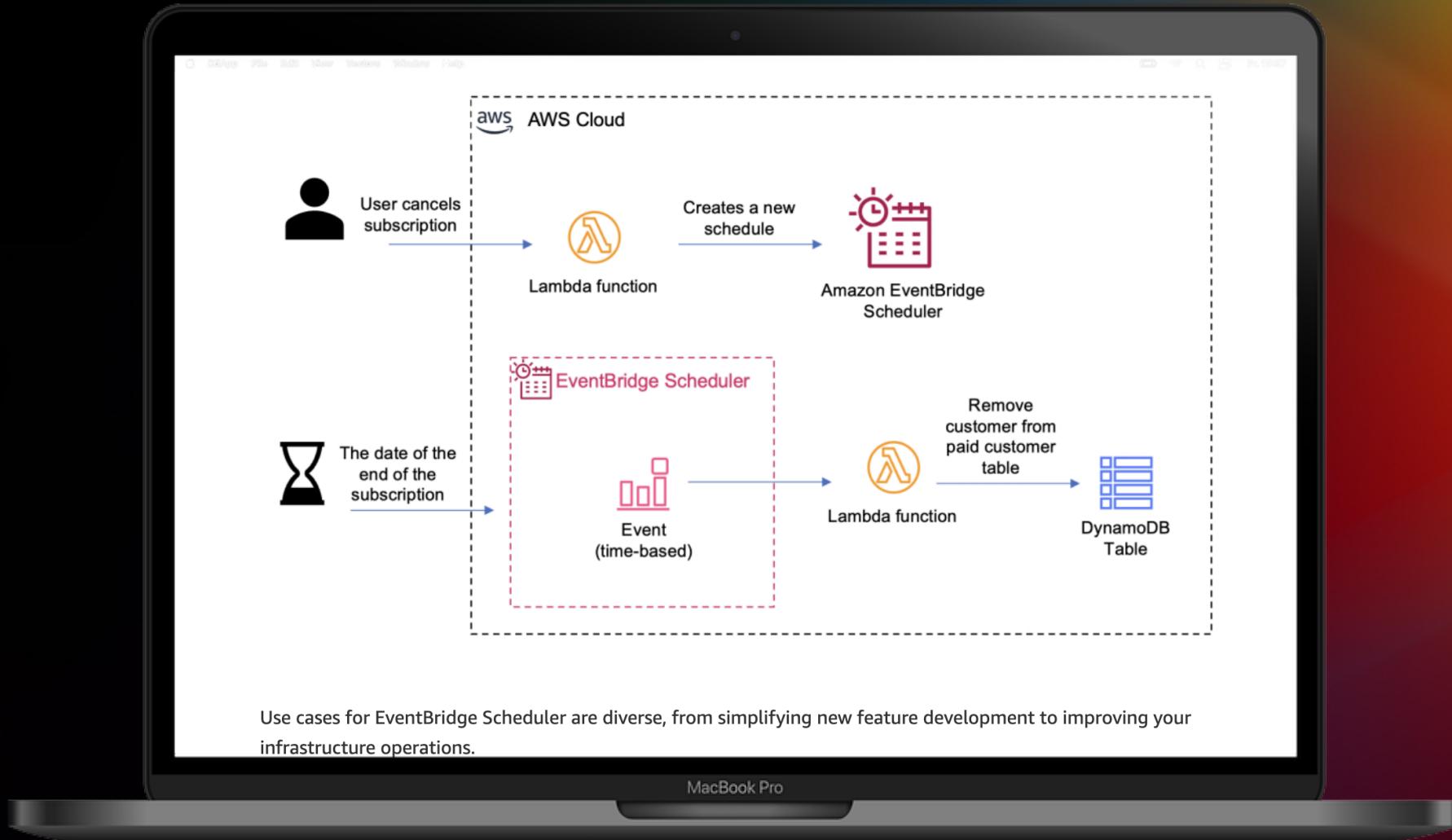


<https://aws.amazon.com/blogs/compute/introducing-payload-based-message-filtering-for-amazon-sns/>



© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# EventBridge Scheduler

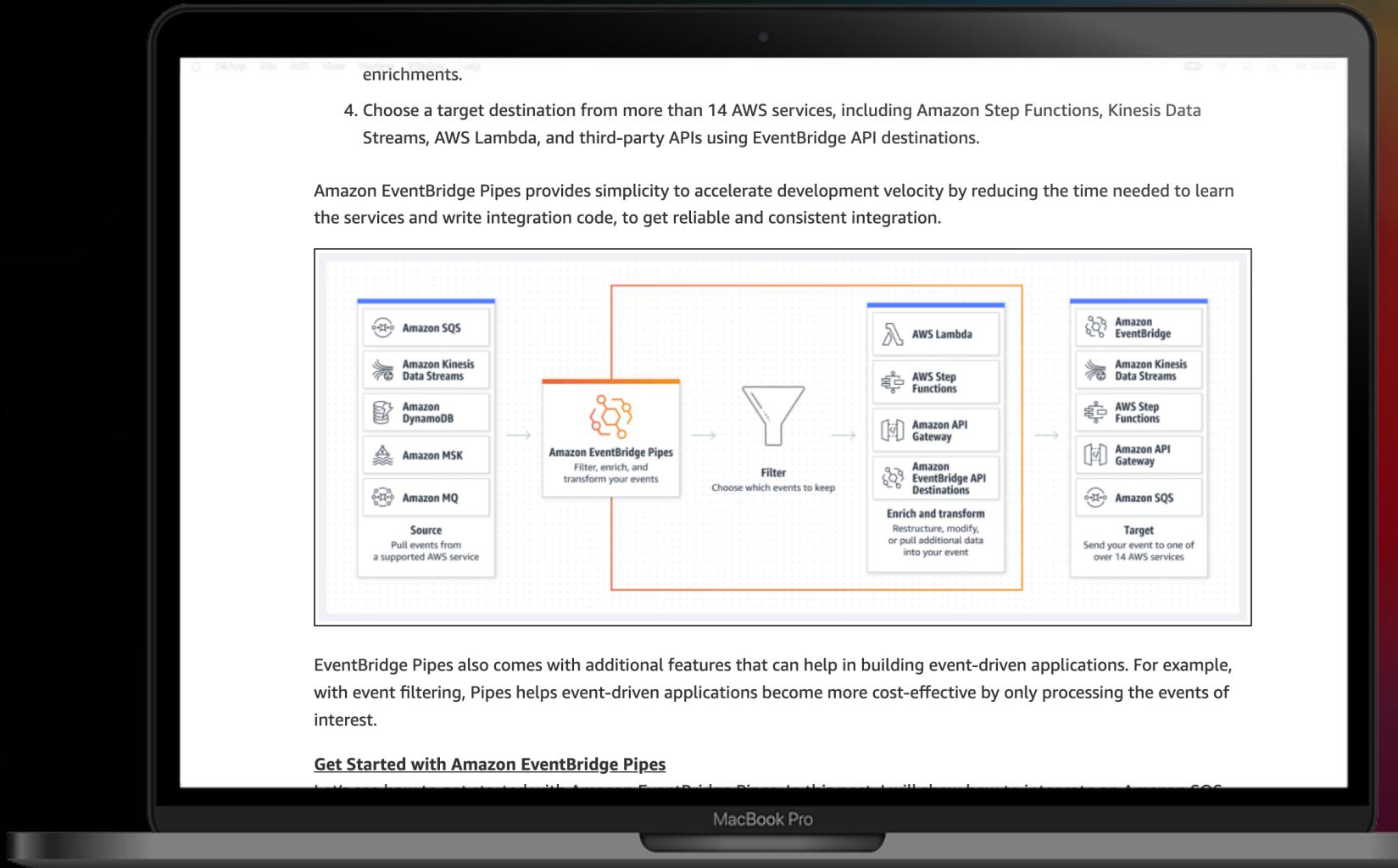


<https://aws.amazon.com/blogs/compute/introducing-amazon-eventbridge-scheduler/>



© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

