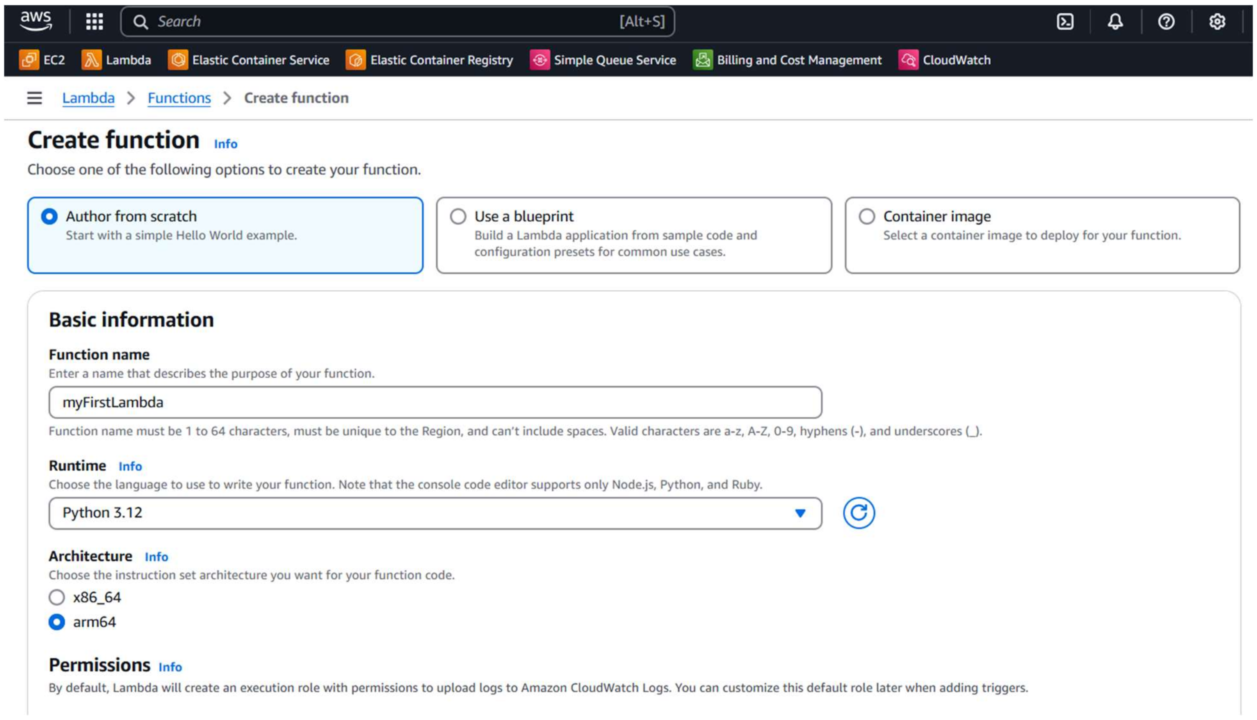
Report on AWS Lambda and EventBridge Rule Implementation

Introduction

This report outlines the steps taken to implement an AWS Lambda function triggered by an EventBridge rule that runs every minute. The function prints event details and is monitored for execution metrics.

Step 1: Creating the AWS Lambda Function

1. Navigate to the **AWS Lambda** service in the AWS Management Console.
2. Click on **Create function**.
3. Select **Author from scratch**.
4. Provide the function name: `myFirstLambda`.
5. Choose **Python 3.12** as the runtime.
6. Click **Create function**.


aws
Search [Alt+S]
EC2
Lambda
Elastic Container Service
Elastic Container Registry
Simple Queue Service
Billing and Cost Management
CloudWatch
Lambda > Functions > Create function

Create function Info

Choose one of the following options to create your function.

[✓ Author from scratch] Start with a simple Hello World example.

Use a blueprint: Build a Lambda application from sample code and configuration presets for common use cases.

Container image: Select a container image to deploy for your function.

Basic information

Function name: Enter a name that describes the purpose of your function.

myFirstLambda

Function name must be 1 to 64 characters, must be unique to the Region, and can't include spaces. Valid characters are a-z, A-Z, 0-9, hyphens (-), and underscores (_).

Runtime Info: Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.

