Best Practices to Reduce Latency on your Smart Home Skill

Michael Williams Nov 20, 2020

Smart Home Skills | Tips & Tools | Smart Home API

When it comes to your Alexa skill, speed matters. When customers use your skill to control their smart home devices, they expect her to respond quickly. If your skill has high latency, this can lead to a loss of customer engagement and low ratings. Below are 9 recommendations to help improve the performance of your smart home skill.

Low Hanging Fruit

1. Adjust your Skill Lambda function memory allocation to a minimum of 512MB RAM:
Allocating more memory (MB) will provide more CPU power to your Lambda function and will execute faster. For smart home skills, we recommend you to allocate at least 512MB RAM to your Skill Lambda function.

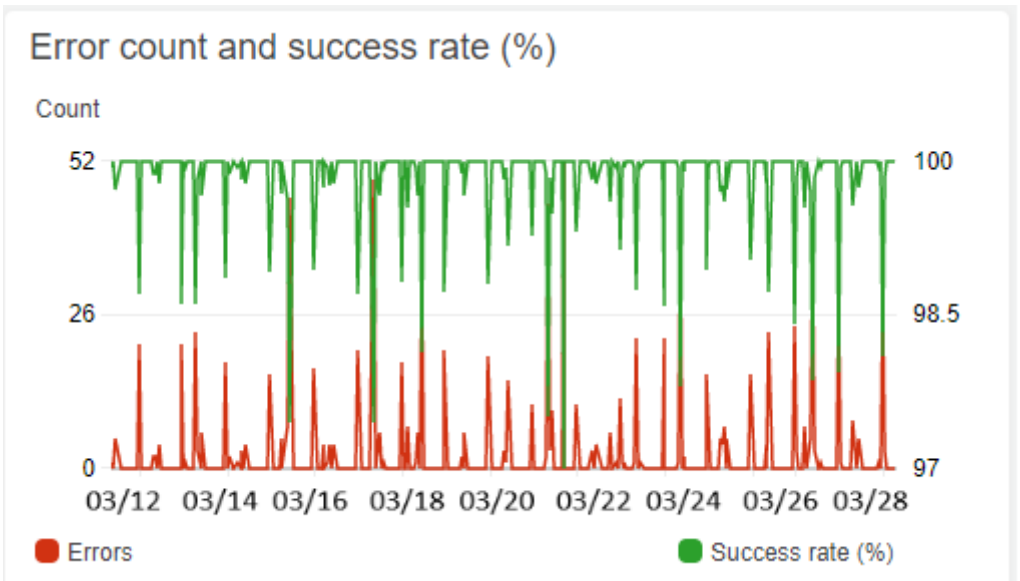
To further optimize your memory allocation, you can:

a) Analyze the max memory used on your skill: You can analyze the memory used by your Alexa skill in your AWS CloudWatch Logs. By analyzing the "Max Memory Used" field, you can determine if your function needs more memory or if you over-provisioned your function's memory size. On each skill invocation, a "REPORT" entry will be made to your CloudWatch logs, as shown below:

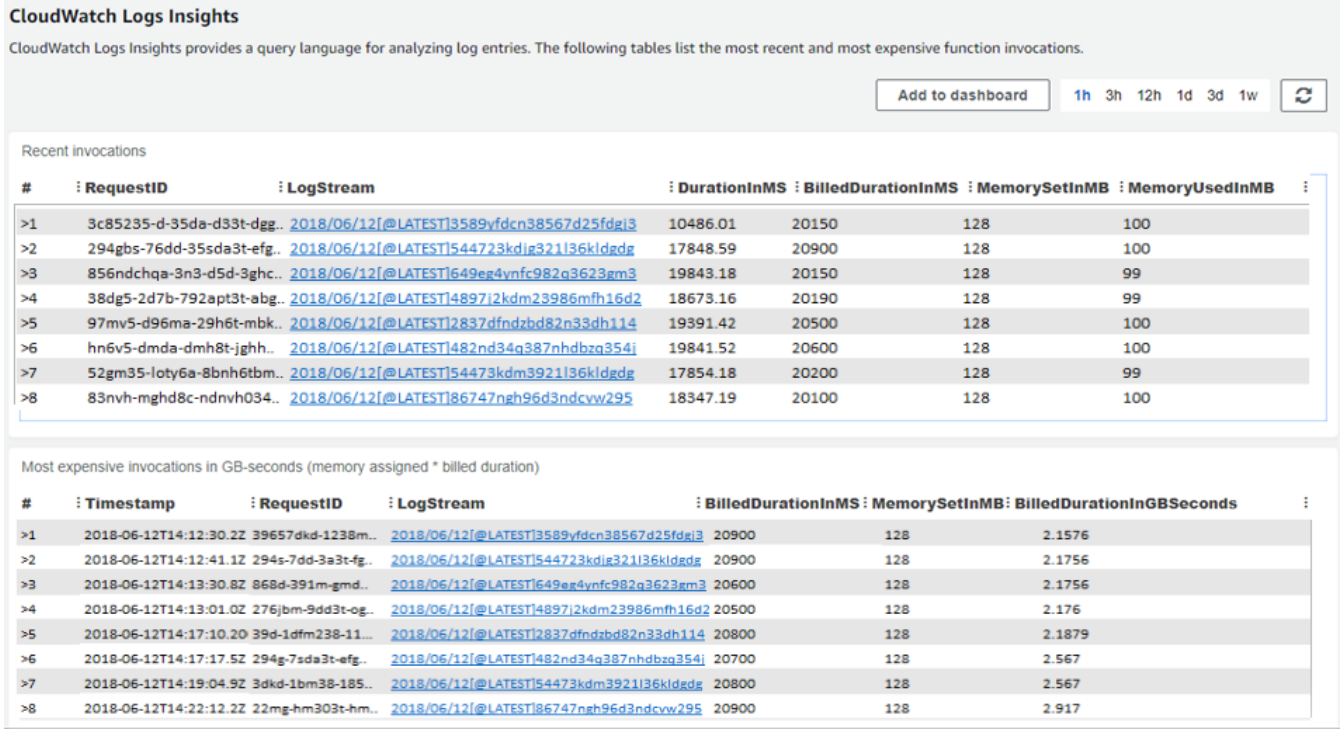
REPORT RequestId: 3604209a-e9a3-11e6-939a-754dd98c7be3 Duration: 12.34 ms Billed Duration: 100 ms Memory Size: 128 MB Max Memory Used: 18 MB

