Smart Service AI Assistant – Technical Documentation

Amazon Bedrock • Knowledge Bases • OpenSearch Serverless • Lambda • SES • Serverless AI Architecture

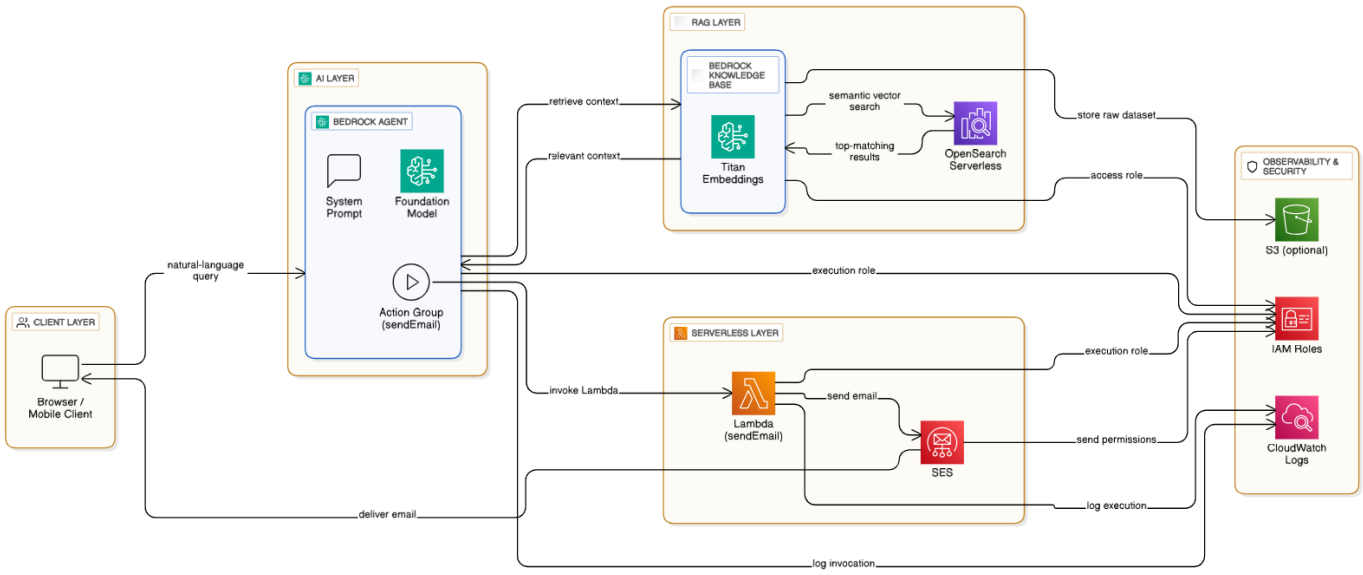
1. Executive Summary

The **Smart Service AI Assistant** is an intelligent, serverless, generative AI system built on **Amazon Bedrock Agents**, **Knowledge Bases**, **OpenSearch Serverless**, **AWS Lambda**, and **Amazon SES**.
The assistant interprets natural language questions, retrieves contextual information using RAG (Retrieval-Augmented Generation), generates conversational responses, and performs real-world actions such as sending emails.

The solution demonstrates how agentic AI, vector retrieval, and serverless compute can be combined to build production-grade intelligent assistants suitable for enterprise workflows, customer support automation, and personalized recommendation systems.

2. Architecture Overview

The solution is designed as a fully managed, serverless AI pipeline:



Core Architecture Components

Component	Purpose
Amazon Bedrock Agent	Interprets user queries, orchestrates tools, and generates human-like responses.
Knowledge Base (KB)	Stores unstructured text (food deals, restaurant offers, shopping/electronics insights). Uses embeddings for semantic search.
Amazon OpenSearch Serverless	Vector index backend that enables fast, relevant retrieval using approximate nearest neighbor (ANN) search.
Lambda Action Group	Custom tool integrated into the agent for executing backend logic (e.g., sending emails).
Amazon SES	Delivers formatted emails triggered by the AI assistant.
Amazon S3	Stores dataset files and KB documents for ingestion.
IAM Roles & Policies	Secures access across all services using least-privilege design.

The architecture is fully serverless, automatically scalable, and cost-optimized.

