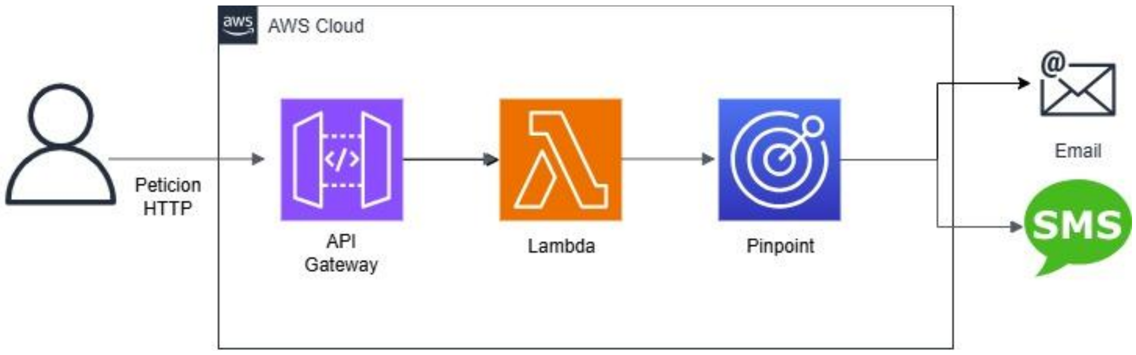
Serverless System for Sending SMS Notifications

The objective of this implementation is to provide the company or user with an efficient and scalable solution for sending SMS notifications, for example, to their customers. This system is based on several AWS services working together as follows:

We use Amazon API Gateway to receive HTTP requests, which trigger an AWS Lambda function. This Lambda function interacts with Amazon Pinpoint to send SMS messages. This approach enables an efficient serverless architecture, where API Gateway handles incoming requests, Lambda executes the business logic, and Pinpoint manages the SMS delivery, eliminating the need for dedicated servers and reducing operational costs.

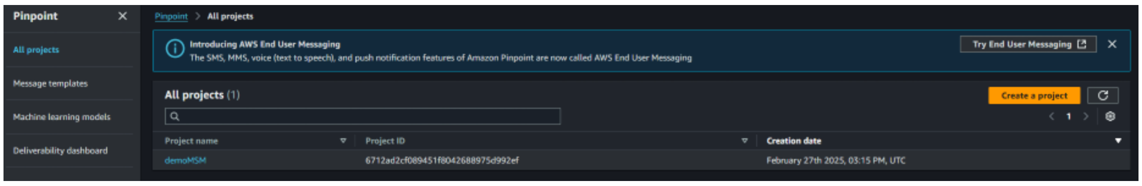
<u>Architecture</u>



<u>Amazon Pinpoint</u>

Amazon Pinpoint is an AWS service that allows businesses to send personalized and segmented messages across multiple channels, such as email, SMS, push notifications, and more. It is ideal for marketing campaigns, app notifications, and customer communications. It is important to note that this is a demo, and when using the Amazon Pinpoint sandbox account for SMS, you can only send messages to verified phone numbers. Additionally, your monthly spending limit is set at $1, and you can have up to 10 verified phone numbers per AWS region. To remove these restrictions, you can request a quota increase through AWS support.

Create a new project:

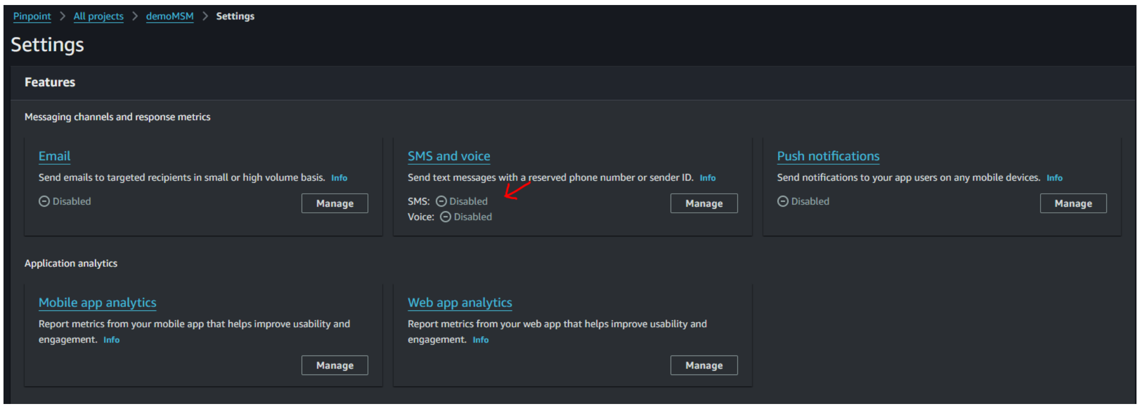


Introducing AWS End User Messaging
The SMS, MMS, voice (text to speech), and push notification features of Amazon Pinpoint are now called AWS End User Messaging

All projects (1)

Project name	Project ID	Creation date
demoMSM	6712ad2cf089451f8042688975d992ef	February 27th 2025, 03:15 PM, UTC

The "Project ID" is important because we will need it to identify our service later in the Lambda function. When configuring Amazon Pinpoint, SMS and other services are disabled by default.