b) Load test your Lambda function: To determine an optimum timeout and memory values, it is important to analyze how long your function runs. This will allow you to discover any dependency issues your function may be running into while waiting for a response from a service, which may be increasing the concurrency of the function beyond what you expect. This is especially important

2. Identify and fix control errors that are causing high latency: Using CloudWatch metrics you can dig deeper into lambda requests that are failing. If a lambda request fails, this can cause Alexa to wait 8 seconds before Alexa responds back to customer. This not only affects your overall skill latency but also causes slow response time to customers. To see a breakdown of your success and failures, select the **Monitoring** tab inside your lambda window:



You can identify at particular time intervals where requests are failing, and then dig even further through CloudWatch logs to see why these failed.



3. **<u>Ensure no synchronous database calls are done before response to Alexa</u>:** For example, avoid updating databases before responding to Alexa as both should be done asynchronously. If these calls are left synchronous prior to responding to Alexa, it will delay response to Alexa, and in turn the customer.

4. **<u>Avoid cross-region calls</u>:** We recommend placement of Lambda functions and media services in additional regions based on geographic distances contributing to latency and regions supported by the Alexa Service. At the very minimum, if you serve in NA & EU, your AWS should support 3 regions: N. Virginia (us-east-1), Dublin (eu-west-1), Oregon (us-west-2). The list of all supported regions for Lambda can be found on the AWS website. Distance between services is a contributing factor for latency, it is important to keep http calls within region to reduce latency.

5. **<u>Optimize Amazon Simple Queue Service values:</u>** If you are using Amazon Simple Queue Service (SQS) as an event source, make sure the value of the function's expected execution time does not exceed the Visibility Timeout value on the queue. This applies both to CreateFunction and UpdateFunctionConfiguration. AWS terminates every function exceeding its configured timeout so utterances will fail incorrectly.

6. **<u>Implement Wake-On-Lan Controller for AV devices:</u>** Standby mode for AV devices such as TVs can take a long time to wake up the device before invoking a command which causes latency. Implementing Wake-On-LAN controller will power on the devices in low power mode and when an Alexa request is sent to you, you can decide whether you want to wake up the device using your cloud or you want Alexa to execute the request using the WoL protocol. Read how to implement WakeOnLan Controller here.

Addressing Lambda Latency

Inside every Alexa Smart Home architecture, the AWS lambda instance can be another source of latency and control errors. The following recommendations can help reduce the latency caused by your Lambda.

7. Take advantage of execution context reuse to improve the performance of your function:
Starting-up the Execution Context can take some time but is inevitable before running the function for the first time, this is known as the "cold start". The very first time your Lambda function is invoked, you have to wait for "bootstrapping" to complete.

Initialize SDK clients and database connections outside of the function handler, and cache static assets locally in the /tmp directory. Subsequent invocations processed by the same instance of your function can reuse these resources. This saves execution time and cost. To avoid potential data leaks across invocations, don't use the execution context to store user data, events, or other information with security implications. If your function relies on a mutable state that can't be stored in memory within the handler, consider creating a separate function or separate versions of a function for each user. You can learn more about lambda best practices here.

8. Minimize the complexity of your dependencies: Simpler frameworks will load more quickly on the Execution Context startup. You can simplify your Lambda by:

- Using Java dependency injection (IoC) frameworks like Dagger or Guice, over more complex ones like Spring Framework.
- Ensuring you are using most current versions of packages.
- Packaging your function into a JAR which will result in a function of a few KB's.
- Deleting unused lambdas to avoid dead code inflating deployment package size.

9. Ensure that your Lambda has no recursive code: Recursive code can lead to unintended volume of function invocations and escalated costs.

Get Started

Go to the Alexa Skills Kit console to manage your smart home skill.