---

3. System Workflow

End-to-End Request Flow

1. **User Query**
   The user asks: *"Find me food deals and send them to my email."*
2. **Bedrock Agent Invocation**
   The agent parses intent and determines required tools (RAG + email action).
3. **RAG Retrieval via Knowledge Base**
   - Query is embedded using the model.
   - Embedding is matched against documents stored in **OpenSearch Serverless**.
   - Relevant offers/deals are retrieved.
4. **Generative Response**
   Bedrock synthesizes a clean, conversational answer without citations or technical metadata.
5. **Action Group Execution**
   If the user requests "send it to my email," the agent triggers the **sendEmail** tool.
6. **Lambda Action Group**
   - Lambda receives structured parameters from the agent.
   - The message is formatted into HTML/text.
   - SES is invoked to send the email.

7. **SES Email Delivery**
   Email is delivered securely to the user.
8. **Final AI Response**
   The agent confirms delivery and returns the final message.

---

4. Key Features & Capabilities

- Natural Language Understanding using Amazon Bedrock foundation models
- RAG-powered retrieval using **vector embeddings**
- Clean, citation-free responses for production use
- Automatic semantic search for unstructured content (deals, offers, recommendations)
- Custom **Lambda Action Groups** for business logic
- Secure, automated email delivery using Amazon SES
- Fully serverless, scalable architecture
- Plug-and-play integration for mobile/web applications

---

5. Technologies & AWS Services Used

- **Amazon Bedrock** (Agents, Knowledge Bases, Embeddings)
- **Amazon OpenSearch Serverless** (Vector Storage & Indexing)
- **AWS Lambda** (Action Group backend)
- **Amazon SES** (Transactional email delivery)
- **Amazon S3** (Document storage for KB ingestion)
- **IAM** (Fine-grained role-based access control)
- **Python** (Lambda handler code)

---

6. Troubleshooting & Challenges Resolved

1. Knowledge Base Indexing Failures

- Issue: Incorrect embedding model configuration and OpenSearch vector settings
- Fix: Updated space_type, dimensions, and vector field mappings in the KB configuration

2. Lambda Permission Errors

- Issue: Lambda could not call SES

- Fix: Added ses:SendEmail and ses:SendRawEmail IAM permissions

3. Retrieval Quality Issues

- Issue: KB returned irrelevant or empty results
- Fix: Improved dataset formatting and removed noisy text

