Production-Style GenAI API– Latency-First Engineering (AWS)

Project Overview

This project evaluates and optimizes **latency behavior in a serverless Generative AI API** built on AWS. The system was designed to identify dominant latency contributors and improve tail latency through **measured, minimal changes** rather than architectural overengineering.
The primary goal was to **reduce p95 and p99 response times** while maintaining functional correctness and lowering cost.

Architecture

Request Flow:

Client → API Gateway → AWS Lambda → Amazon Bedrock → Lambda → Client

Key AWS Services:

- Amazon API Gateway (HTTP endpoint)
- AWS Lambda (Python runtime)
- Amazon Bedrock (Foundation Model inference)
- Amazon CloudWatch (Logs, Metrics, Dashboards)

Problem Statement

Initial testing showed **high tail latency (p95 / p99)** for the GenAI API, even under low request volume.
To improve system performance, the project focused on answering:

- Where is the latency actually coming from?
- Is the bottleneck compute, networking, or inference?
- Can latency be reduced without adding infrastructure?

Measurement Approach

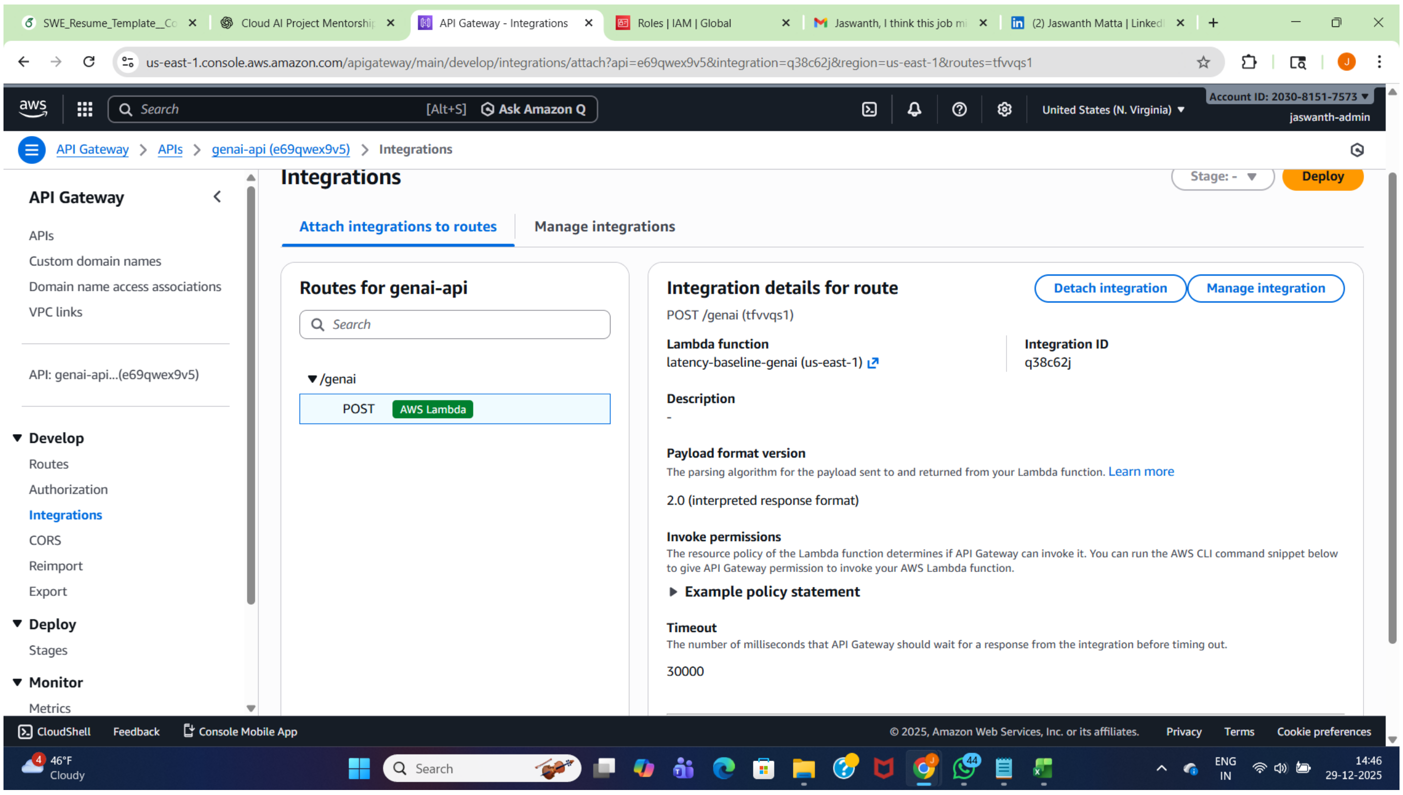
Latency was measured directly inside the Lambda function using high-resolution timestamps:

- **Bedrock inference time**
- **Total Lambda execution time**
- **Cold start vs warm execution behavior**

Multiple requests were issued via CloudShell to observe:

- p50 (median)
- p95 (tail latency)
- p99 (cold start impact)

CloudWatch Logs and Dashboards were used for validation.



us-east-1.console.aws.amazon.com/apigateway/main/develop/integrations/attach?api=e69qwex9v5&integration=q38c62j&region=us-east-1&routes=tfvvqs1

API Gateway > APIs > genai-api (e69qwex9v5) > Integrations

Integrations

Attach integrations to routes | Manage integrations

Routes for genai-api

/genai
POST AWS Lambda

Integration details for route

POST /genai (tfvvqs1)

Lambda function
latency-baseline-genai (us-east-1)

Integration ID
q38c62j

Description
-

Payload format version
The parsing algorithm for the payload sent to and returned from your Lambda function. Learn more

2.0 (interpreted response format)

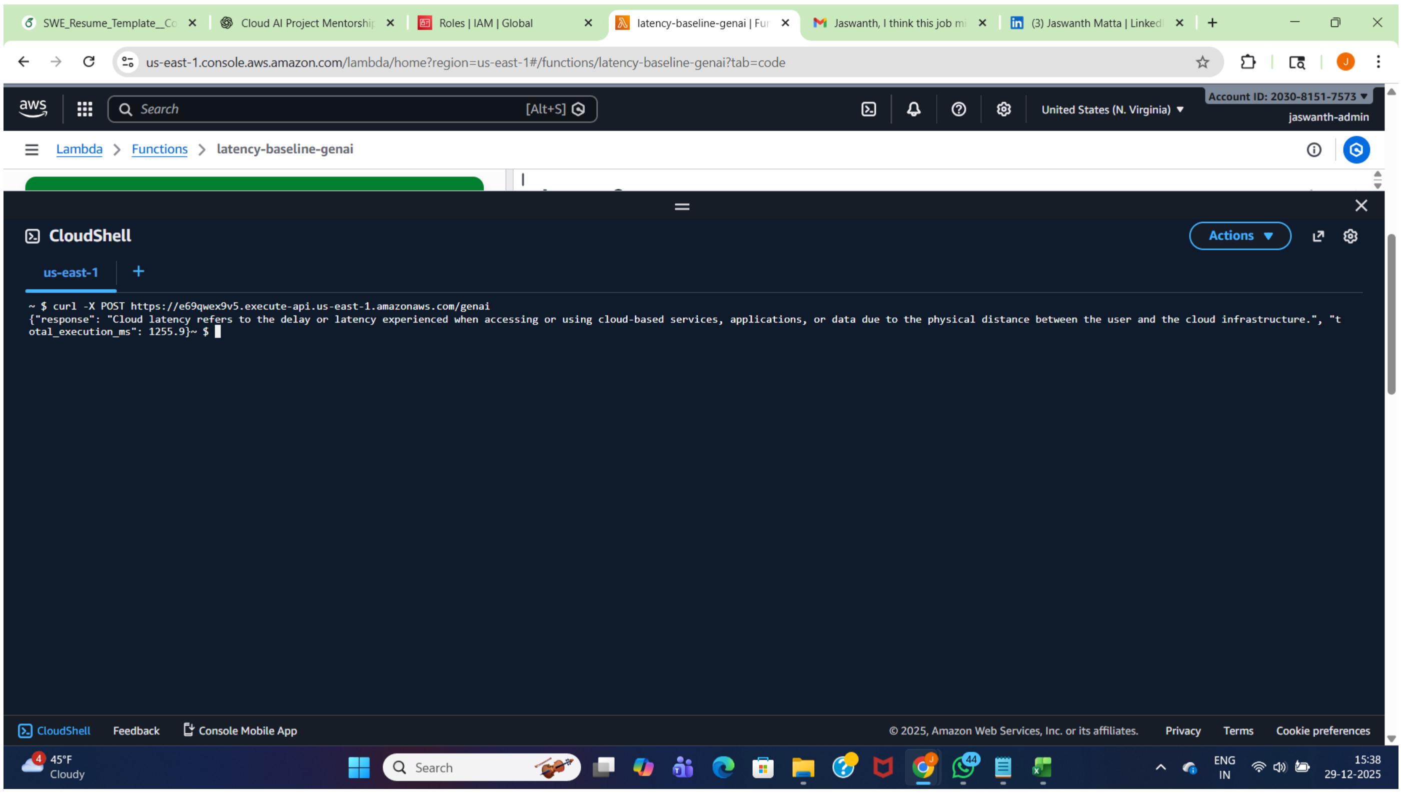
Invoke permissions
The resource policy of the Lambda function determines if API Gateway can invoke it. You can run the AWS CLI command snippet below to give API Gateway permission to invoke your AWS Lambda function.

Example policy statement

Timeout
The number of milliseconds that API Gateway should wait for a response from the integration before timing out.