Pinpoint > All projects > demoMSM > Settings

Settings

Features

Messaging channels and response metrics

Email
Send emails to targeted recipients in small or high volume basis. Info
Disabled
Manage

SMS and voice
Send text messages with a reserved phone number or sender ID. Info
SMS: Disabled
Voice: Disabled
Manage

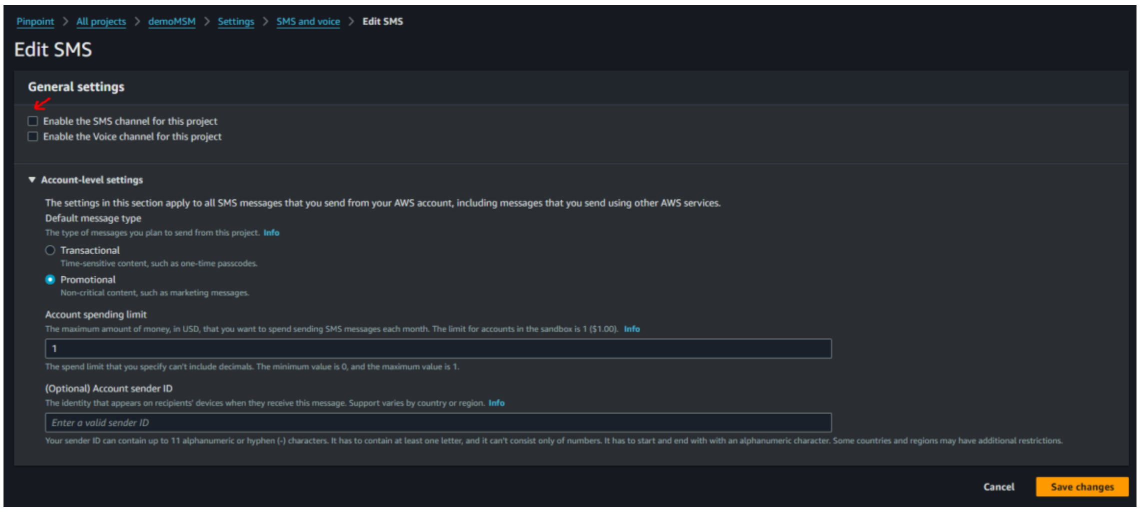
Push notifications
Send notifications to your app users on any mobile devices. Info
Disabled
Manage

Application analytics

Mobile app analytics
Report metrics from your mobile app that helps improve usability and engagement. Info
Manage

Web app analytics
Report metrics from your web app that helps improve usability and engagement. Info
Manage

You need to enable it. Therefore, in Settings/SMS > Edit, we activate the SMS service:



Pinpoint > All projects > demoMSM > Settings > SMS and voice > Edit SMS

Edit SMS

General settings

- [ ] Enable the SMS channel for this project
- [ ] Enable the Voice channel for this project

▼ Account-level settings

The settings in this section apply to all SMS messages that you send from your AWS account, including messages that you send using other AWS services.

Default message type
The type of messages you plan to send from this project. Info

○ Transactional
Time-sensitive content, such as one-time passcodes.

● Promotional
Non-critical content, such as marketing messages.

Account spending limit
The maximum amount of money, in USD, that you want to spend sending SMS messages each month. The limit for accounts in the sandbox is 1 ($1.00). Info

1

The spend limit that you specify can't include decimals. The minimum value is 0, and the maximum value is 1.