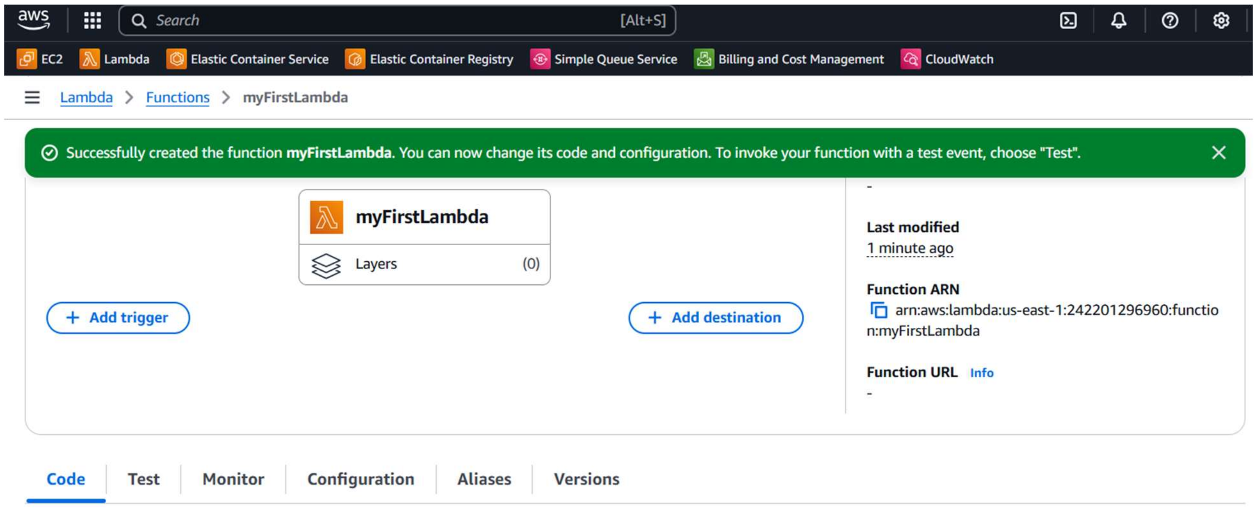
Python 3.12

Architecture Info: Choose the instruction set architecture you want for your function code.

x86_64 [✓ arm64]

Permissions Info

By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.


aws
Search
[Alt+S]
EC2
Lambda
Elastic Container Service
Elastic Container Registry
Simple Queue Service
Billing and Cost Management
CloudWatch
Lambda > Functions > myFirstLambda

Successfully created the function **myFirstLambda**. You can now change its code and configuration. To invoke your function with a test event, choose "Test".