30000

End-to-end API invocation through API Gateway triggering a Lambda-based GenAI inference request.



us-east-1.console.aws.amazon.com/lambda/home?region=us-east-1#/functions/latency-baseline-genai?tab=code

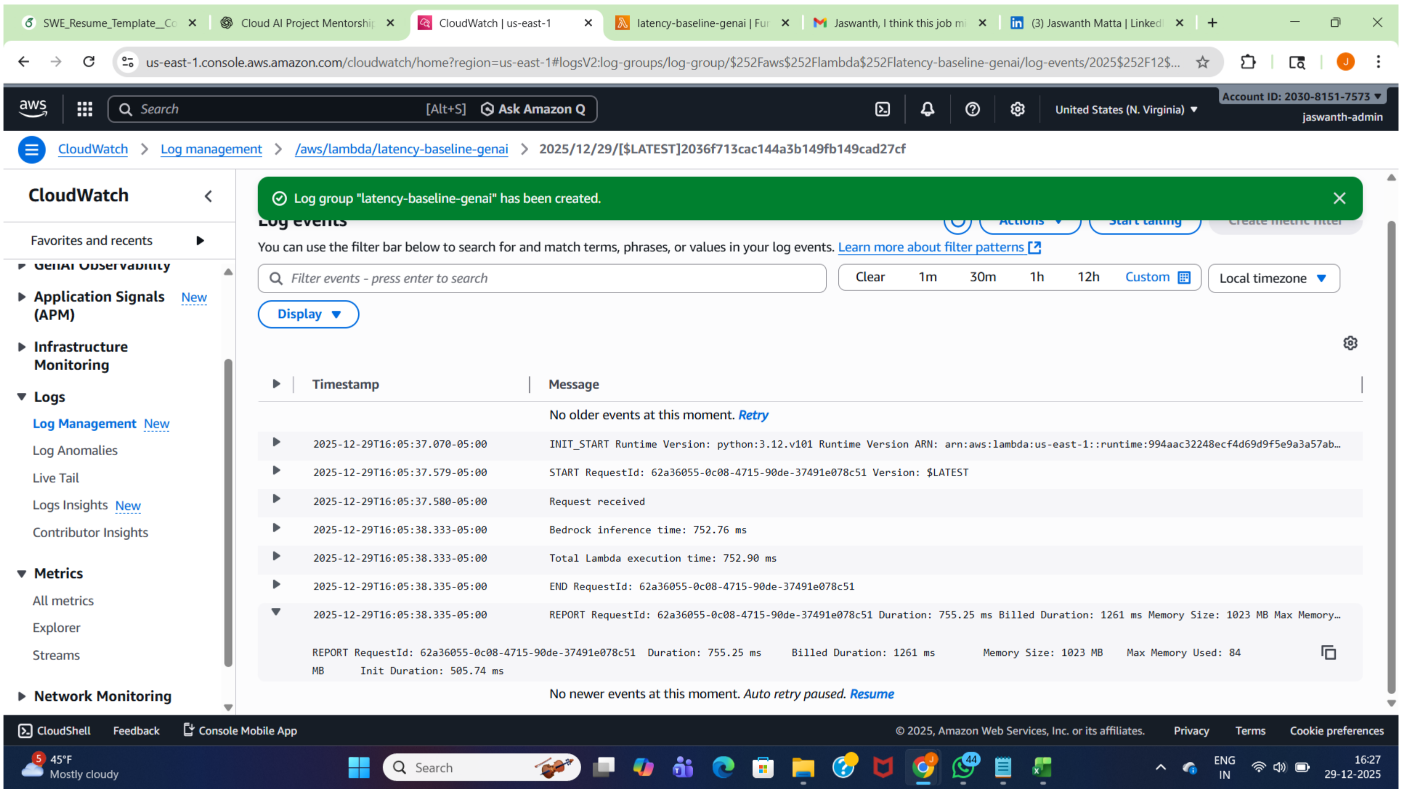
Lambda > Functions > latency-baseline-genai

CloudShell

us-east-1

```
~ $ curl -X POST https://e69qwex9v5.execute-api.us-east-1.amazonaws.com/genai
{"response": "Cloud latency refers to the delay or latency experienced when accessing or using cloud-based services, applications, or data due to the physical distance between the user and the cloud infrastructure.", "t
otal_execution_ms": 1255.9}~ $ 
```

CloudWatch logs capturing cold start initialization, model inference time, and total Lambda execution latency.



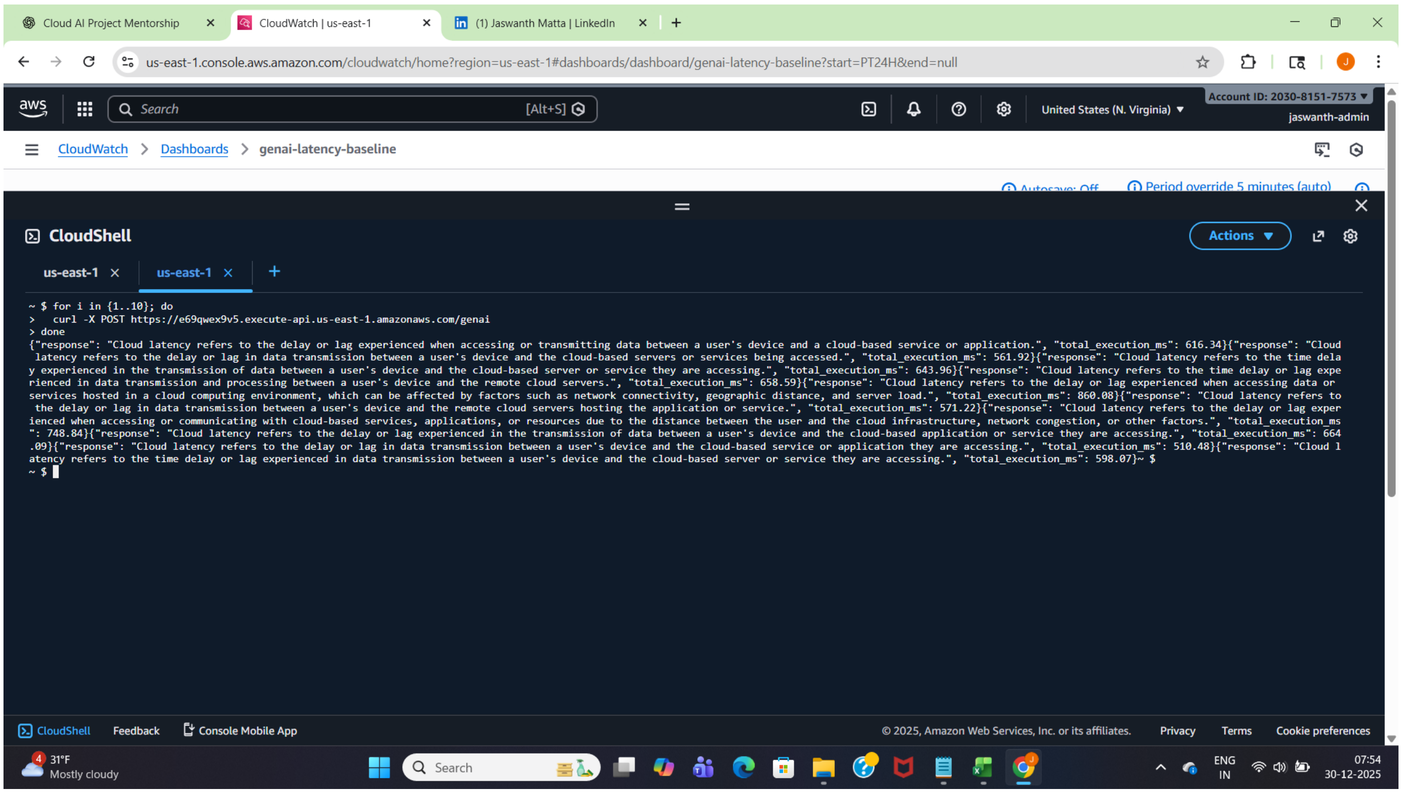
Log group "latency-baseline-genai" has been created.

CloudWatch > Log management > /aws/lambda/latency-baseline-genai > 2025/12/29/[$LATEST]2036f713cac144a3b149fb149cad27cf

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

```
Timestamp                      Message
                               No older events at this moment. Retry
2025-12-29T16:05:37.070-05:00  INIT_START Runtime Version: python:3.12.v101 Runtime Version ARN: arn:aws:lambda:us-east-1::runtime:994aac32248ecf4d69d9f5e9a3a57ab…
2025-12-29T16:05:37.579-05:00  START RequestId: 62a36055-0c08-4715-90de-37491e078c51 Version: $LATEST
2025-12-29T16:05:37.580-05:00  Request received
2025-12-29T16:05:38.333-05:00  Bedrock inference time: 752.76 ms
2025-12-29T16:05:38.333-05:00  Total Lambda execution time: 752.90 ms
2025-12-29T16:05:38.335-05:00  END RequestId: 62a36055-0c08-4715-90de-37491e078c51
2025-12-29T16:05:38.335-05:00  REPORT RequestId: 62a36055-0c08-4715-90de-37491e078c51 Duration: 755.25 ms Billed Duration: 1261 ms Memory Size: 1023 MB Max Memory…

REPORT RequestId: 62a36055-0c08-4715-90de-37491e078c51  Duration: 755.25 ms    Billed Duration: 1261 ms      Memory Size: 1023 MB    Max Memory Used: 84
MB      Init Duration: 505.74 ms
                               No newer events at this moment. Auto retry paused. Resume
```