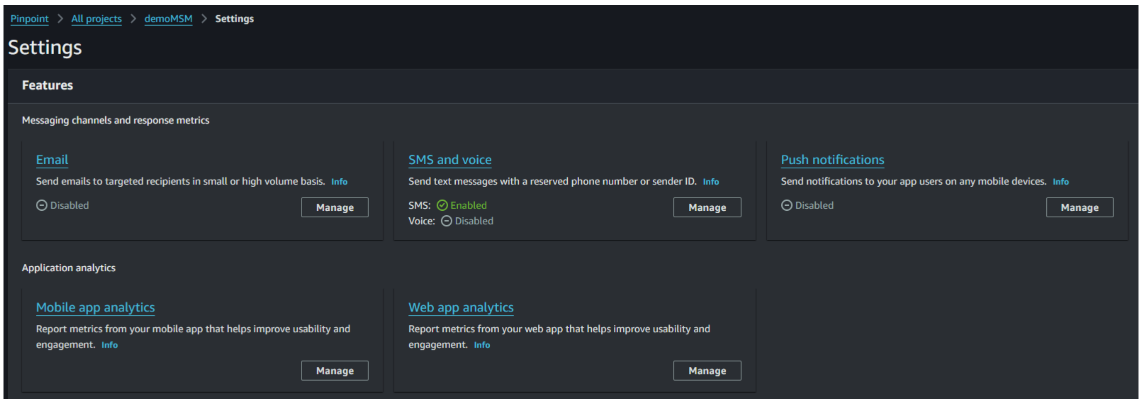
(Optional) Account sender ID
The identity that appears on recipients' devices when they receive this message. Support varies by country or region. Info

Enter a valid sender ID

Your sender ID can contain up to 11 alphanumeric or hyphen (-) characters. It has to contain at least one letter, and it can't consist only of numbers. It has to start and end with with an alphanumeric character. Some countries and regions may have additional restrictions.

Cancel | Save changes

Check that it is now enabled:



Pinpoint > All projects > demoMSM > Settings

Settings

Features

Messaging channels and response metrics

Email
Send emails to targeted recipients in small or high volume basis. Info
Disabled
Manage

SMS and voice
Send text messages with a reserved phone number or sender ID. Info
SMS: Enabled
Voice: Disabled
Manage

Push notifications
Send notifications to your app users on any mobile devices. Info
Disabled
Manage

Application analytics

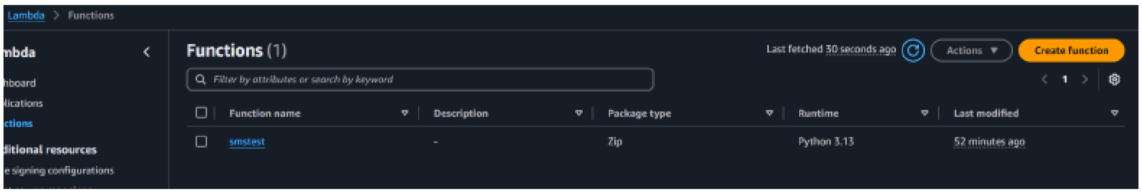
Mobile app analytics
Report metrics from your mobile app that helps improve usability and engagement. Info
Manage

Web app analytics
Report metrics from your web app that helps improve usability and engagement. Info
Manage

Lambda

We have chosen AWS Lambda to execute scripts on demand because it offers automatic scalability, reduced costs by paying only for execution time, easy integration with other AWS services, and rapid deployment. This eliminates the need to manage servers, allowing us to have a completely serverless system and avoid the costs and maintenance of a dedicated 24/7 instance.

We will create a new function, and in this case, we will develop it in Python. Therefore, we need to select this language in the configuration steps.


Lambda > Functions

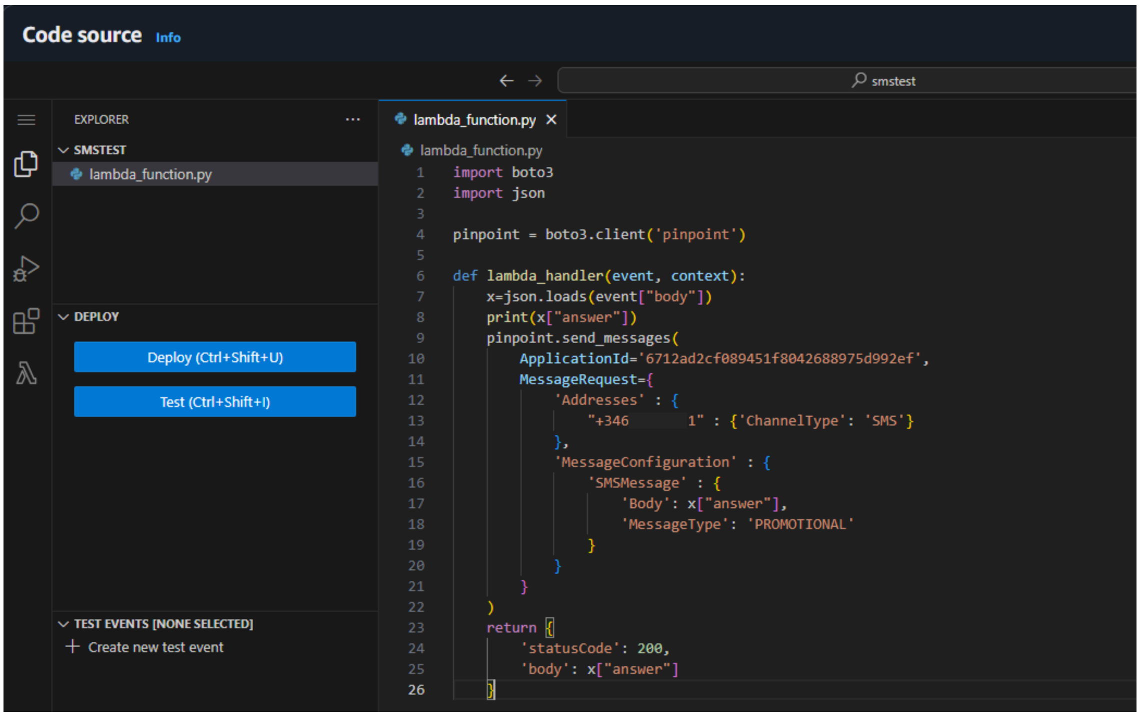
Functions (1)