Last modified
1 minute ago

Function ARN
arn:aws:lambda:us-east-1:242201296960:function:myFirstLambda

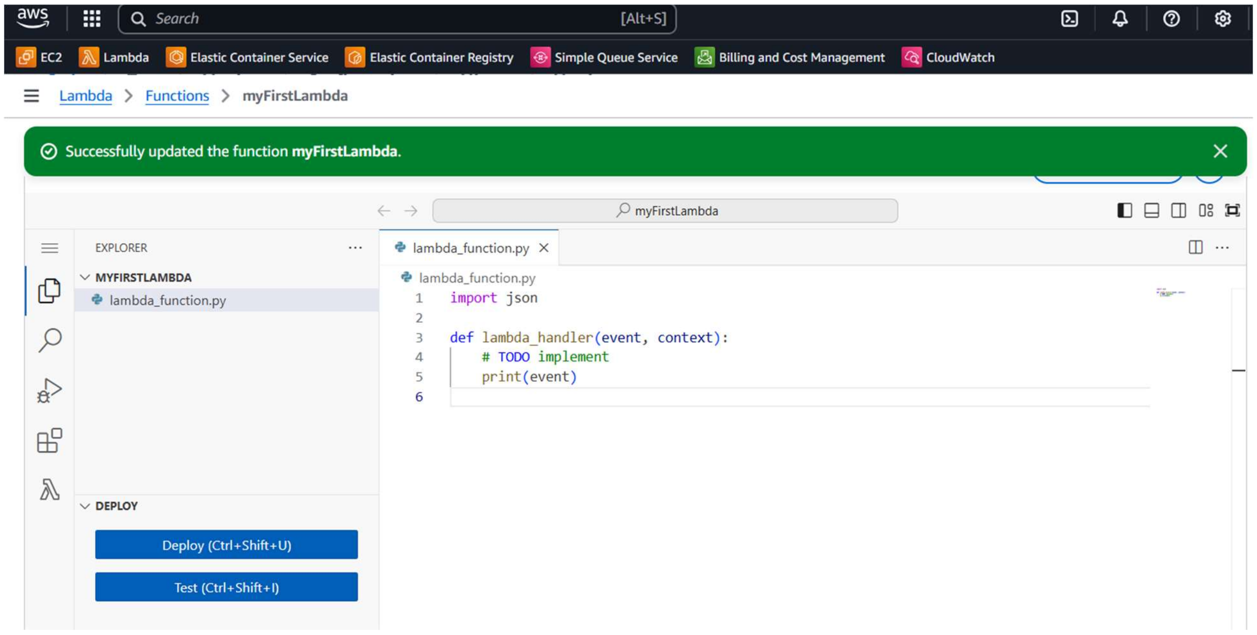
Function URL Info
-

Step 2: Writing the Python Code

1. In the Lambda function editor, replace the default code with the following Python script:

```
def lambda_handler(event, context):
   print(event)
```

2. Click **Deploy** to save the function.


aws
Search
[Alt+S]
EC2
Lambda
Elastic Container Service
Elastic Container Registry
Simple Queue Service
Billing and Cost Management
CloudWatch
Lambda > Functions > myFirstLambda

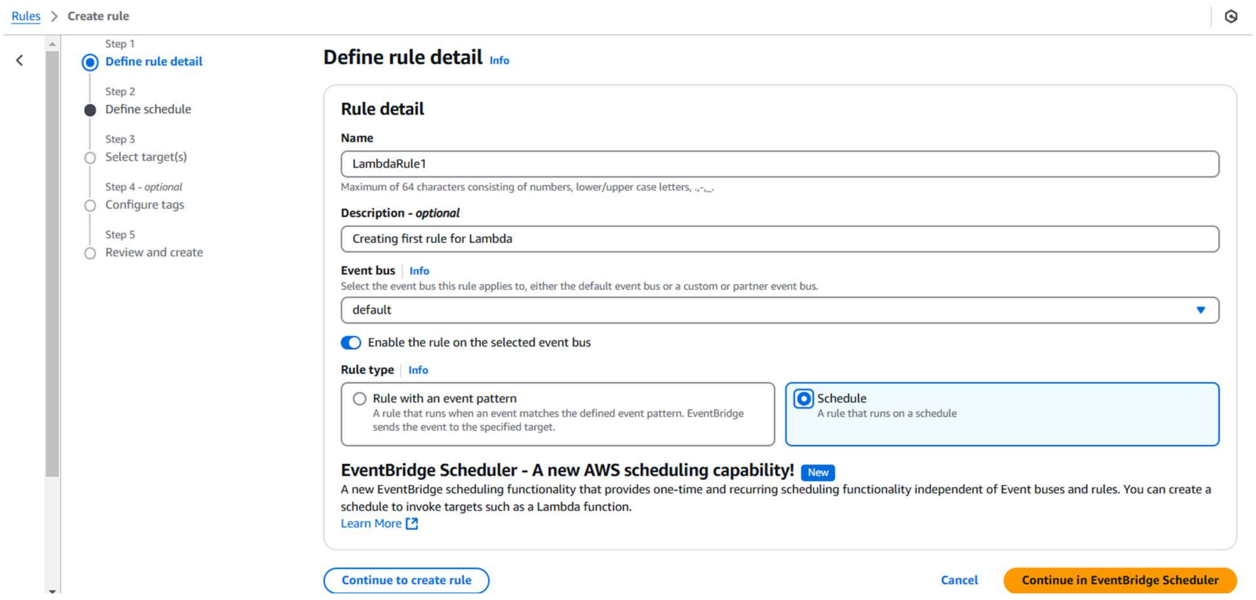
Successfully updated the function **myFirstLambda**.

```
import json

def lambda_handler(event, context):
    # TODO implement
    print(event)
```

Step 3: Creating an EventBridge Rule

1. Navigate to the **Amazon EventBridge** service.
2. Click on **Rules** in the sidebar.
3. Click **Create rule**.
4. Enter the rule name: `LambdaRule1`.
5. Choose **Schedule** as the rule type.
6. In the **Schedule pattern**, enter the cron expression: `* * * * ? *` (runs every minute).