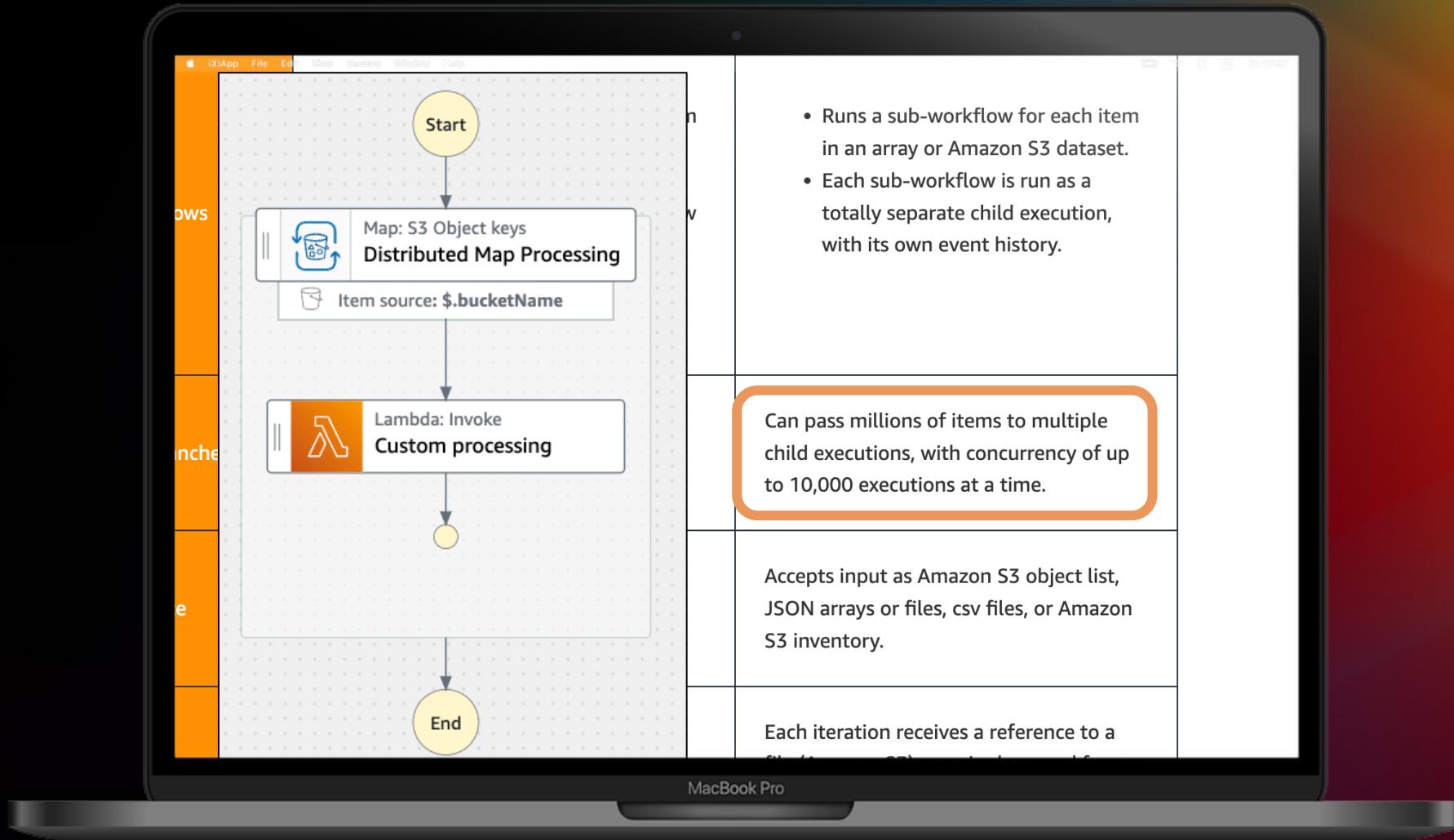
# EventBridge Pipes



<https://aws.amazon.com/blogs/aws/new-create-point-to-point-integrations-between-event-producers-and-consumers-with-amazon-eventbridge-pipes/>