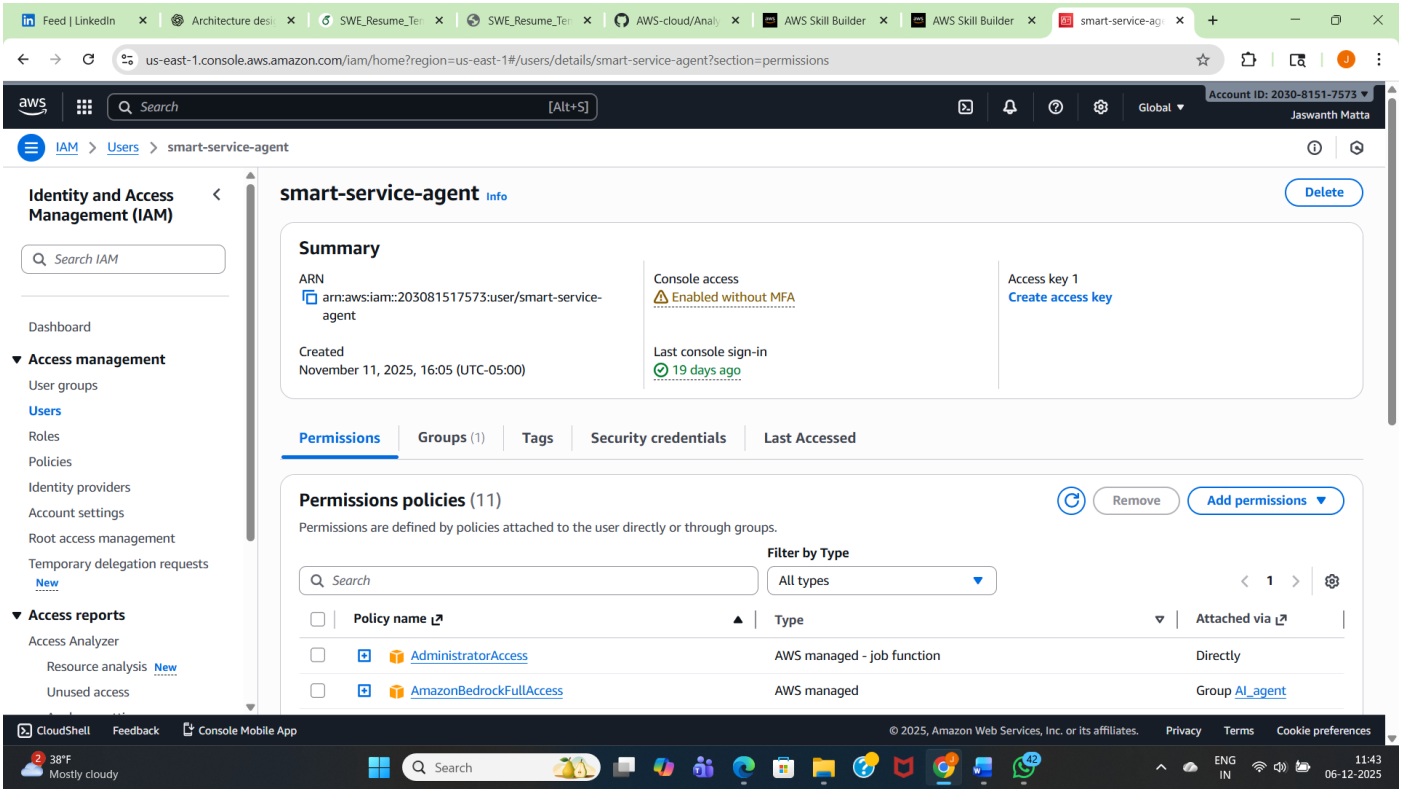
4. Agent Tool Invocation Not Triggering

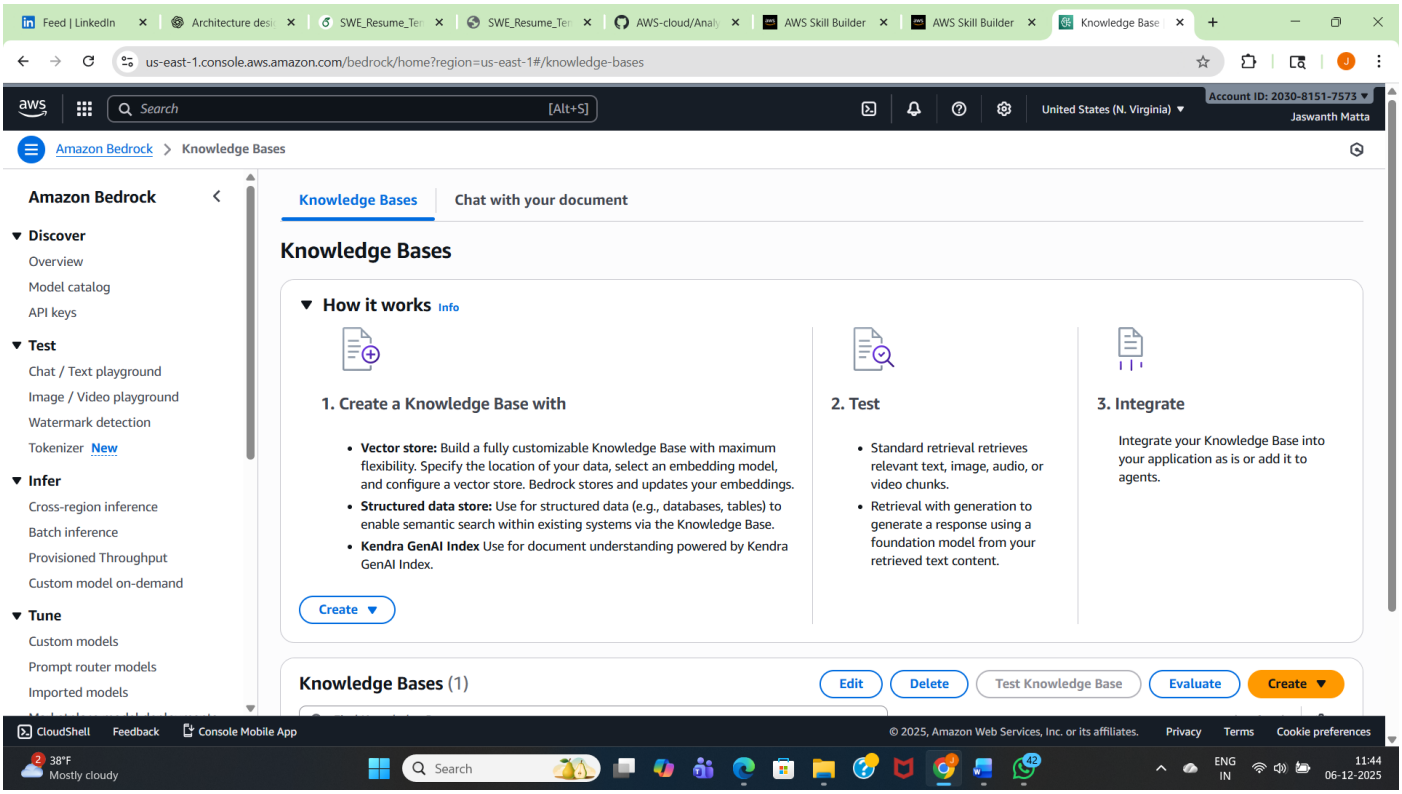
- Issue: Improper action schema in Bedrock Agent
- Fix: Redefined JSON schema and validated parameter mapping