Rules > Create rule

Step 1
Define rule detail

Step 2
Define schedule

Step 3
Select target(s)

Step 4 - *optional*
Configure tags

Step 5
Review and create

Define rule detail Info

Rule detail

Name
LambdaRule1
Maximum of 64 characters consisting of numbers, lower/upper case letters, .,-,_.

Description - *optional*
Creating first rule for Lambda

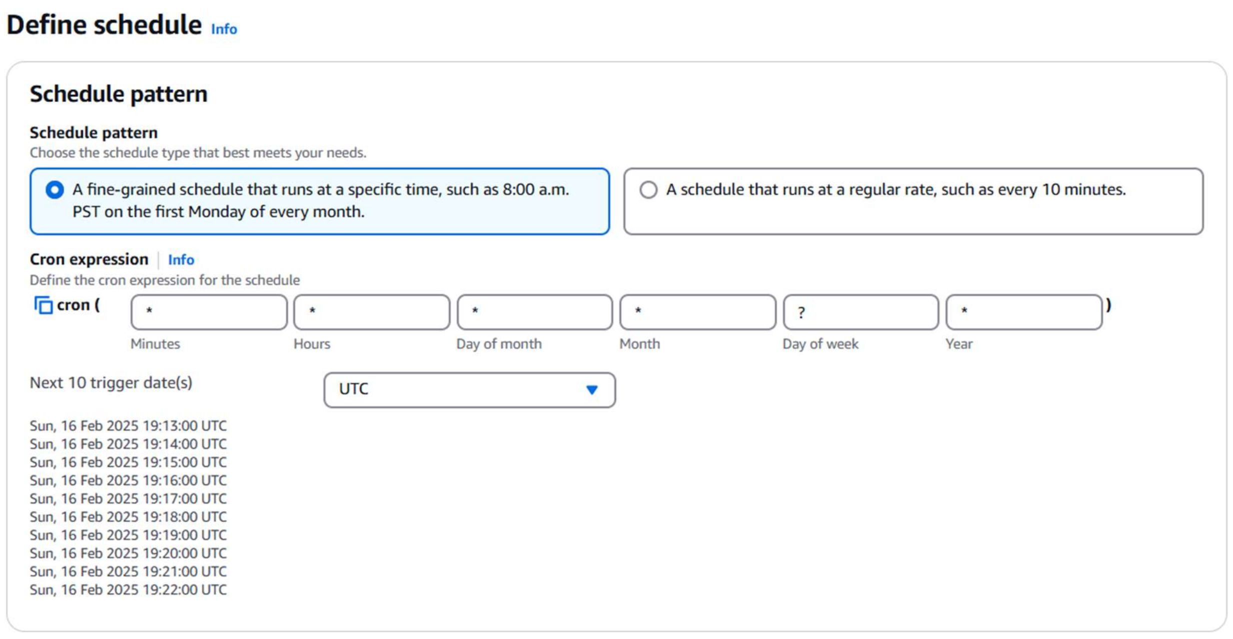
Event bus | Info
Select the event bus this rule applies to, either the default event bus or a custom or partner event bus.
default

Enable the rule on the selected event bus

Rule type | Info
Rule with an event pattern — A rule that runs when an event matches the defined event pattern. EventBridge sends the event to the specified target.
[✓ Schedule — A rule that runs on a schedule]

EventBridge Scheduler - A new AWS scheduling capability! New
A new EventBridge scheduling functionality that provides one-time and recurring scheduling functionality independent of Event buses and rules. You can create a schedule to invoke targets such as a Lambda function.
Learn More

Continue to create rule | Cancel | Continue in EventBridge Scheduler



Define schedule Info

Schedule pattern

Schedule pattern
Choose the schedule type that best meets your needs.
[✓ A fine-grained schedule that runs at a specific time, such as 8:00 a.m. PST on the first Monday of every month.]
A schedule that runs at a regular rate, such as every 10 minutes.