Last fetched 30 seconds ago

	Function name	Description	Package type	Runtime	Last modified
☐	smstest	-	Zip	Python 3.13	52 minutes ago

Below, show the Python code, which will receive the request through an HTTP request routed via an API Gateway, process it, and invoke the Amazon Pinpoint service. In the "Code" section, we enter the code for our function:

https://github.com/garrijuan/AWS-SMS-Notification/blob/main/pinpoint.py


Code source Info

lambda_function.py

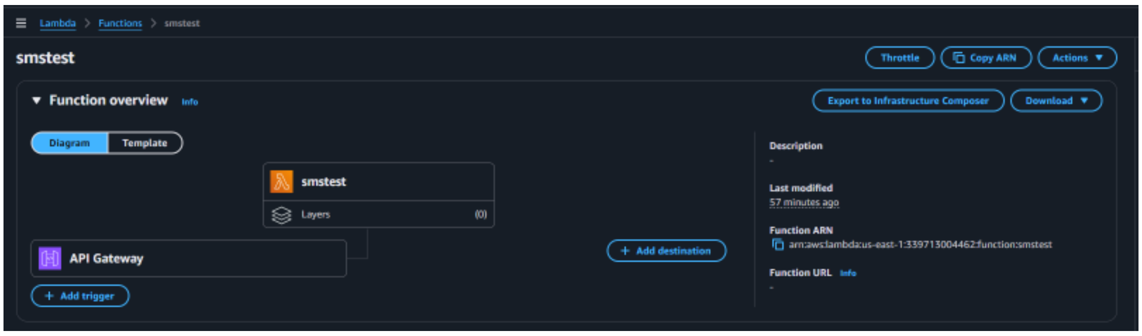
```python
import boto3
import json

pinpoint = boto3.client('pinpoint')

def lambda_handler(event, context):
    x=json.loads(event["body"])
    print(x["answer"])
    pinpoint.send_messages(
        ApplicationId='6712ad2cf089451f8042688975d992ef',
        MessageRequest={
            'Addresses' : {
                "+346[redacted]1" : {'ChannelType': 'SMS'}
            },
            'MessageConfiguration' : {
                'SMSMessage' : {
                    'Body': x["answer"],
                    'MessageType': 'PROMOTIONAL'
                }
            }
        }
    )
    return {
        'statusCode': 200,
        'body': x["answer"]
    }
```

We need to specify the "ProjectID" mentioned earlier to invoke the messaging application. We have also parameterized the message to send the desired content at any given time, and we could similarly parameterize the destination phone number to send messages wherever needed.

Once the code is ready, we will "Deploy" the function to make the Lambda operational. In this case, to interact with Lambda, we have created an API Gateway to handle HTTP requests. To do this, we need to add a Trigger in this section and configure it.