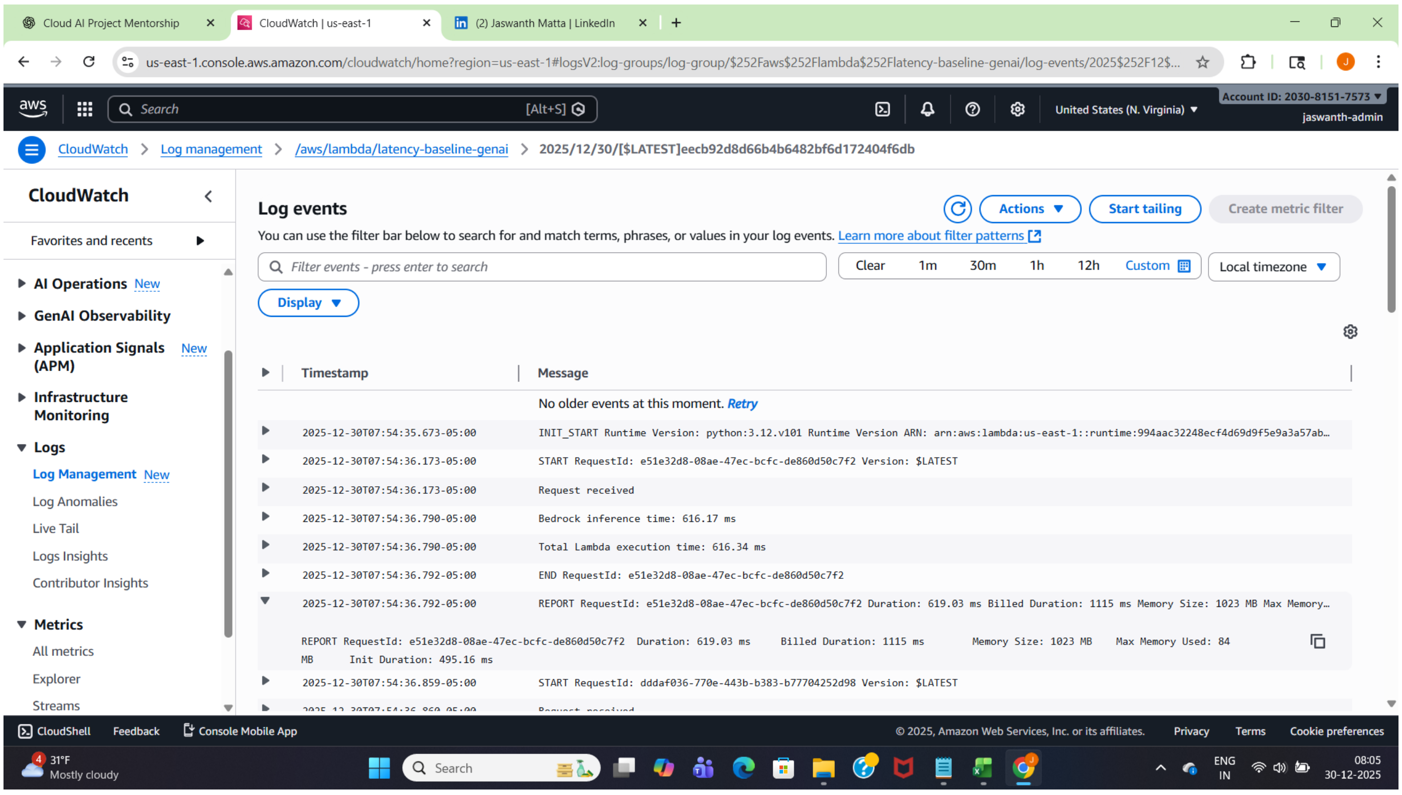
Repeated API invocations used to generate latency distribution data for p50, p95, and p99 analysis.



CloudWatch > Dashboards > genai-latency-baseline

```
~ $ for i in {1..10}; do
>   curl -X POST https://e69qwex9v5.execute-api.us-east-1.amazonaws.com/genai
> done
{"response": "Cloud latency refers to the delay or lag experienced when accessing or transmitting data between a user's device and a cloud-based service or application.", "total_execution_ms": 616.34}{"response": "Cloud
 latency refers to the delay or lag in data transmission between a user's device and the cloud-based servers or services being accessed.", "total_execution_ms": 561.92}{"response": "Cloud latency refers to the time dela
y experienced in the transmission of data between a user's device and the cloud-based server or service they are accessing.", "total_execution_ms": 643.96}{"response": "Cloud latency refers to the time delay or lag expe
rienced in data transmission and processing between a user's device and the remote cloud servers.", "total_execution_ms": 658.59}{"response": "Cloud latency refers to the delay or lag experienced when accessing data or
services hosted in a cloud computing environment, which can be affected by factors such as network connectivity, geographic distance, and server load.", "total_execution_ms": 860.08}{"response": "Cloud latency refers to
 the delay or lag in data transmission between a user's device and the remote cloud servers hosting the application or service.", "total_execution_ms": 571.22}{"response": "Cloud latency refers to the delay or lag exper
ienced when accessing or communicating with cloud-based services, applications, or resources due to the distance between the user and the cloud infrastructure, network congestion, or other factors.", "total_execution_ms
": 748.84}{"response": "Cloud latency refers to the delay or lag experienced in the transmission of data between a user's device and the cloud-based application or service they are accessing.", "total_execution_ms": 664
.09}{"response": "Cloud latency refers to the delay or lag in data transmission between a user's device and the cloud-based service or application they are accessing.", "total_execution_ms": 510.48}{"response": "Cloud l
atency refers to the time delay or lag experienced in data transmission between a user's device and the cloud-based server or service they are accessing.", "total_execution_ms": 598.07}~ $
~ $
```

CloudWatch logs showing cold start initialization overhead and inference-dominated execution latency



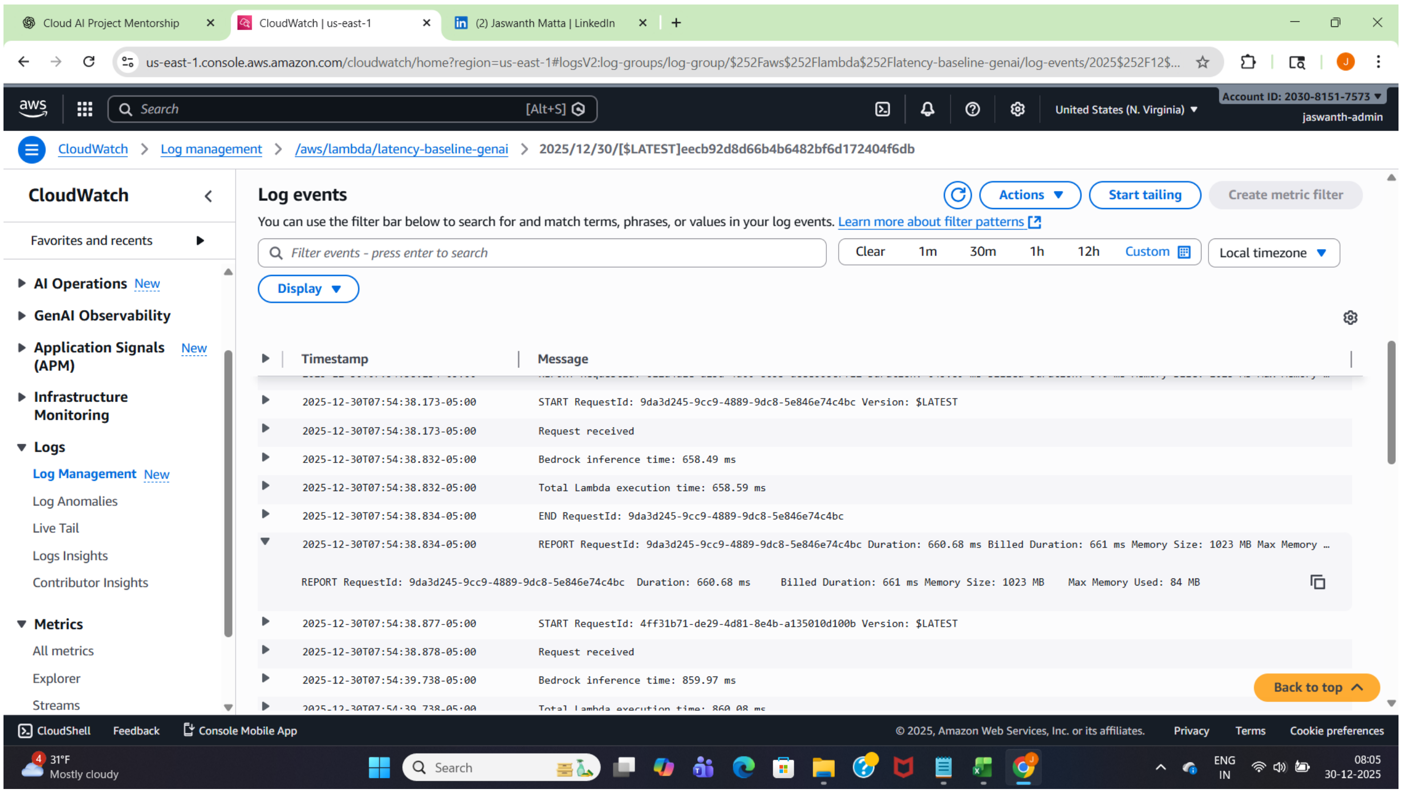
CloudWatch > Log management > /aws/lambda/latency-baseline-genai > 2025/12/30/[$LATEST]eecb92d8d66b4b6482bf6d172404f6db

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

```
Timestamp                        Message
                                 No older events at this moment. Retry
2025-12-30T07:54:35.673-05:00    INIT_START Runtime Version: python:3.12.v101 Runtime Version ARN: arn:aws:lambda:us-east-1::runtime:994aac32248ecf4d69d9f5e9a3a57ab…
2025-12-30T07:54:36.173-05:00    START RequestId: e51e32d8-08ae-47ec-bcfc-de860d50c7f2 Version: $LATEST
2025-12-30T07:54:36.173-05:00    Request received
2025-12-30T07:54:36.790-05:00    Bedrock inference time: 616.17 ms
2025-12-30T07:54:36.790-05:00    Total Lambda execution time: 616.34 ms
2025-12-30T07:54:36.792-05:00    END RequestId: e51e32d8-08ae-47ec-bcfc-de860d50c7f2
2025-12-30T07:54:36.792-05:00    REPORT RequestId: e51e32d8-08ae-47ec-bcfc-de860d50c7f2 Duration: 619.03 ms Billed Duration: 1115 ms Memory Size: 1023 MB Max Memory…

REPORT RequestId: e51e32d8-08ae-47ec-bcfc-de860d50c7f2  Duration: 619.03 ms    Billed Duration: 1115 ms      Memory Size: 1023 MB    Max Memory Used: 84
MB      Init Duration: 495.16 ms
2025-12-30T07:54:36.859-05:00    START RequestId: dddaf036-770e-443b-b383-b77704252d98 Version: $LATEST
```

Warm Lambda execution demonstrating reduced and consistent inference latency without initialization overhead.

