


[Event dashboard](#) > [Use cases](#) > [Document extraction and summarization](#) > [Scaling with serverless workflows](#) > [Additional su...](#)

## Additional summarization techniques (Optional)

In this section, you will see two architecture patterns commonly used for summarization.

### Event-driven summarization

While this module introduced you to batch summarization, you may want to initiate the summarization process as soon as a document is available in S3. The following architectural pattern can be employed for such requirements. In this pattern, document upload events from Amazon S3 are sent to [Amazon SQS](#) . A Lambda function then consumes these messages from SQS to execute the summarization process.

You might wonder why we don't directly send S3 event notifications to Lambda. One challenge in event processing is managing the invocation rate of low-scaling systems. Although Lambda can handle bursts of events from S3, downstream services, such as Amazon Bedrock, often have lower rate limits than Lambda's burst capacity.



anyamanee ▼

[Pre-requisite] Enable foundation model access in Amazon Bedrock

[Pre-requisite] Configuring the front-end application

► Playground

▼ Use cases

► Building a RAG pipeline

▼ Document extraction and summarization

► Intelligent document processing with Generative AI

▼ Scaling with serverless workflows

than the specified number of Lambda execution environments are processing messages from SQS at any given time.

Building the workflow  
 Verifying the workflow execution  
 High level Code Walkthrough  
 Scheduling using Amazon EventBridge Scheduler (Optional)

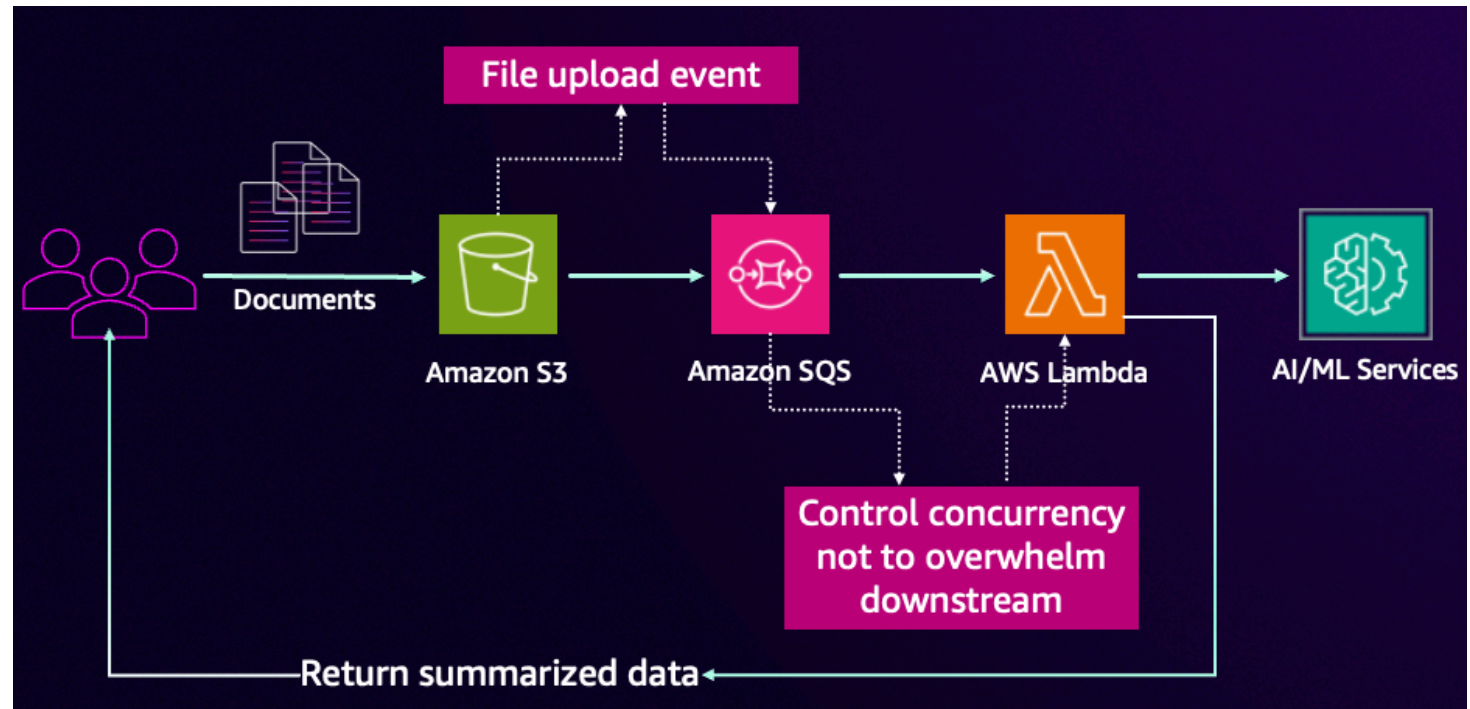
**Additional summarization techniques (Optional)**

Summary

#### ▼ AWS account access

[Open AWS console \(us-west-2\)](#)

[Get AWS CLI credentials](#)



© 2008 - 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved. [Privacy policy](#) [Terms of use](#) [Cookie preferences](#)

## Cost-effective batch summarization

1. When using distributed map for large scale summarization, if an iteration can be completed in 5 minutes, choose Express distributed map option to save cost. If it can not be completed in 5 minutes, explore batching the items in distributed map configuration so that you process the messages in batch rather than one by one. Batching will reduce the number of state transitions thereby reducing the cost. Checkout the [pricing page](#) for the difference in the pricing model for Express and Standard
2. When you are running batch summarization using custom model running in [Amazon EKS](#) or [Amazon SageMaker](#), you only need the model available during the summarization process. As part of the summarization process, you can create the endpoints, wait for the endpoint to be available, run the summarization and delete the endpoint. Below is a pattern that you can use for such use cases with Amazon SageMaker