Lambda > Functions > smstest

smstest

Function overview

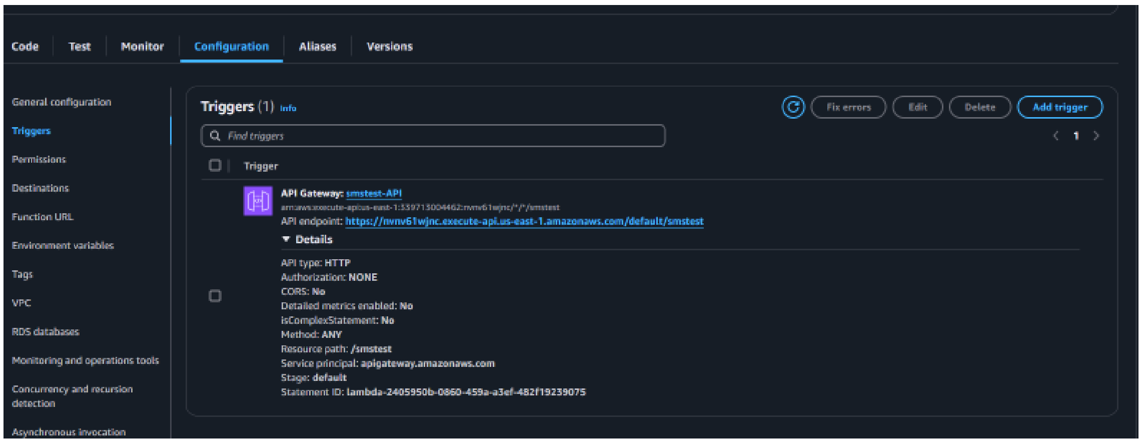
Description
-

Last modified
57 minutes ago

Function ARN
arn:aws:lambda:us-east-1:339713004462:function:smstest

Function URL
-

Endpoint:



Triggers (1)

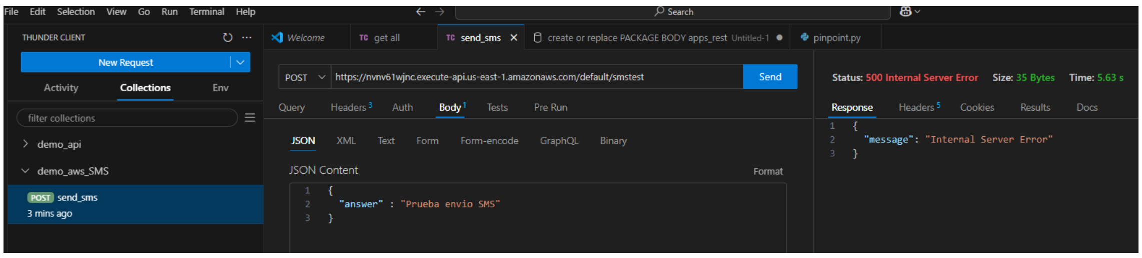
Trigger

API Gateway: smstest-API
arn:aws:execute-api:us-east-1:339713004462:nvnv61wjnc/*/*/smstest
API endpoint: https://nvnv61wjnc.execute-api.us-east-1.amazonaws.com/default/smstest

Details

API type: HTTP
Authorization: NONE
CORS: No
Detailed metrics enabled: No
isComplexStatement: No
Method: ANY
Resource path: /smstest
Service principal: apigateway.amazonaws.com
Stage: default
Statement ID: lambda-2405950b-0860-459a-a3ef-482f19239075

After all steps, we are ready to perform an "end-to-end" test. We'll prepare a client to send requests, in this case using Thunder Client in VS Code. Next, we can check that the request has failed:



POST https://nvnv61wjnc.execute-api.us-east-1.amazonaws.com/default/smstest

JSON Content

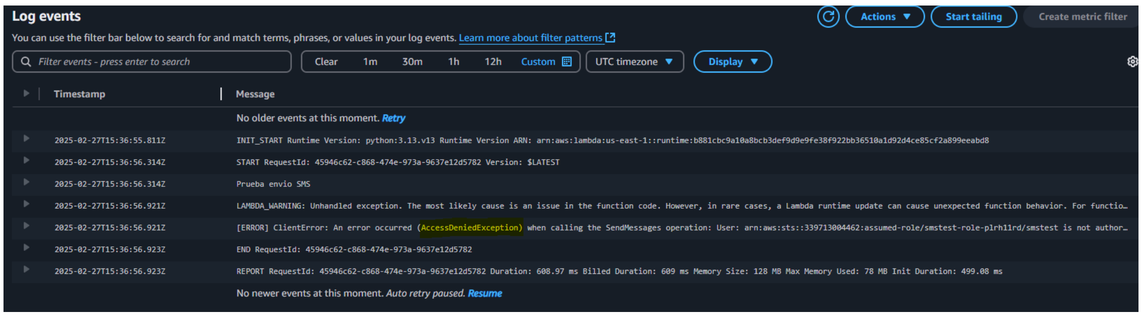
```
{
  "answer" : "Prueba envio SMS"
}
```