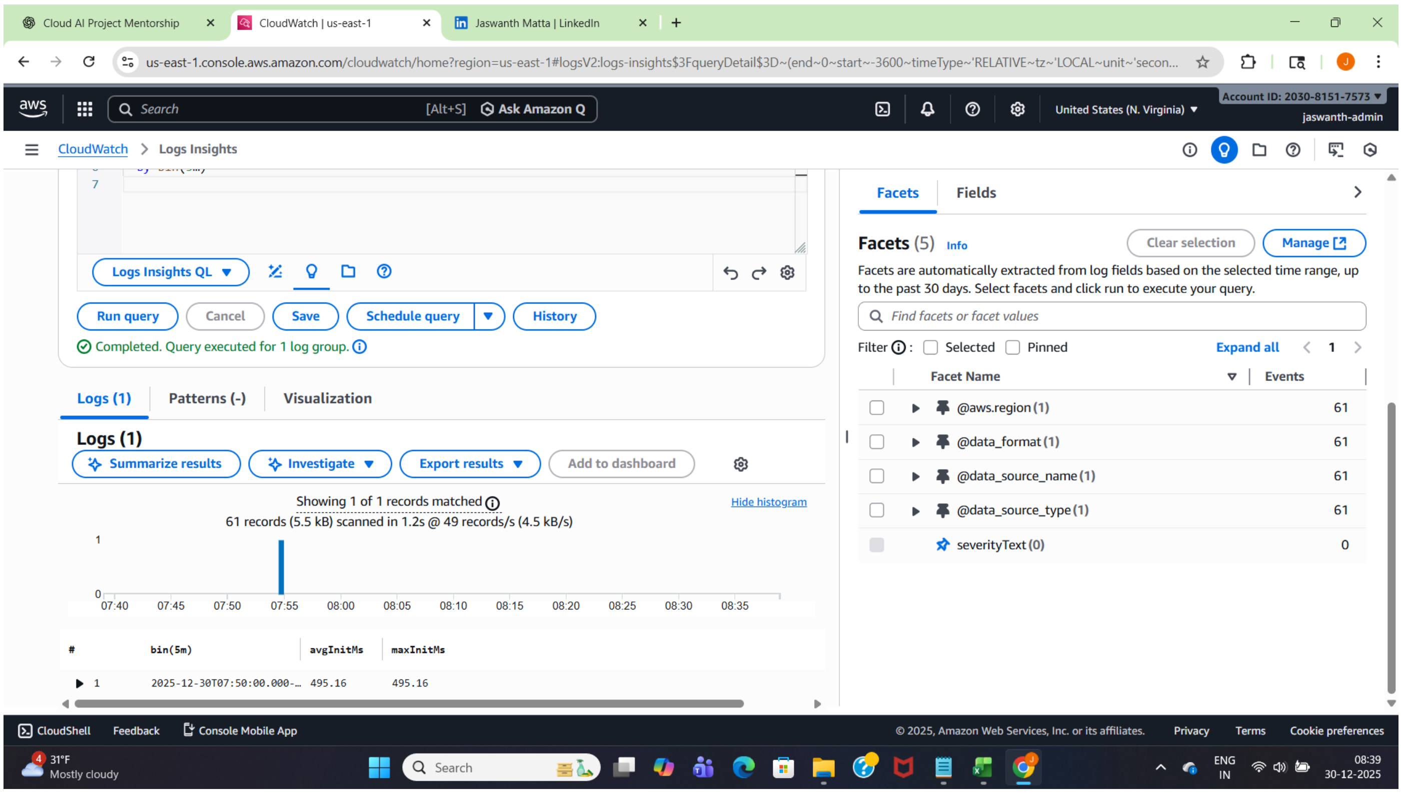


CloudWatch > Log management > /aws/lambda/latency-baseline-genai > 2025/12/30/[$LATEST]eecb92d8d66b4b6482bf6d172404f6db

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

```
Timestamp                        Message
2025-12-30T07:54:38.173-05:00    START RequestId: 9da3d245-9cc9-4889-9dc8-5e846e74c4bc Version: $LATEST
2025-12-30T07:54:38.173-05:00    Request received
2025-12-30T07:54:38.832-05:00    Bedrock inference time: 658.49 ms
2025-12-30T07:54:38.832-05:00    Total Lambda execution time: 658.59 ms
2025-12-30T07:54:38.834-05:00    END RequestId: 9da3d245-9cc9-4889-9dc8-5e846e74c4bc
2025-12-30T07:54:38.834-05:00    REPORT RequestId: 9da3d245-9cc9-4889-9dc8-5e846e74c4bc Duration: 660.68 ms Billed Duration: 661 ms Memory Size: 1023 MB Max Memory …

REPORT RequestId: 9da3d245-9cc9-4889-9dc8-5e846e74c4bc  Duration: 660.68 ms    Billed Duration: 661 ms Memory Size: 1023 MB    Max Memory Used: 84 MB
2025-12-30T07:54:38.877-05:00    START RequestId: 4ff31b71-de29-4d81-8e4b-a135010d100b Version: $LATEST
2025-12-30T07:54:38.878-05:00    Request received
2025-12-30T07:54:39.738-05:00    Bedrock inference time: 859.97 ms
```


us-east-1.console.aws.amazon.com/cloudwatch/home?region=us-east-1#logsV2:logs-insights$3FqueryDetail$3D~(end~0~start~-3600~timeType~'RELATIVE~tz~'LOCAL~unit~'secon...

CloudWatch > Logs Insights

Completed. Query executed for 1 log group.

Logs (1)

Showing 1 of 1 records matched

61 records (5.5 kB) scanned in 1.2s @ 49 records/s (4.5 kB/s)

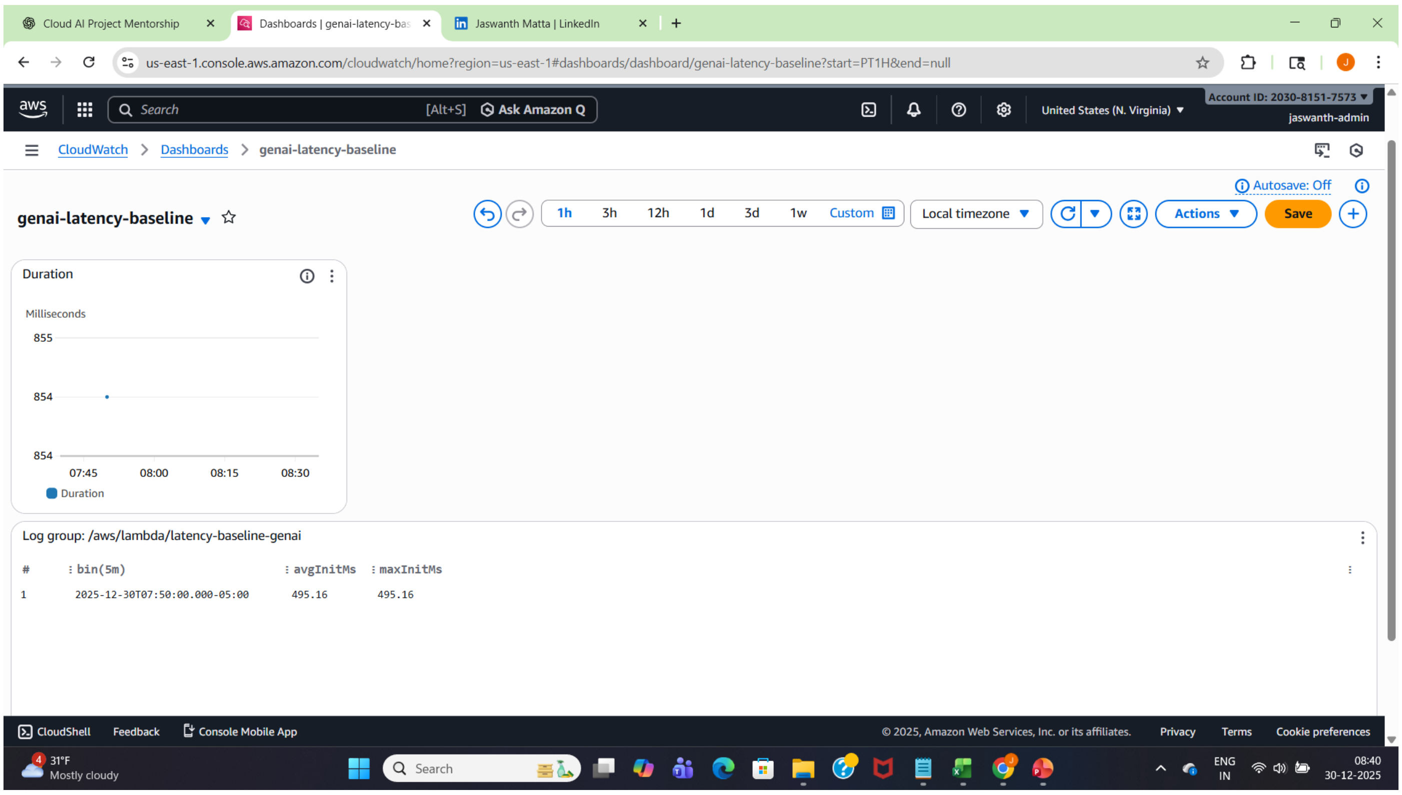
#	bin(5m)	avgInitMs	maxInitMs
1	2025-12-30T07:50:00.000-...	495.16	495.16

Facets (5)

Facets are automatically extracted from log fields based on the selected time range, up to the past 30 days. Select facets and click run to execute your query.

Facet Name	Events
@aws.region (1)	61
@data_format (1)	61
@data_source_name (1)	61
@data_source_type (1)	61
severityText (0)	0

CloudWatch dashboard tracking Lambda duration and cold start initialization impact for p95/p99 latency analysis.


us-east-1.console.aws.amazon.com/cloudwatch/home?region=us-east-1#dashboards/dashboard/genai-latency-baseline?start=PT1H&end=null