Cron expression | Info
Define the cron expression for the schedule

Minutes	Hours	Day of month	Month	Day of week	Year
*	*	*	*	?	*

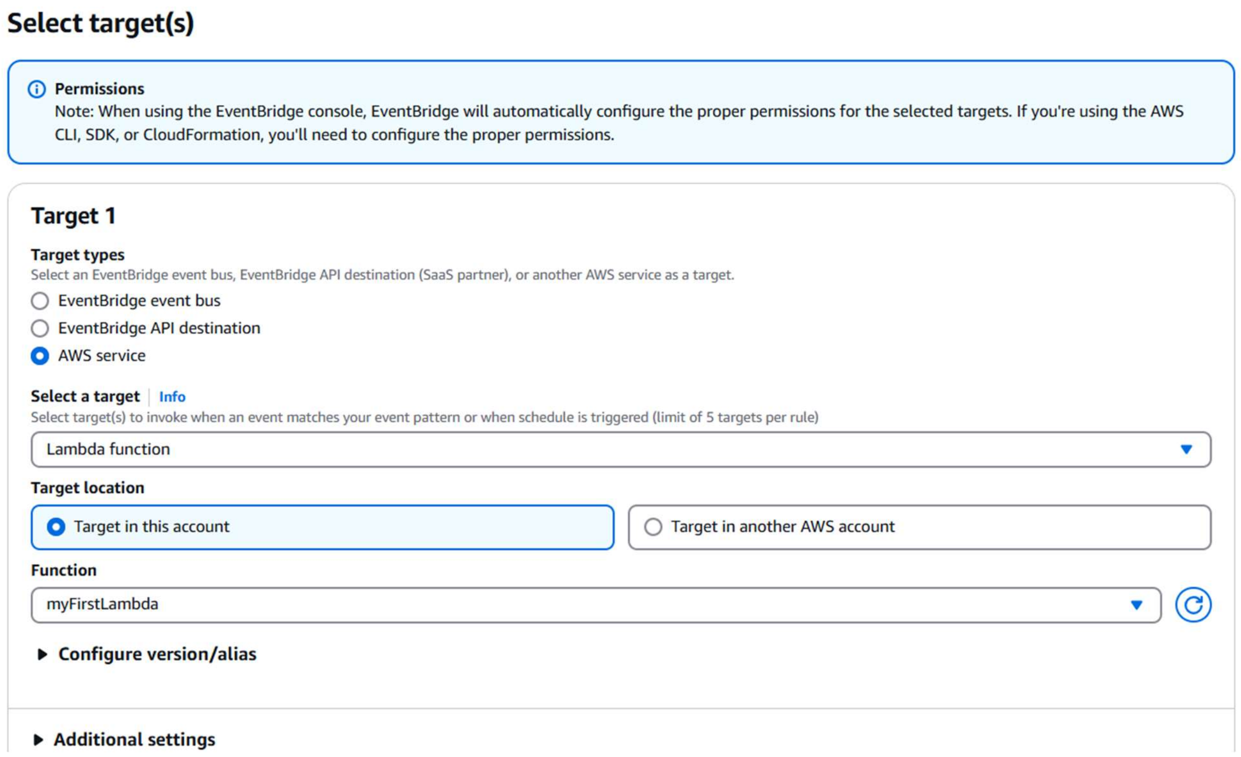
cron (* * * * ? *)

Next 10 trigger date(s) — UTC

```
Sun, 16 Feb 2025 19:13:00 UTC
Sun, 16 Feb 2025 19:14:00 UTC
Sun, 16 Feb 2025 19:15:00 UTC
Sun, 16 Feb 2025 19:16:00 UTC
Sun, 16 Feb 2025 19:17:00 UTC
Sun, 16 Feb 2025 19:18:00 UTC
Sun, 16 Feb 2025 19:19:00 UTC
Sun, 16 Feb 2025 19:20:00 UTC
Sun, 16 Feb 2025 19:21:00 UTC
Sun, 16 Feb 2025 19:22:00 UTC
```

Step 4: Adding the Lambda Function as the Target

1. In the **Target** section, select **AWS Lambda function**.
2. Choose `myFirstLambda` from the dropdown.
3. Click **Create rule**.



Select target(s)

Permissions
Note: When using the EventBridge console, EventBridge will automatically configure the proper permissions for the selected targets. If you're using the AWS CLI, SDK, or CloudFormation, you'll need to configure the proper permissions.

Target 1

Target types
Select an EventBridge event bus, EventBridge API destination (SaaS partner), or another AWS service as a target.