Status: 500 Internal Server Error   Size: 35 Bytes   Time: 5.63 s

Response

```
{
  "message": "Internal Server Error"
}
```

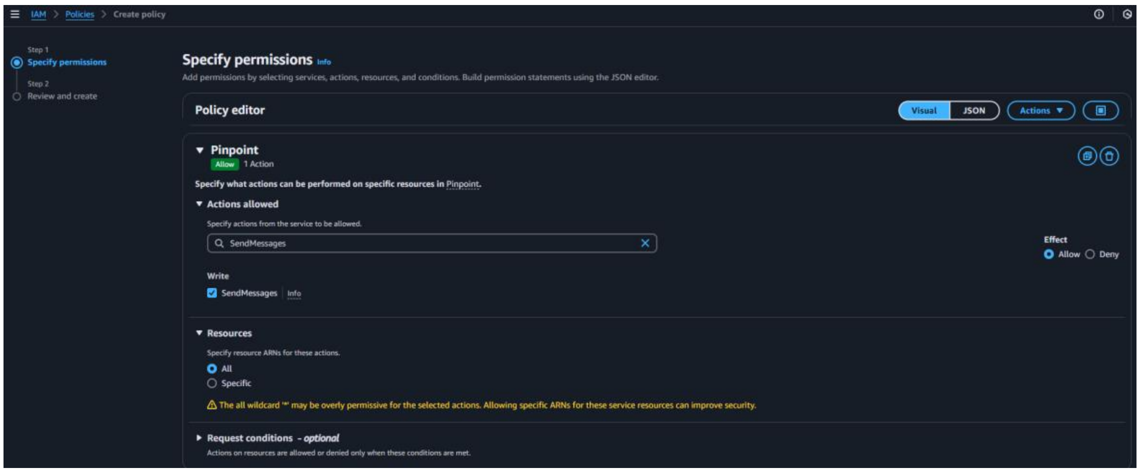
Lambda log showing access denied error:


Log events
You can use the filter bar below to search for and match terms, phrases, or values in your log events. Learn more about filter patterns
Actions
Start tailing
Create metric filter
Filter events - press enter to search
Clear 1m 30m 1h 12h Custom
UTC timezone
Display

```
Timestamp                  Message
                           No older events at this moment. Retry
2025-02-27T15:36:55.811Z   INIT_START Runtime Version: python:3.13.v13 Runtime Version ARN: arn:aws:lambda:us-east-1::runtime:b881cbc9a10a8bcb3def9d9e9fe38f922bb36510a1d92d4ce85cf2a899eeabd8
2025-02-27T15:36:56.314Z   START RequestId: 45946c62-c868-474e-973a-9637e12d5782 Version: $LATEST
2025-02-27T15:36:56.314Z   Prueba envio SMS
2025-02-27T15:36:56.921Z   LAMBDA_WARNING: Unhandled exception. The most likely cause is an issue in the function code. However, in rare cases, a Lambda runtime update can cause unexpected function behavior. For functio…
2025-02-27T15:36:56.921Z   [ERROR] ClientError: An error occurred (AccessDeniedException) when calling the SendMessages operation: User: arn:aws:sts::339713004462:assumed-role/smstest-role-plrh11rd/smstest is not author…
2025-02-27T15:36:56.923Z   END RequestId: 45946c62-c868-474e-973a-9637e12d5782
2025-02-27T15:36:56.923Z   REPORT RequestId: 45946c62-c868-474e-973a-9637e12d5782 Duration: 608.97 ms Billed Duration: 609 ms Memory Size: 128 MB Max Memory Used: 78 MB Init Duration: 499.08 ms
                           No newer events at this moment. Auto retry paused. Resume
```

Policy

It is necessary to create a policy and assign it to the role that Lambda automatically created during its setup. Next, create the policy:



Specify permissions Info

Add permissions by selecting services, actions, resources, and conditions. Build permission statements using the JSON editor.

Policy editor

Pinpoint
Allow 1 Action

Specify what actions can be performed on specific resources in Pinpoint.