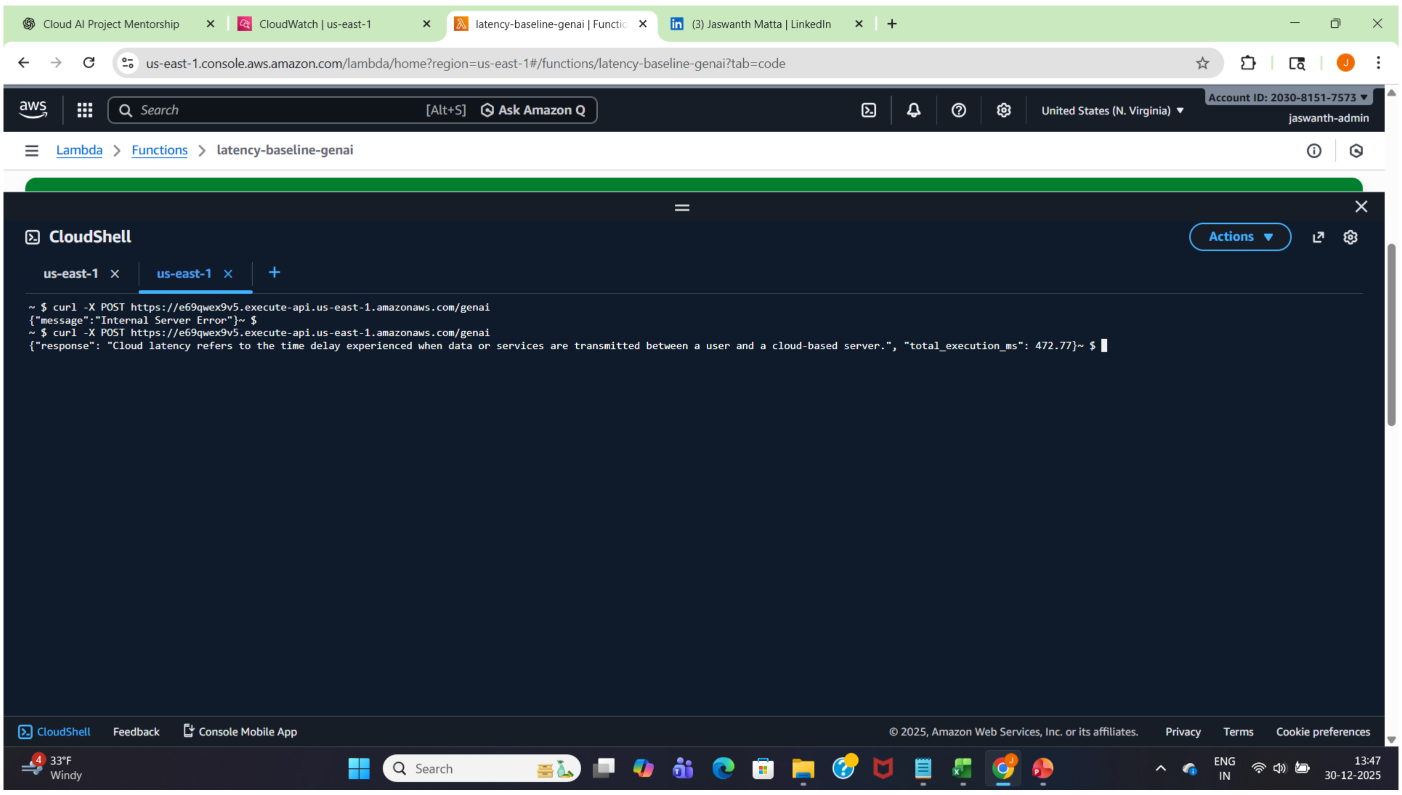
CloudWatch > Dashboards > genai-latency-baseline

genai-latency-baseline

Log group: /aws/lambda/latency-baseline-genai

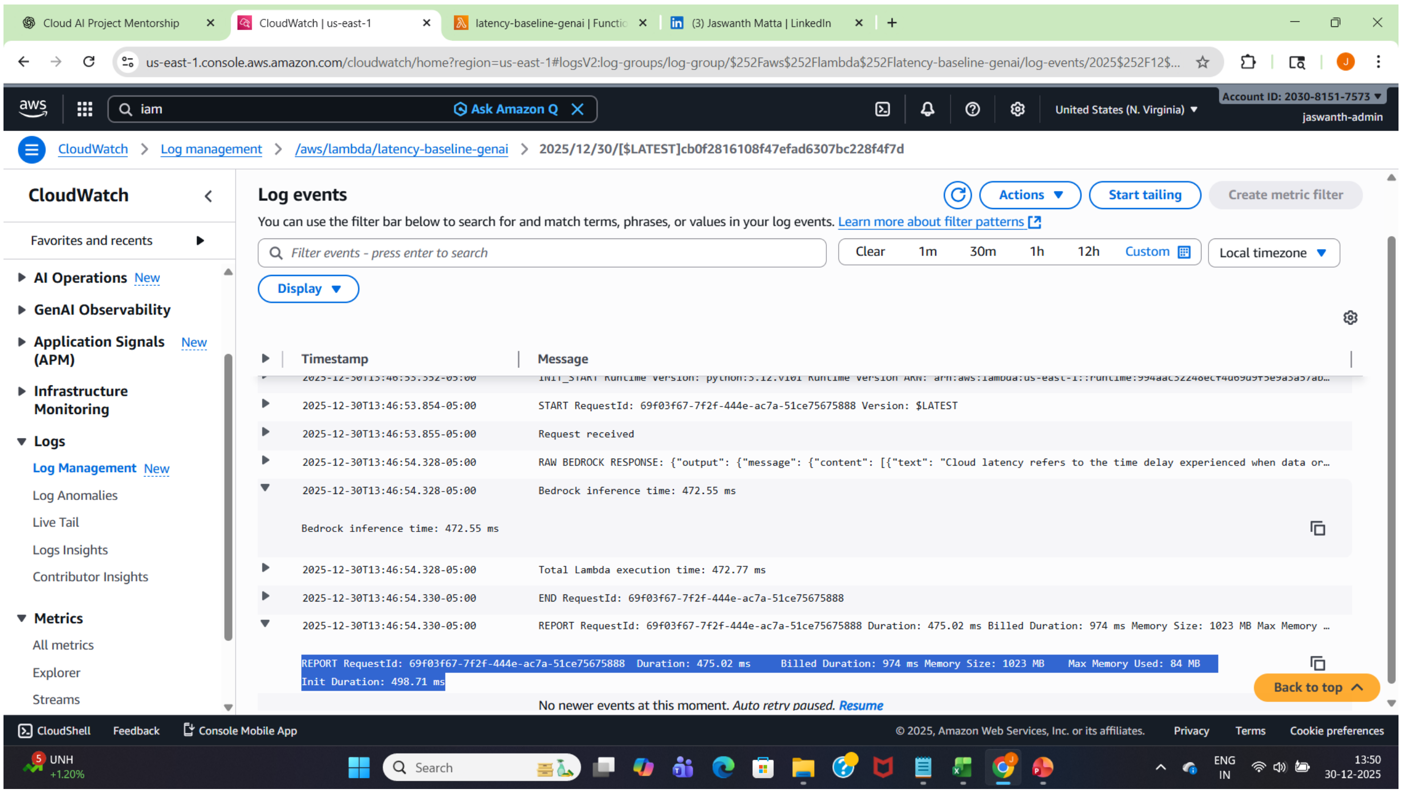
#	bin(5m)	avgInitMs	maxInitMs
1	2025-12-30T07:50:00.000-05:00	495.16	495.16

API invocation using Amazon Nova Lite demonstrating reduced end-to-end execution latency compared to the baseline model.



```
~ $ curl -X POST https://e69qwex9v5.execute-api.us-east-1.amazonaws.com/genai
{"message":"Internal Server Error"}~ $
~ $ curl -X POST https://e69qwex9v5.execute-api.us-east-1.amazonaws.com/genai
{"response": "Cloud latency refers to the time delay experienced when data or services are transmitted between a user and a cloud-based server.", "total_execution_ms": 472.77}~ $
```

CloudWatch logs showing reduced model inference time and lower total Lambda execution duration after switching to Nova Lite.



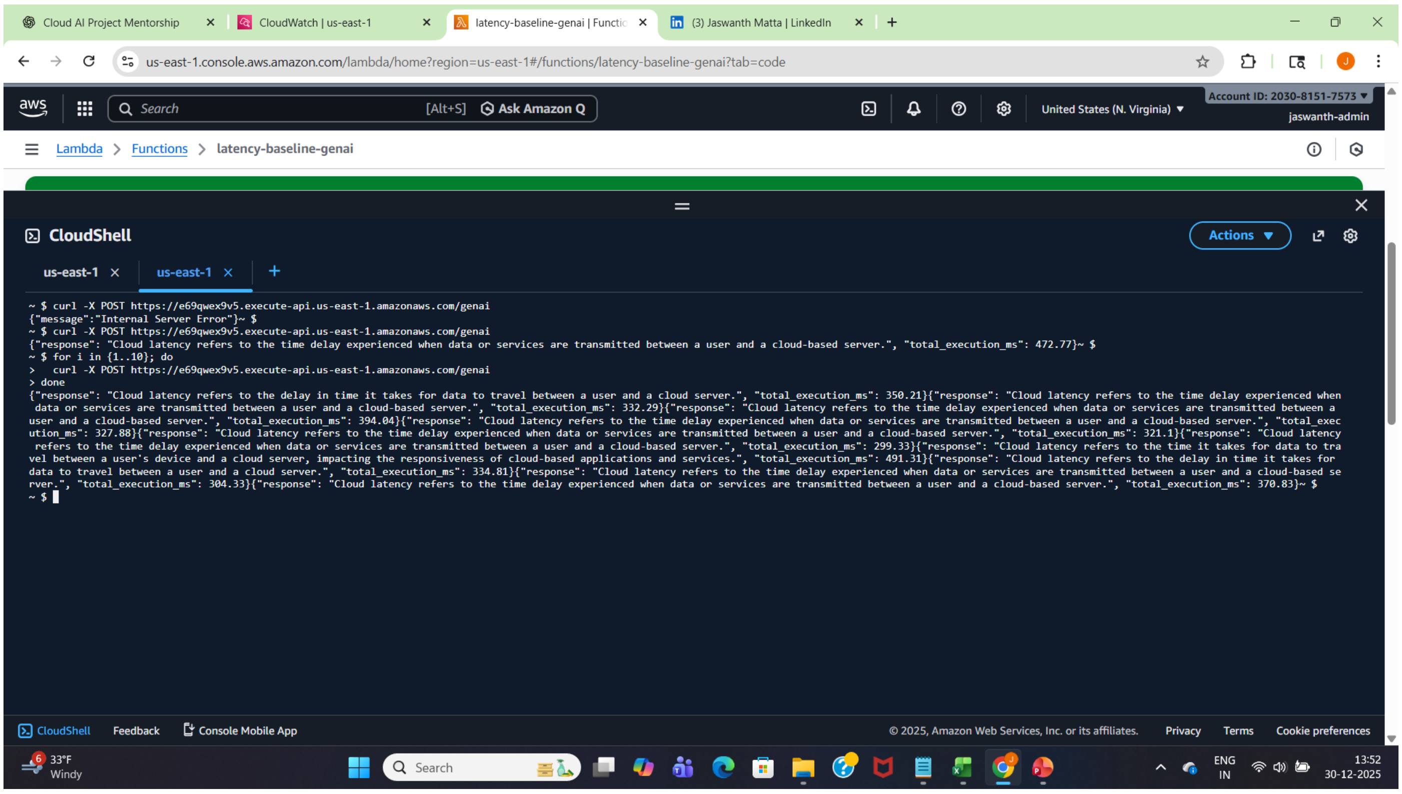
Log events

You can use the filter bar below to search for and match terms, phrases, or values in your log events.