# Step Functions Distributed Maps

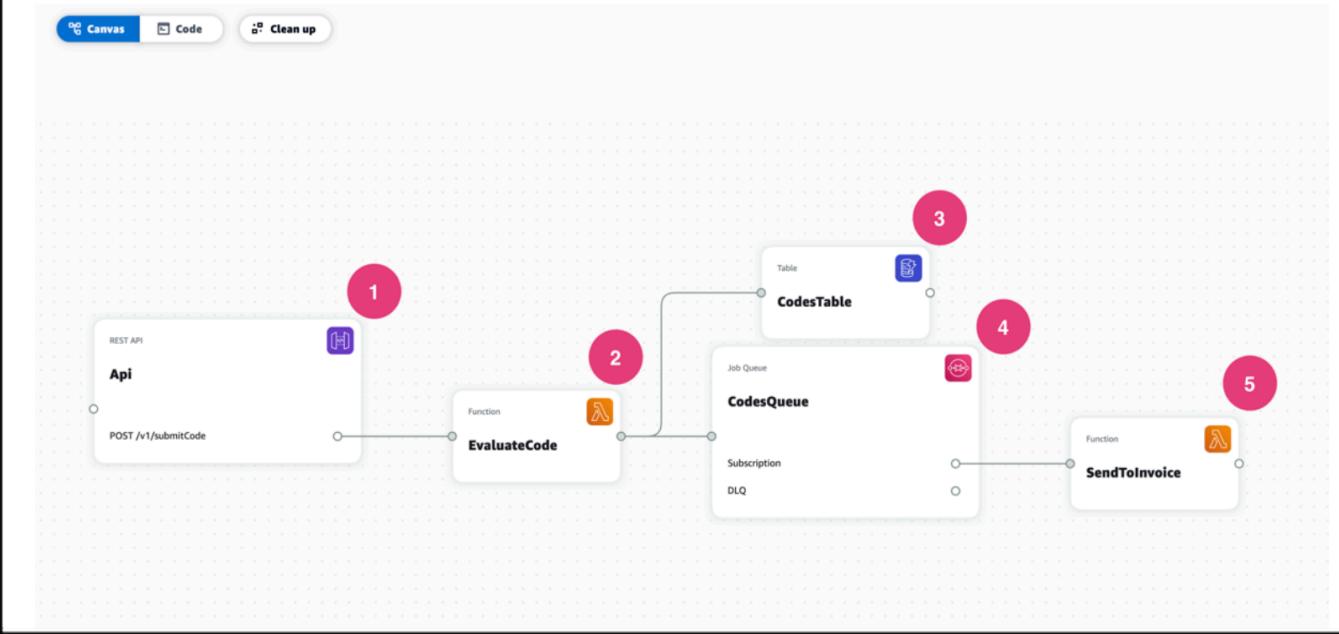


<https://aws.amazon.com/blogs/aws/step-functions-distributed-map-a-serverless-solution-for-large-scale-parallel-data-processing/>

# AWS Application Composer

- This helps new builders when designing their first serverless applications and provides an initial configuration, which more advanced builders can amend. This allows you to include good operational practices when designing a serverless application.

