aws

Architecture Diagrams

Invoking Asynchronous External APIs

Invoking Asynchronous External APIs: Architecture Diagrams

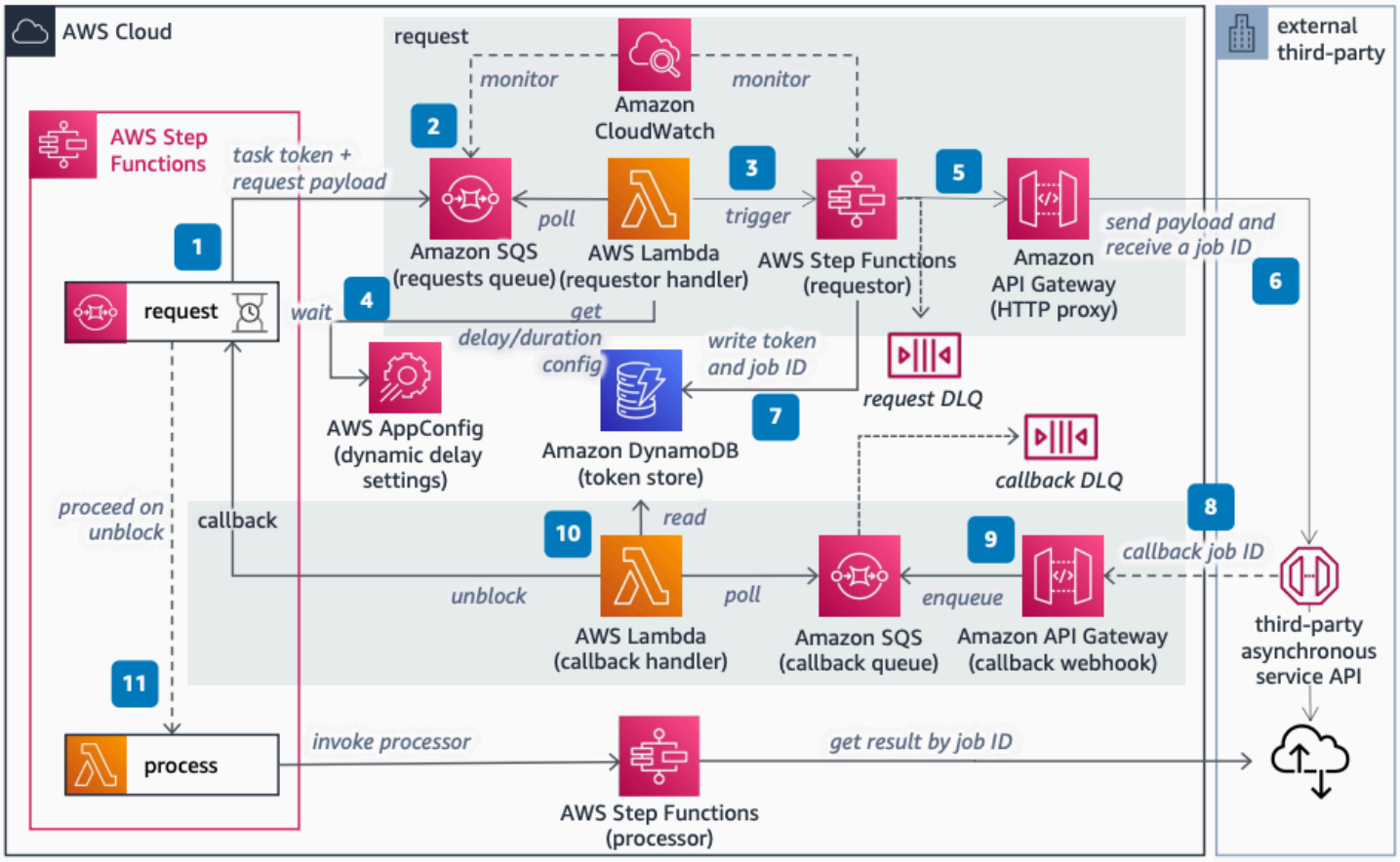
Table of Contents

Invoking Asynchronous External APIs

Publication date: **January 30, 2023 (Diagram history)**

This architecture shows how to call third-party services that implement the service callback pattern to process long-running jobs. You build an orchestration with AWS Step Functions to manage request lifecycles, throttle calls to the external service according to the maximum-requests-per-second (RPS) contract, and pause request workflows until you receive a callback notification.

Invoking Asynchronous External APIs



The following steps describe the architecture:

1. Set up Step Functions to handle the lifecycle of long-running requests to a third-party service. Add a request step that pauses the workflow, waiting for a task token to continue. Set a timeout to fail the workflow if a callback is not received.

2. Send the task token and request payload to an Amazon Simple Queue Service queue. Use Amazon CloudWatch to monitor the queue length and consider adjusting the contract if length grows beyond limits.
3. Use AWS Lambda to poll Amazon SQS and trigger an express Step Functions workflow. Control the invocation rate using polling batch size, reserved concurrency, and maximum concurrency.
4. Add a dynamic delay inside Lambda controlled by AWS AppConfig if the system needs a lower invocation rate to comply with the contracted RPS.
5. Step Functions invokes an Amazon API Gateway HTTP proxy API configured with a rate limit to comply with the contracted RPS.
6. Invoke the external third-party asynchronous service API sending the payload consumed from the requests queue, and receive the job ID. Send failed requests to the dead letter queue (DLQ) using Amazon SQS.
7. Store the workflow's task token and the received job ID in a token store implemented with Amazon DynamoDB.
8. When the external service completes, receive the completed job ID in a callback webhook endpoint implemented with API Gateway.
9. Transform external callbacks with API Gateway mapping templates, add the payload and job ID to an Amazon SQS queue, and respond immediately to the caller.
10. Use Lambda to poll the callback Amazon SQS queue, query the token store, and use the token to unblock the waiting workflow. Store failed messages in a callback DLQ.
11. On the workflow, pass the job ID to the next step and invoke a Step Functions processor to fetch results from the external third-party results store by job ID. Process the external service's results.

Further reading

For additional information, refer to the following resources:

- AWS Architecture Icons
- AWS Architecture Center
- AWS Well-Architected

Diagram history

To be notified about updates to this reference architecture diagram, subscribe to the RSS feed.

Change	Description	Date
Initial publication	Reference architecture diagram first published.	January 30, 2023

Note

To subscribe to RSS updates, you must have an RSS plugin enabled for the browser you are using.