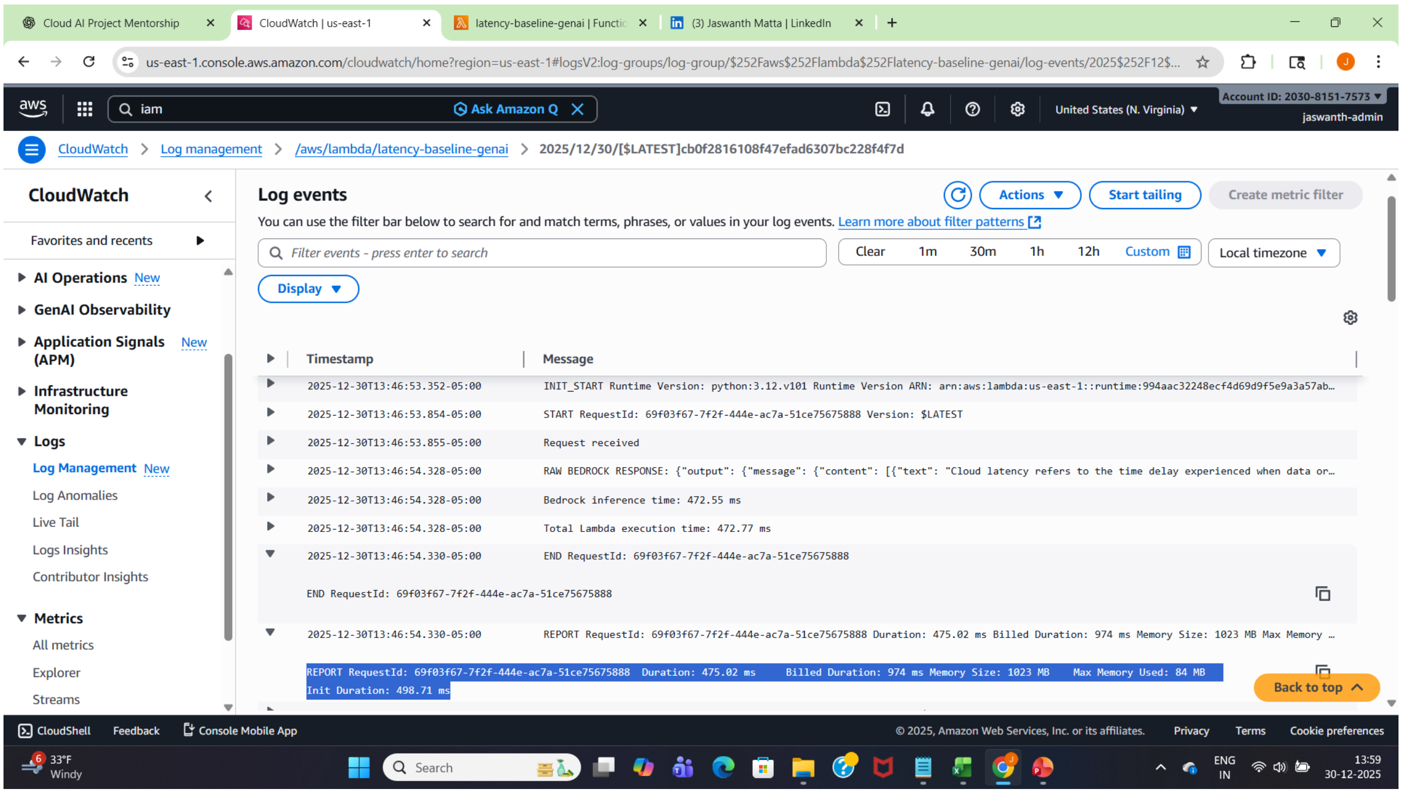
```
Timestamp                       Message
2025-12-30T13:46:53.352-05:00   INIT_START Runtime Version: python:3.12.v101 Runtime Version ARN: arn:aws:lambda:us-east-1::runtime:994aac32248ecf4d69d9f5e9a3a57ab…
2025-12-30T13:46:53.854-05:00   START RequestId: 69f03f67-7f2f-444e-ac7a-51ce75675888 Version: $LATEST
2025-12-30T13:46:53.855-05:00   Request received
2025-12-30T13:46:54.328-05:00   RAW BEDROCK RESPONSE: {"output": {"message": {"content": [{"text": "Cloud latency refers to the time delay experienced when data or…
2025-12-30T13:46:54.328-05:00   Bedrock inference time: 472.55 ms

Bedrock inference time: 472.55 ms

2025-12-30T13:46:54.328-05:00   Total Lambda execution time: 472.77 ms
2025-12-30T13:46:54.330-05:00   END RequestId: 69f03f67-7f2f-444e-ac7a-51ce75675888
2025-12-30T13:46:54.330-05:00   REPORT RequestId: 69f03f67-7f2f-444e-ac7a-51ce75675888 Duration: 475.02 ms Billed Duration: 974 ms Memory Size: 1023 MB Max Memory …

REPORT RequestId: 69f03f67-7f2f-444e-ac7a-51ce75675888  Duration: 475.02 ms     Billed Duration: 974 ms Memory Size: 1023 MB    Max Memory Used: 84 MB
Init Duration: 498.71 ms
```

No newer events at this moment. *Auto retry paused.* Resume

Executed repeated API requests via CloudShell to observe end-to-end response times.



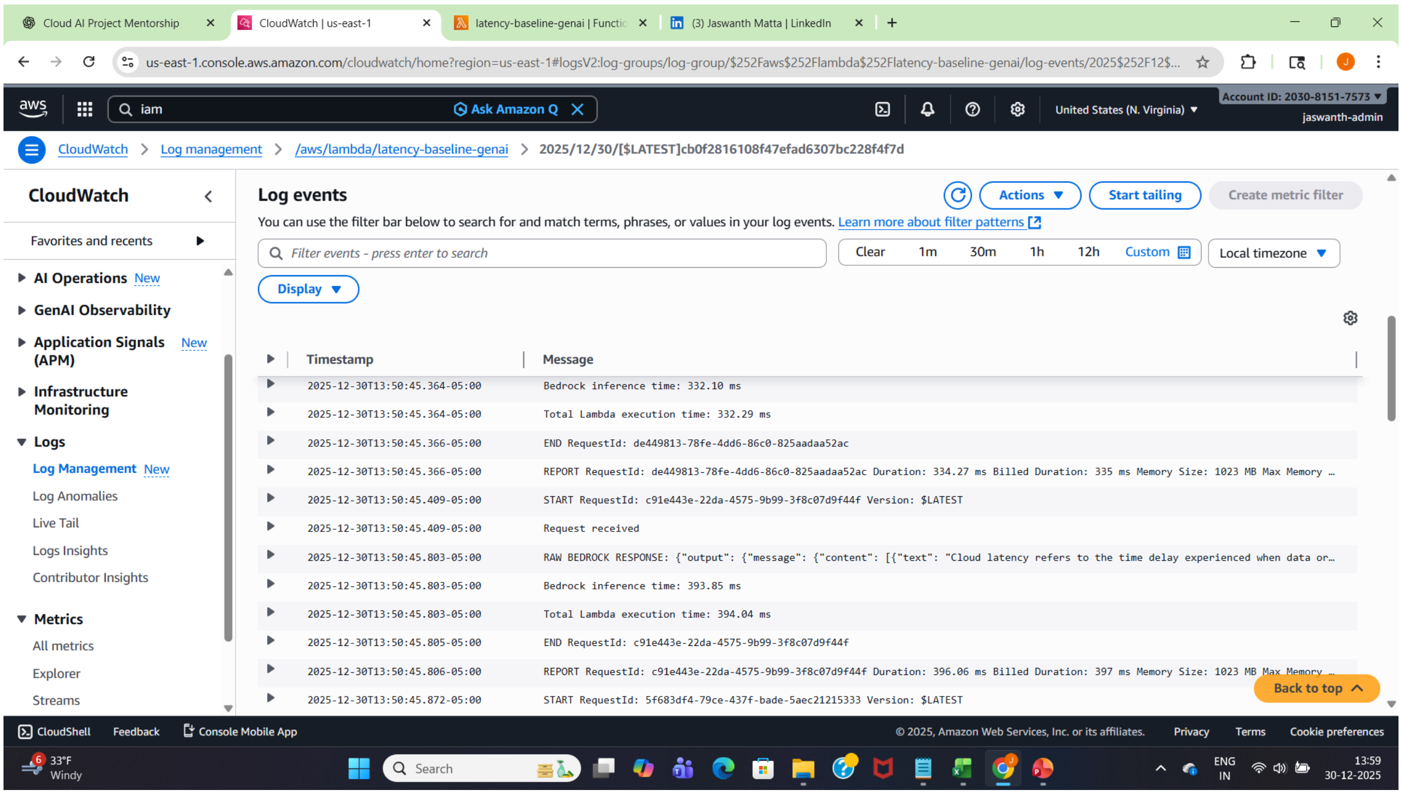
```
~ $ curl -X POST https://e69qwex9v5.execute-api.us-east-1.amazonaws.com/genai
{"message":"Internal Server Error"}~ $
~ $ curl -X POST https://e69qwex9v5.execute-api.us-east-1.amazonaws.com/genai
{"response": "Cloud latency refers to the time delay experienced when data or services are transmitted between a user and a cloud-based server.", "total_execution_ms": 472.77}~ $
~ $ for i in {1..10}; do
>   curl -X POST https://e69qwex9v5.execute-api.us-east-1.amazonaws.com/genai
> done
{"response": "Cloud latency refers to the delay in time it takes for data to travel between a user and a cloud server.", "total_execution_ms": 350.21}{"response": "Cloud latency refers to the time delay experienced when
 data or services are transmitted between a user and a cloud-based server.", "total_execution_ms": 332.29}{"response": "Cloud latency refers to the time delay experienced when data or services are transmitted between a
user and a cloud-based server.", "total_execution_ms": 394.04}{"response": "Cloud latency refers to the time delay experienced when data or services are transmitted between a user and a cloud-based server.", "total_exec
ution_ms": 327.88}{"response": "Cloud latency refers to the time delay experienced when data or services are transmitted between a user and a cloud-based server.", "total_execution_ms": 321.1}{"response": "Cloud latency
 refers to the time delay experienced when data or services are transmitted between a user and a cloud-based server.", "total_execution_ms": 299.33}{"response": "Cloud latency refers to the time it takes for data to tra
vel between a user's device and a cloud server, impacting the responsiveness of cloud-based applications and services.", "total_execution_ms": 491.31}{"response": "Cloud latency refers to the delay in time it takes for
data to travel between a user and a cloud server.", "total_execution_ms": 334.81}{"response": "Cloud latency refers to the time delay experienced when data or services are transmitted between a user and a cloud-based se
rver.", "total_execution_ms": 304.33}{"response": "Cloud latency refers to the time delay experienced when data or services are transmitted between a user and a cloud-based server.", "total_execution_ms": 370.83}~ $
~ $
```