EventBridge event bus
EventBridge API destination
[✓ AWS service]

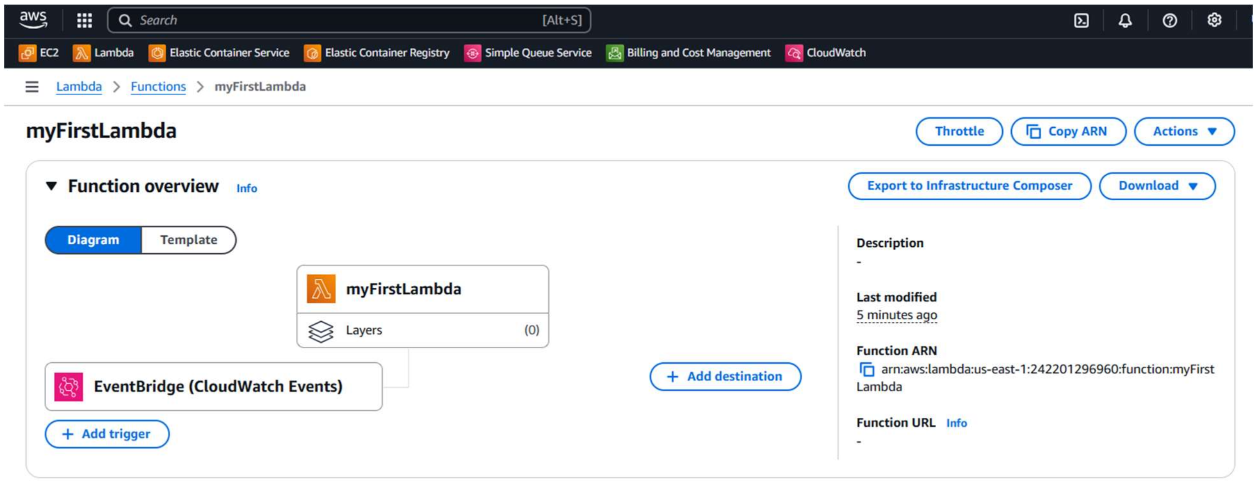
Select a target Info
Select target(s) to invoke when an event matches your event pattern or when schedule is triggered (limit of 5 targets per rule)
Lambda function

Target location
[✓ Target in this account] Target in another AWS account

Function
myFirstLambda

▶ Configure version/alias

▶ Additional settings



aws
Search [Alt+S]
EC2
Lambda
Elastic Container Service
Elastic Container Registry
Simple Queue Service
Billing and Cost Management
CloudWatch
Lambda > Functions > myFirstLambda

myFirstLambda

Throttle
Copy ARN
Actions
Export to Infrastructure Composer
Download
Diagram
Template
Layers (0)
Add destination
Add trigger

▼ **Function overview** Info

myFirstLambda
EventBridge (CloudWatch Events)