Actions allowed
Specify actions from the service to be allowed.
SendMessages

Effect
[✓ Allow] Deny

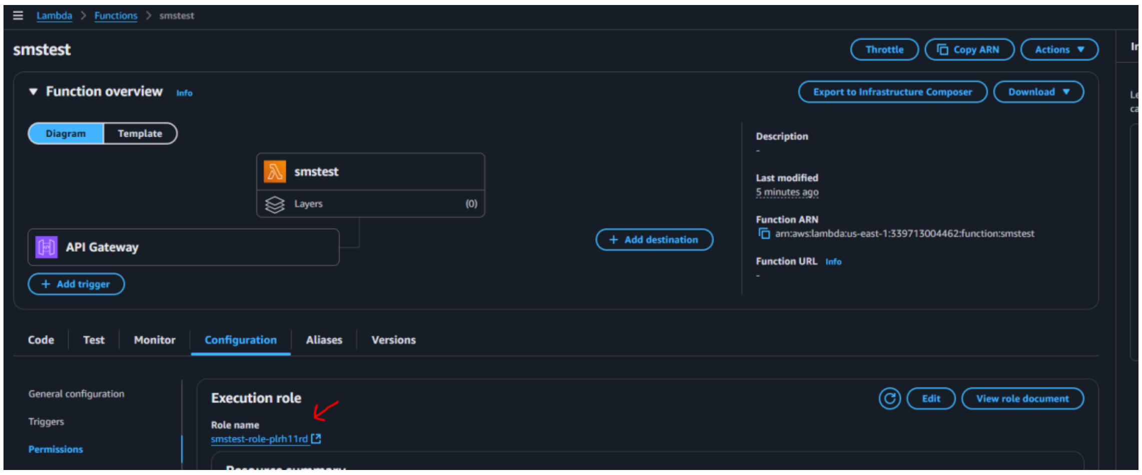
Write
[✓ SendMessages] Info

Resources
Specify resource ARNs for these actions.
[✓ All]
Specific

The all wildcard '*' may be overly permissive for the selected actions. Allowing specific ARNs for these service resources can improve security.

Request conditions - *optional*
Actions on resources are allowed or denied only when these conditions are met.

Role and policy association:



Description
-

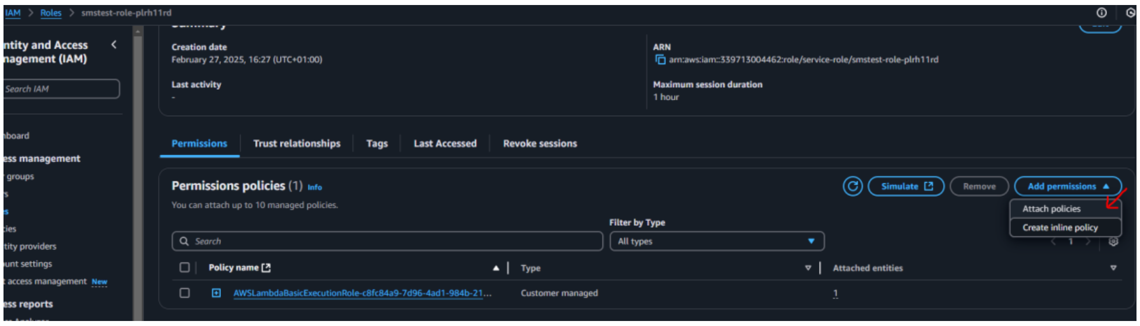
Last modified
5 minutes ago

Function ARN
arn:aws:lambda:us-east-1:339713004462:function:smstest

Function URL Info
-

Role name
smstest-role-plrh11rd

Click on "Attach policy":



IAM > Roles > smstest-role-plrh11rd

Creation date
February 27, 2025, 16:27 (UTC+01:00)

ARN
arn:aws:iam::339713004462:role/service-role/smstest-role-plrh11rd

Last activity
-

Maximum session duration
1 hour

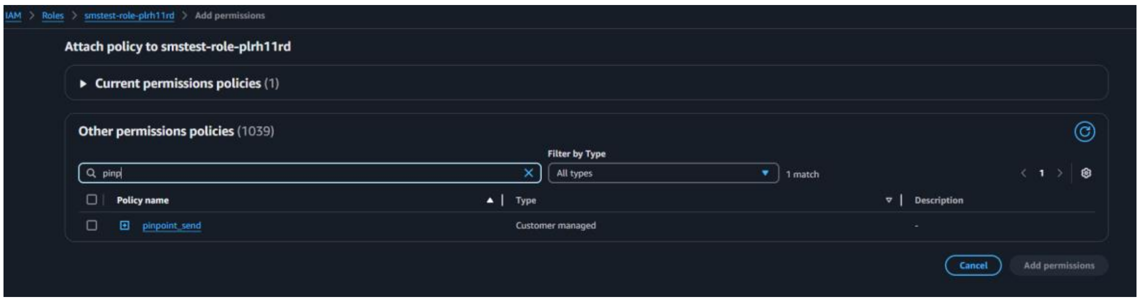
Permissions | Trust relationships | Tags | Last Accessed | Revoke sessions

Permissions policies (1) Info
You can attach up to 10 managed policies.

Filter by Type: All types

Policy name	Type	Attached entities
AWSLambdaBasicExecutionRole-c8fc84a9-7d96-4ad1-984b-21...	Customer managed	1

Select the policy created earlier:



IAM > Roles > smstest-role-plrh11rd > Add permissions

Attach policy to smstest-role-plrh11rd

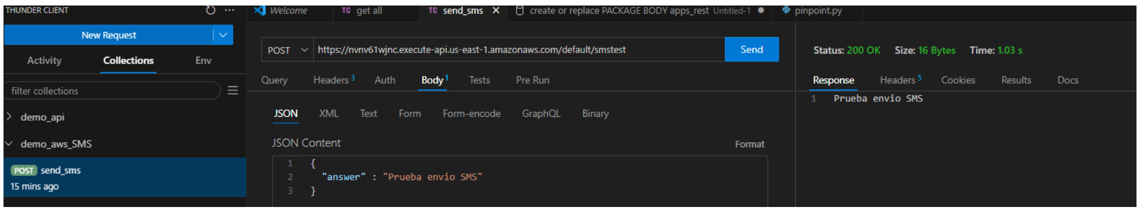
Current permissions policies (1)