These represent real engineering debugging scenarios that demonstrate production-level troubleshooting.

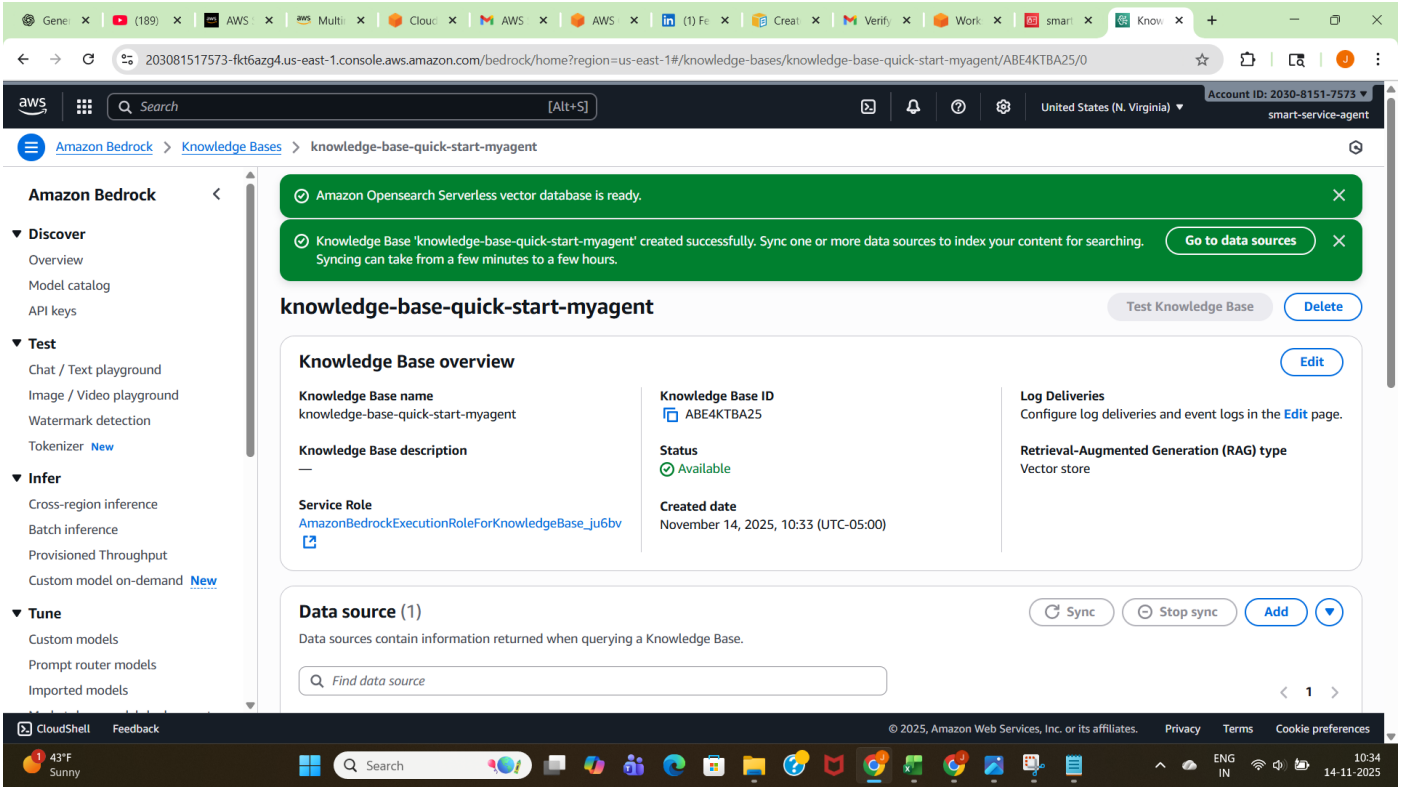
7.Implementation Screenshots

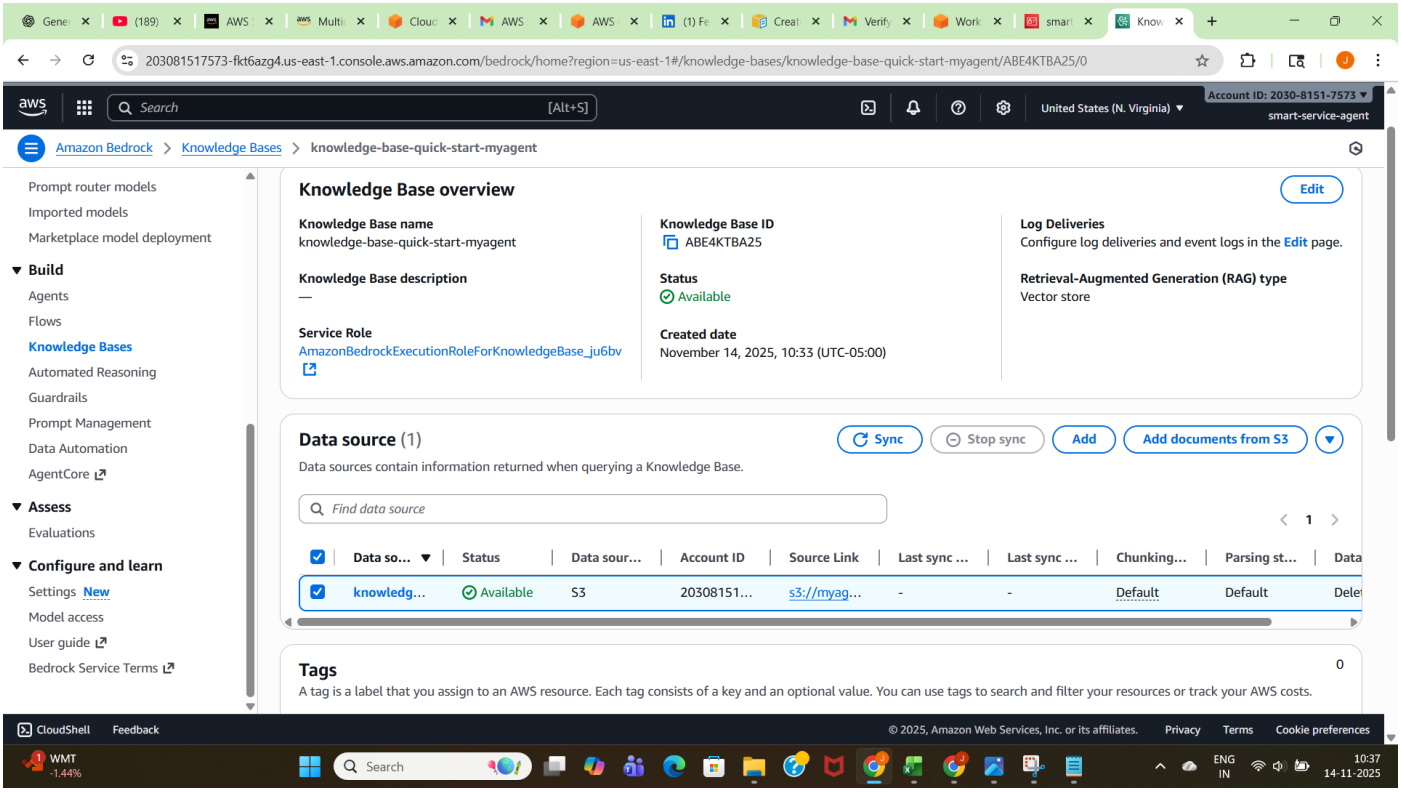
This screenshot shows the IAM user **smart-service-agent** configured with the required permissions to support Bedrock Agent operations. The attached policies provide secure access to Bedrock, Lambda, and other necessary AWS services.