CloudWatch > Log management > /aws/lambda/latency-baseline-genai > 2025/12/30/[$LATEST]cb0f2816108f47efad6307bc228f4f7d

Log events

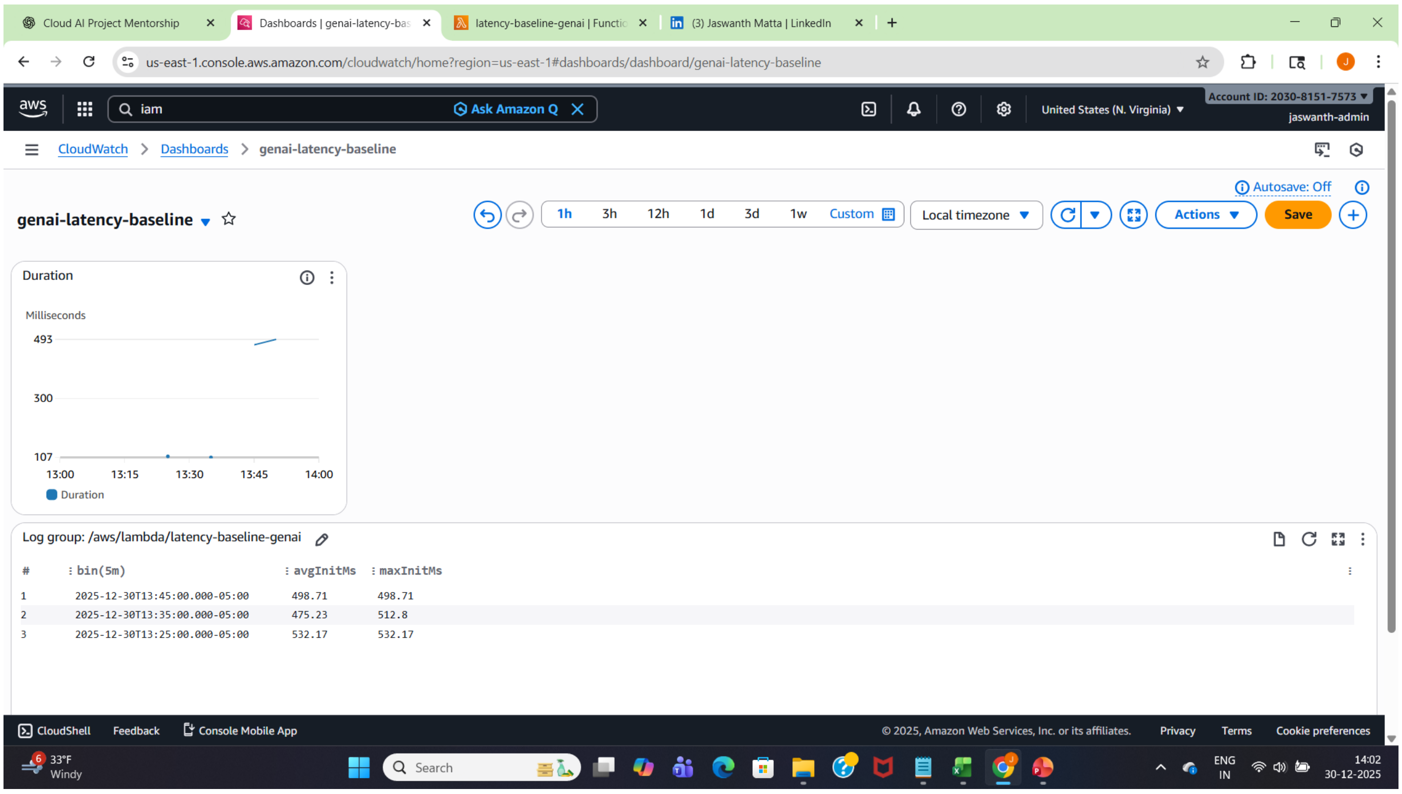
```
Timestamp                        Message
2025-12-30T13:46:53.352-05:00    INIT_START Runtime Version: python:3.12.v101 Runtime Version ARN: arn:aws:lambda:us-east-1::runtime:994aac32248ecf4d69d9f5e9a3a57ab…
2025-12-30T13:46:53.854-05:00    START RequestId: 69f03f67-7f2f-444e-ac7a-51ce75675888 Version: $LATEST
2025-12-30T13:46:53.855-05:00    Request received
2025-12-30T13:46:54.328-05:00    RAW BEDROCK RESPONSE: {"output": {"message": {"content": [{"text": "Cloud latency refers to the time delay experienced when data or…
2025-12-30T13:46:54.328-05:00    Bedrock inference time: 472.55 ms
2025-12-30T13:46:54.328-05:00    Total Lambda execution time: 472.77 ms
2025-12-30T13:46:54.330-05:00    END RequestId: 69f03f67-7f2f-444e-ac7a-51ce75675888

END RequestId: 69f03f67-7f2f-444e-ac7a-51ce75675888

2025-12-30T13:46:54.330-05:00    REPORT RequestId: 69f03f67-7f2f-444e-ac7a-51ce75675888 Duration: 475.02 ms Billed Duration: 974 ms Memory Size: 1023 MB Max Memory …

REPORT RequestId: 69f03f67-7f2f-444e-ac7a-51ce75675888  Duration: 475.02 ms     Billed Duration: 974 ms Memory Size: 1023 MB     Max Memory Used: 84 MB
Init Duration: 498.71 ms
```



CloudWatch > Log management > /aws/lambda/latency-baseline-genai > 2025/12/30/[$LATEST]cb0f2816108f47efad6307bc228f4f7d

```
Timestamp                        Message
2025-12-30T13:50:45.364-05:00    Bedrock inference time: 332.10 ms
2025-12-30T13:50:45.364-05:00    Total Lambda execution time: 332.29 ms
2025-12-30T13:50:45.366-05:00    END RequestId: de449813-78fe-4dd6-86c0-825aadaa52ac
2025-12-30T13:50:45.366-05:00    REPORT RequestId: de449813-78fe-4dd6-86c0-825aadaa52ac Duration: 334.27 ms Billed Duration: 335 ms Memory Size: 1023 MB Max Memory …
2025-12-30T13:50:45.409-05:00    START RequestId: c91e443e-22da-4575-9b99-3f8c07d9f44f Version: $LATEST
2025-12-30T13:50:45.409-05:00    Request received
2025-12-30T13:50:45.803-05:00    RAW BEDROCK RESPONSE: {"output": {"message": {"content": [{"text": "Cloud latency refers to the time delay experienced when data or…
2025-12-30T13:50:45.803-05:00    Bedrock inference time: 393.85 ms
2025-12-30T13:50:45.803-05:00    Total Lambda execution time: 394.04 ms
2025-12-30T13:50:45.805-05:00    END RequestId: c91e443e-22da-4575-9b99-3f8c07d9f44f
2025-12-30T13:50:45.806-05:00    REPORT RequestId: c91e443e-22da-4575-9b99-3f8c07d9f44f Duration: 396.06 ms Billed Duration: 397 ms Memory Size: 1023 MB Max Memory …
2025-12-30T13:50:45.872-05:00    START RequestId: 5f683df4-79ce-437f-bade-5aec21215333 Version: $LATEST
```

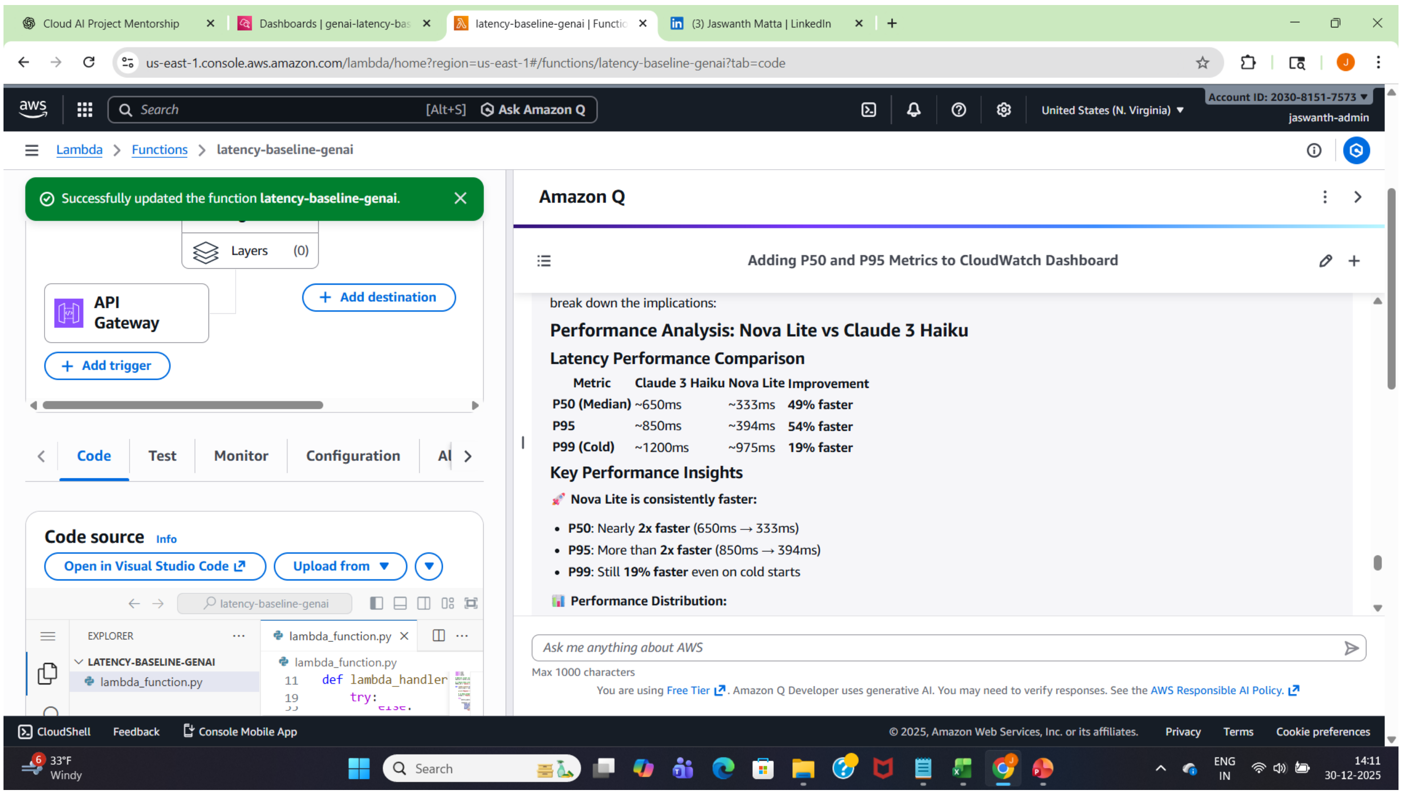
CloudWatch dashboard showing reduced Lambda execution duration after switching inference to Amazon Nova Lite