Other permissions policies (1039)

Search: pinp
Filter by Type: All types
1 match

Policy name	Type	Description
pinpoint_send	Customer managed	-

At this point, we are ready to repeat the test. From Thunder Client, we send the request:



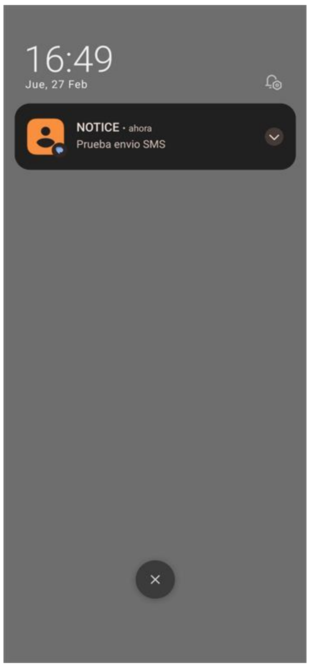
POST https://nvnv61wjnc.execute-api.us-east-1.amazonaws.com/default/smstest

Status: 200 OK Size: 16 Bytes Time: 1.03 s

JSON Content
```
{
  "answer" : "Prueba envio SMS"
}
```

Response
```
Prueba envio SMS
```

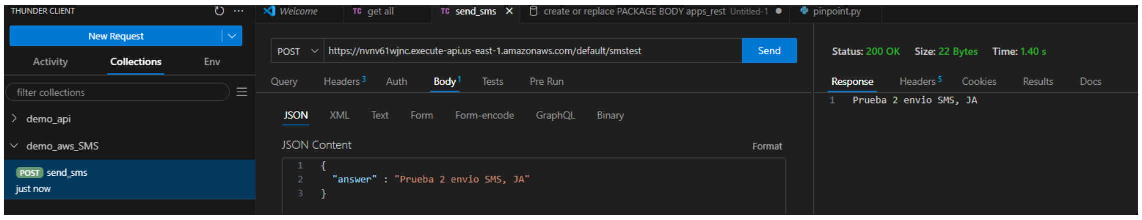
Now we can verify that the request is processed correctly.



NOTICE · ahora
Prueba envio SMS

The SMS is sent immediately.

We repeat the test again by providing another message:



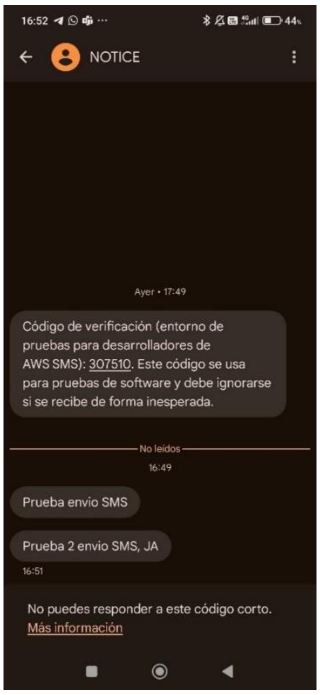
POST https://nvnv61wjnc.execute-api.us-east-1.amazonaws.com/default/smstest

```
{
  "answer" : "Prueba 2 envio SMS, JA"
}
```

Status: 200 OK Size: 22 Bytes Time: 1.40 s

```
Prueba 2 envio SMS, JA
```

Received SMS:



Ayer • 17:49

Código de verificación (entorno de pruebas para desarrolladores de AWS SMS): 307510. Este código se usa para pruebas de software y debe ignorarse si se recibe de forma inesperada.

No leídos

16:49

Prueba envio SMS

Prueba 2 envio SMS, JA

16:51

No puedes responder a este código corto.
Más información

Conclusions

In this implementation, we have used Amazon API Gateway and AWS Lambda to create an efficient and scalable system for sending SMS notifications to customers. It is important to note that both API Gateway and Lambda, as well as Pinpoint, only charge for usage, not for having them deployed and hosted in the cloud, which allows for more effective cost control. The main idea is that any invoking application or service can call this Web Service to use the communication system, which is extendable to other clouds offering similar services. For example, we could use services like Twilio instead of Amazon Pinpoint for sending SMS, depending on the company's needs and preferences. This approach not only provides a serverless solution that reduces costs and simplifies management but also offers flexibility to adapt to different providers and cloud environments, ensuring efficient and timely communication with customers.