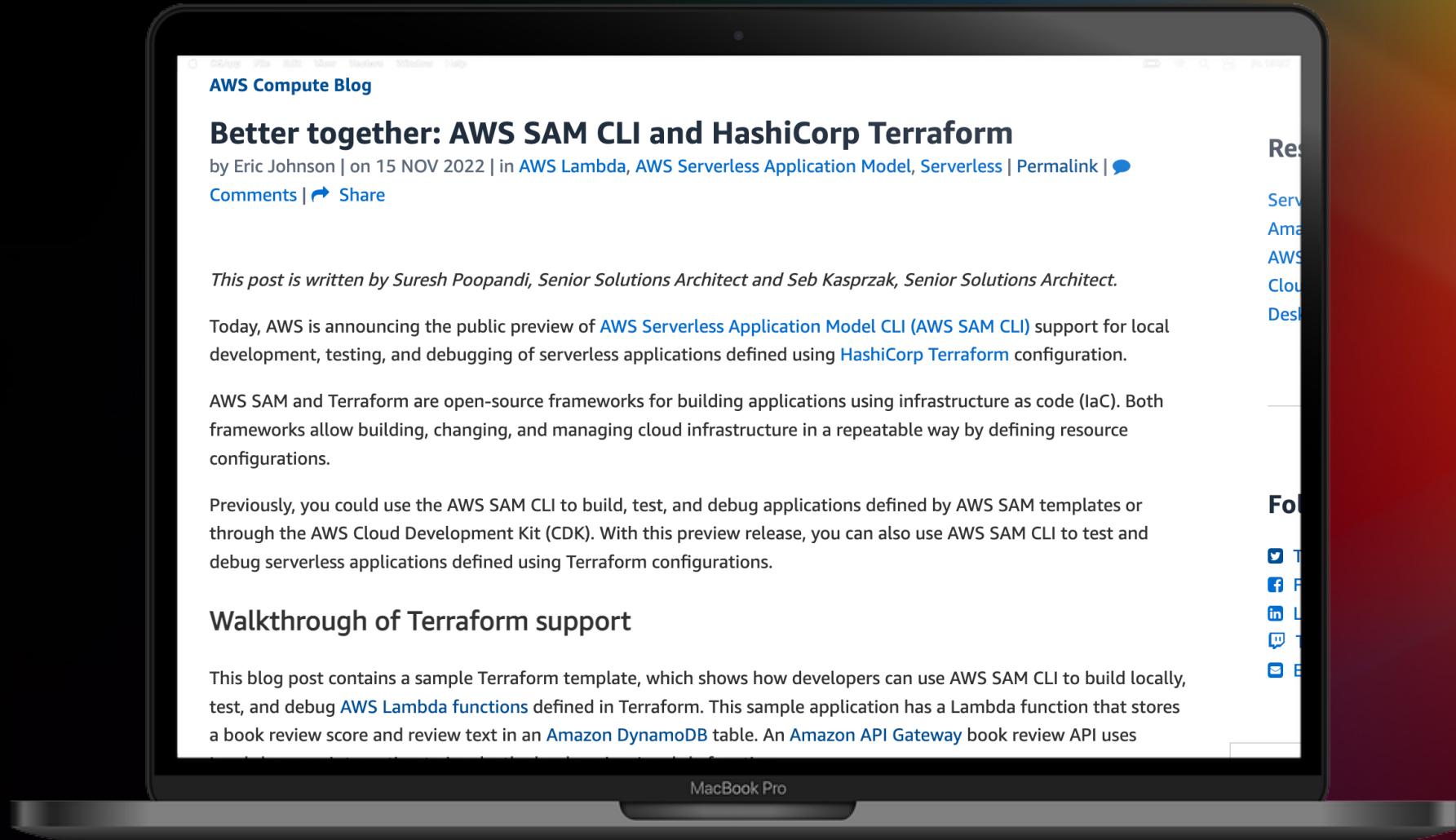
Emma's team continues to add together the different services needed to express the discount code architecture. This is the final result in Application Composer:



<https://aws.amazon.com/blogs/compute/visualize-and-create-your-serverless-workloads-with-aws-application-composer/>



# AWS SAM ❤️ Terraform (preview)



<https://aws.amazon.com/blogs/compute/better-together-aws-sam-cli-and-hashicorp-terraform/>



© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.



# Thank you!



**Anton Aleksandrov**

Principal Solutions Architect  
AWS Serverless and Event-Driven  
Architectures