Description
-

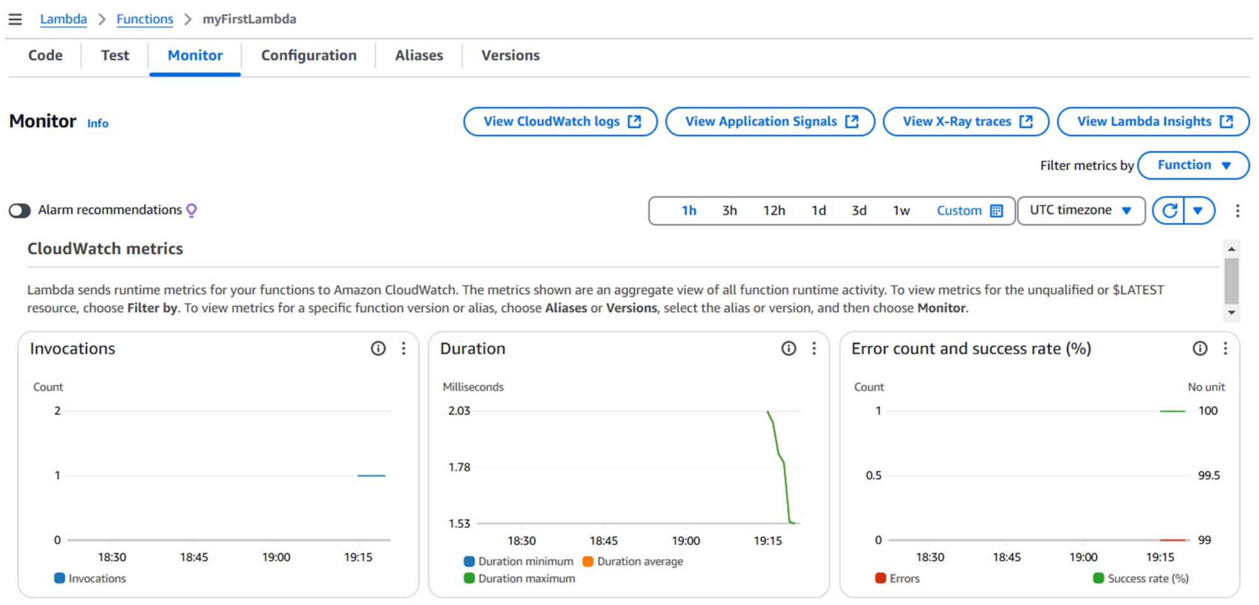
Last modified
5 minutes ago

Function ARN
arn:aws:lambda:us-east-1:242201296960:function:myFirstLambda

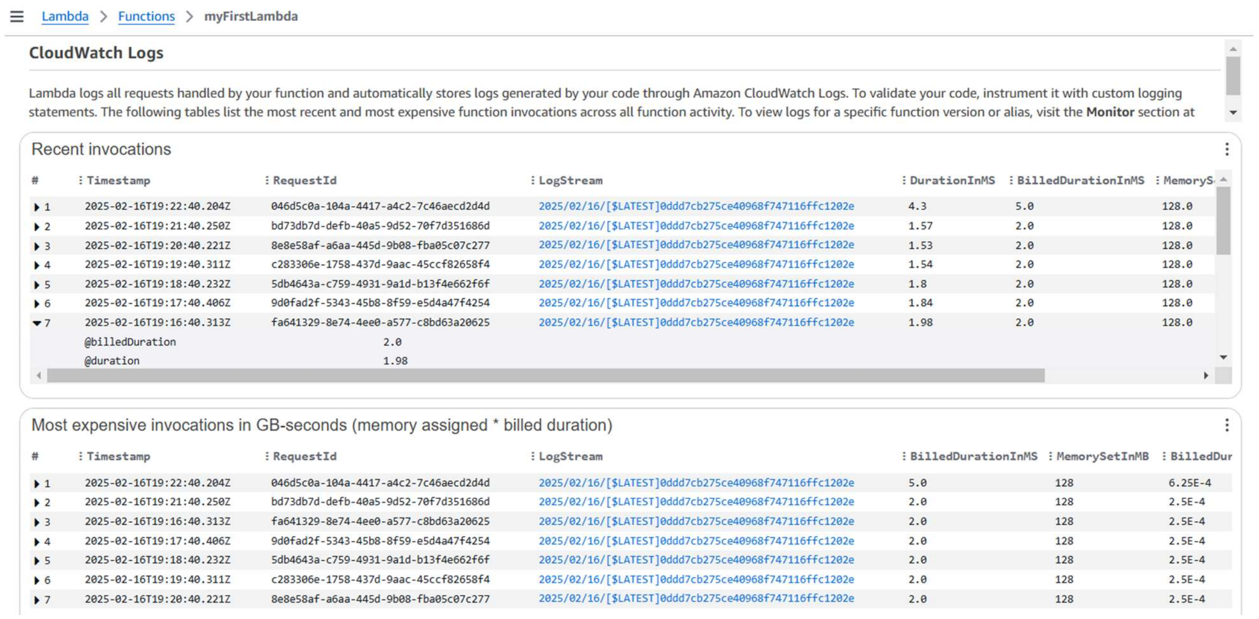
Function URL Info
-

Step 5: Monitoring the Execution

1. Navigate to **AWS CloudWatch**.
2. Open the **Logs** section and find `myFirstLambda`.
3. Check the logs for event execution details.
4. Open **Metrics** and verify the invocation count.



Lambda sends runtime metrics for your functions to Amazon CloudWatch. The metrics shown are an aggregate view of all function runtime activity. To view metrics for the unqualified or $LATEST resource, choose **Filter by**. To view metrics for a specific function version or alias, choose **Aliases** or **Versions**, select the alias or version, and then choose **Monitor**.