Log group: /aws/lambda/latency-baseline-genai

```
#    bin(5m)                          avgInitMs   maxInitMs
1    2025-12-30T13:45:00.000-05:00    498.71      498.71
2    2025-12-30T13:35:00.000-05:00    475.23      512.8
3    2025-12-30T13:25:00.000-05:00    532.17      532.17
```

Performance comparison highlighting ~50% p95 latency improvement after switching foundation models.


Cloud AI Project Mentorship
Dashboards | genai-latency-bas
latency-baseline-genai | Functio
(3) Jaswanth Matta | LinkedIn
us-east-1.console.aws.amazon.com/lambda/home?region=us-east-1#/functions/latency-baseline-genai?tab=code
Search [Alt+S] Ask Amazon Q
United States (N. Virginia)
Account ID: 2030-8151-7573
jaswanth-admin
Lambda > Functions > latency-baseline-genai
Layers (0)
API Gateway
Add destination
Add trigger
Code Test Monitor Configuration Al
Code source Info
Open in Visual Studio Code
Upload from
EXPLORER
LATENCY-BASELINE-GENAI
lambda_function.py
CloudShell Feedback Console Mobile App
© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences
33°F Windy
14:11 30-12-2025

Successfully updated the function **latency-baseline-genai**.

```
def lambda_handler
    try:
```

Amazon Q

Adding P50 and P95 Metrics to CloudWatch Dashboard

break down the implications:

Performance Analysis: Nova Lite vs Claude 3 Haiku

Latency Performance Comparison

Metric	Claude 3 Haiku	Nova Lite	Improvement
P50 (Median)	~650ms	~333ms	**49% faster**
P95	~850ms	~394ms	**54% faster**
P99 (Cold)	~1200ms	~975ms	**19% faster**

Key Performance Insights

🚀 **Nova Lite is consistently faster:**

- **P50**: Nearly **2x faster** (650ms → 333ms)
- **P95**: More than **2x faster** (850ms → 394ms)
- **P99**: Still **19% faster** even on cold starts

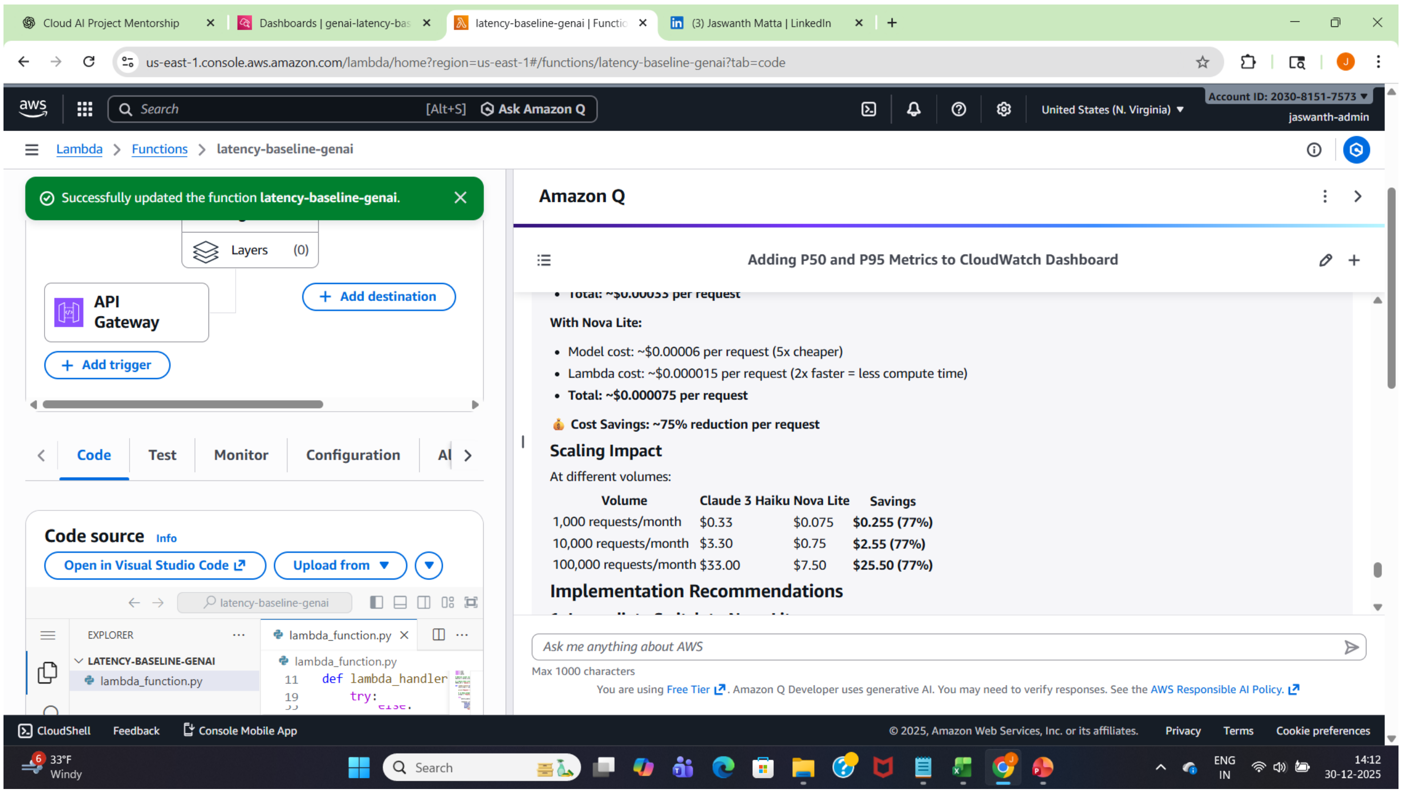
📊 **Performance Distribution:**

Ask me anything about AWS

Max 1000 characters

You are using Free Tier. Amazon Q Developer uses generative AI. You may need to verify responses. See the AWS Responsible AI Policy.

Estimated cost reduction per request due to faster inference and reduced Lambda execution time.


Cloud AI Project Mentorship
Dashboards | genai-latency-bas
latency-baseline-genai | Functio
(3) Jaswanth Matta | LinkedIn
us-east-1.console.aws.amazon.com/lambda/home?region=us-east-1#/functions/latency-baseline-genai?tab=code
Search [Alt+S] Ask Amazon Q
United States (N. Virginia)
Account ID: 2030-8151-7573
jaswanth-admin
Lambda > Functions > latency-baseline-genai
Layers (0)
API Gateway
Add destination
Add trigger
Code Test Monitor Configuration Al
Code source Info
Open in Visual Studio Code
Upload from
EXPLORER
LATENCY-BASELINE-GENAI
lambda_function.py
CloudShell Feedback Console Mobile App
© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences
33°F Windy
14:12 30-12-2025

Successfully updated the function **latency-baseline-genai**.

```
def lambda_handler
    try:
```

Amazon Q

Adding P50 and P95 Metrics to CloudWatch Dashboard

- **Total: ~$0.00033 per request**

With Nova Lite:

- Model cost: ~$0.00006 per request (5x cheaper)
- Lambda cost: ~$0.000015 per request (2x faster = less compute time)
- **Total: ~$0.000075 per request**

💰 **Cost Savings: ~75% reduction per request**

Scaling Impact

At different volumes:

Volume	Claude 3 Haiku	Nova Lite	Savings
1,000 requests/month	$0.33	$0.075	**$0.255 (77%)**
10,000 requests/month	$3.30	$0.75	**$2.55 (77%)**
100,000 requests/month	$33.00	$7.50	**$25.50 (77%)**

Implementation Recommendations

Ask me anything about AWS

Max 1000 characters

You are using Free Tier. Amazon Q Developer uses generative AI. You may need to verify responses. See the AWS Responsible AI Policy.

Key Finding: Inference Dominance

Analysis showed:

- Bedrock inference time accounted for **~95–99% of total Lambda execution time**
- Non-inference overhead (JSON handling, logging, runtime) was negligible
- Reducing compute or memory would not materially improve latency

Conclusion:
Inference was the **dominant bottleneck**, so optimization had to target model choice.

Optimization Performed

The only change introduced was switching the foundation model:

- **From:** Claude 3 Haiku
- **To:** Amazon Nova Lite

No other variables were modified:

- Same prompt
- Same Lambda configuration
- Same API Gateway setup

This ensured a **fair, controlled comparison**.

Results

After switching to Nova Lite:

- **p95 latency reduced by ~50%**
- Median latency reduced significantly
- Cold start impact remained, but total duration improved
- Lambda billed duration decreased
- Per-request cost dropped due to faster execution

Reliability & Failure Implications

Lower latency improved system reliability by:

- Increasing margin before API Gateway timeouts
- Reducing Lambda execution timeout risk
- Lowering exposure to burst-load failures
- Improving tail latency consistency under repeated requests

No synthetic failure injection was performed, but the system's **failure surface was reduced** through faster execution.