Lambda logs all requests handled by your function and automatically stores logs generated by your code through Amazon CloudWatch Logs. To validate your code, instrument it with custom logging statements. The following tables list the most recent and most expensive function invocations across all function activity. To view logs for a specific function version or alias, visit the **Monitor** section at

Recent invocations

#	Timestamp	RequestId	LogStream	DurationInMS	BilledDurationInMS	MemoryS
▸ 1	2025-02-16T19:22:40.204Z	046d5c0a-104a-4417-a4c2-7c46aecd2d4d	2025/02/16/[$LATEST]0ddd7cb275ce40968f747116ffc1202e	4.3	5.0	128.0
▸ 2	2025-02-16T19:21:40.250Z	bd73db7d-defb-40a5-9d52-70f7d351686d	2025/02/16/[$LATEST]0ddd7cb275ce40968f747116ffc1202e	1.57	2.0	128.0
▸ 3	2025-02-16T19:20:40.221Z	8e8e58af-a6aa-445d-9b08-fba05c07c277	2025/02/16/[$LATEST]0ddd7cb275ce40968f747116ffc1202e	1.53	2.0	128.0
▸ 4	2025-02-16T19:19:40.311Z	c283306e-1758-437d-9aac-45ccf82658f4	2025/02/16/[$LATEST]0ddd7cb275ce40968f747116ffc1202e	1.54	2.0	128.0
▸ 5	2025-02-16T19:18:40.232Z	5db4643a-c759-4931-9a1d-b13f4e662f6f	2025/02/16/[$LATEST]0ddd7cb275ce40968f747116ffc1202e	1.8	2.0	128.0
▸ 6	2025-02-16T19:17:40.406Z	9d0fad2f-5343-45b8-8f59-e5d4a47f4254	2025/02/16/[$LATEST]0ddd7cb275ce40968f747116ffc1202e	1.84	2.0	128.0
▾ 7	2025-02-16T19:16:40.313Z	fa641329-8e74-4ee0-a577-c8bd63a20625	2025/02/16/[$LATEST]0ddd7cb275ce40968f747116ffc1202e	1.98	2.0	128.0
	@billedDuration	2.0				
	@duration	1.98				

Most expensive invocations in GB-seconds (memory assigned * billed duration)

#	Timestamp	RequestId	LogStream	BilledDurationInMS	MemorySetInMB	BilledDur
▸ 1	2025-02-16T19:22:40.204Z	046d5c0a-104a-4417-a4c2-7c46aecd2d4d	2025/02/16/[$LATEST]0ddd7cb275ce40968f747116ffc1202e	5.0	128	6.25E-4
▸ 2	2025-02-16T19:21:40.250Z	bd73db7d-defb-40a5-9d52-70f7d351686d	2025/02/16/[$LATEST]0ddd7cb275ce40968f747116ffc1202e	2.0	128	2.5E-4
▸ 3	2025-02-16T19:16:40.313Z	fa641329-8e74-4ee0-a577-c8bd63a20625	2025/02/16/[$LATEST]0ddd7cb275ce40968f747116ffc1202e	2.0	128	2.5E-4
▸ 4	2025-02-16T19:17:40.406Z	9d0fad2f-5343-45b8-8f59-e5d4a47f4254	2025/02/16/[$LATEST]0ddd7cb275ce40968f747116ffc1202e	2.0	128	2.5E-4
▸ 5	2025-02-16T19:18:40.232Z	5db4643a-c759-4931-9a1d-b13f4e662f6f	2025/02/16/[$LATEST]0ddd7cb275ce40968f747116ffc1202e	2.0	128	2.5E-4
▸ 6	2025-02-16T19:19:40.311Z	c283306e-1758-437d-9aac-45ccf82658f4	2025/02/16/[$LATEST]0ddd7cb275ce40968f747116ffc1202e	2.0	128	2.5E-4
▸ 7	2025-02-16T19:20:40.221Z	8e8e58af-a6aa-445d-9b08-fba05c07c277	2025/02/16/[$LATEST]0ddd7cb275ce40968f747116ffc1202e	2.0	128	2.5E-4

Conclusion

The AWS Lambda function `myFirstLambda` was successfully created and scheduled using EventBridge Rule `LambdaRule1`. The function was verified to run every minute by monitoring CloudWatch logs and execution metrics.