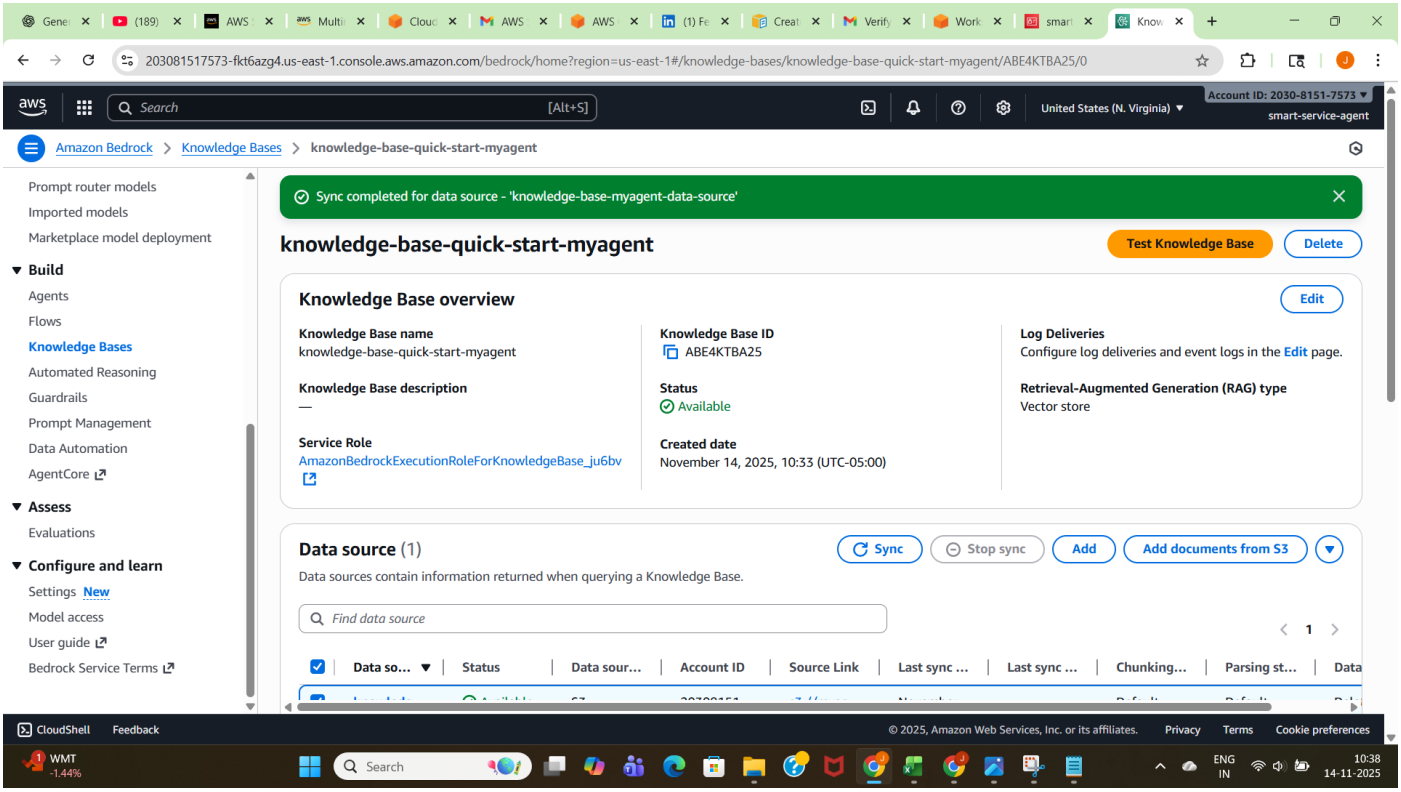


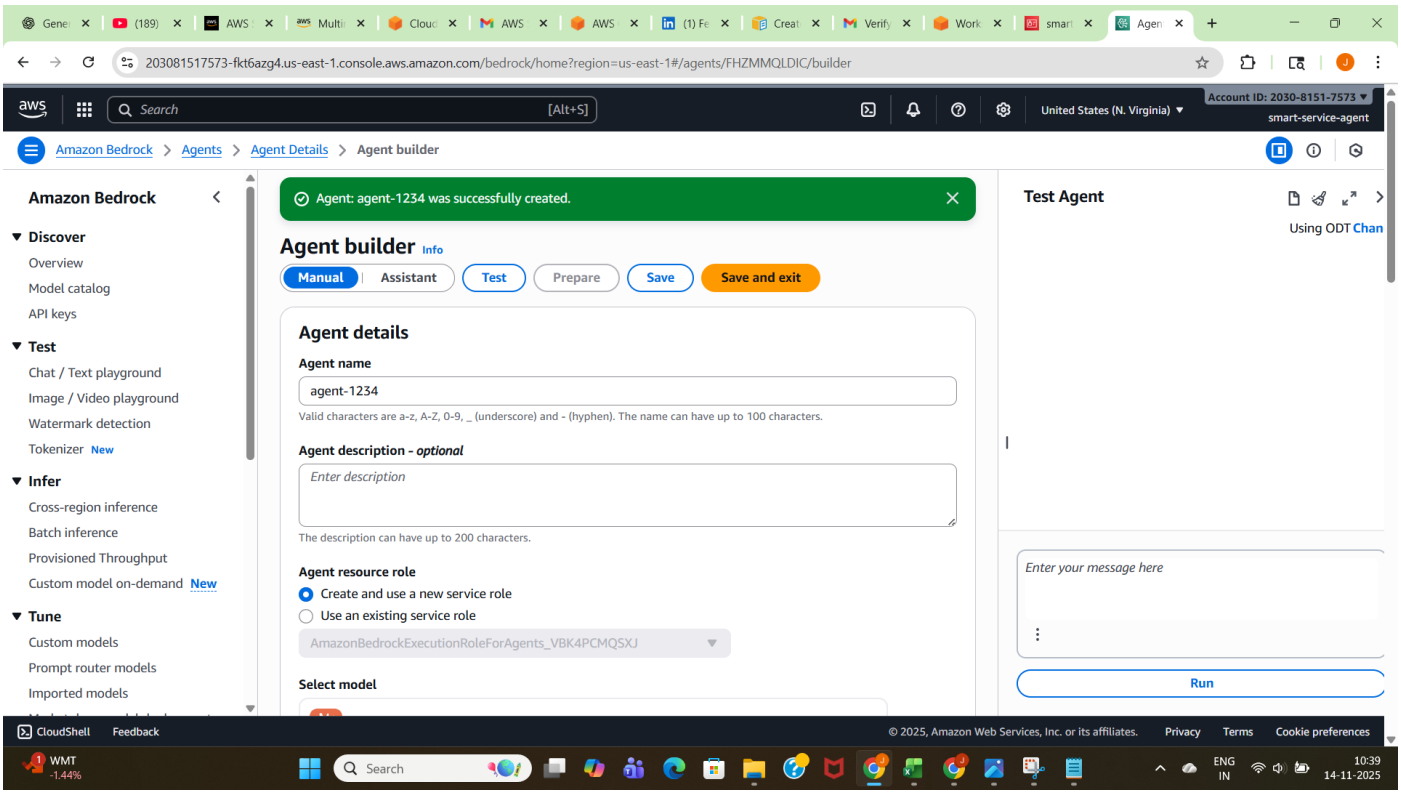


This screenshot shows the Bedrock Knowledge Base successfully created and synced with an OpenSearch Serverless vector database. The Knowledge Base is now in an **Available** state, ready to perform semantic retrieval for RAG-based queries.

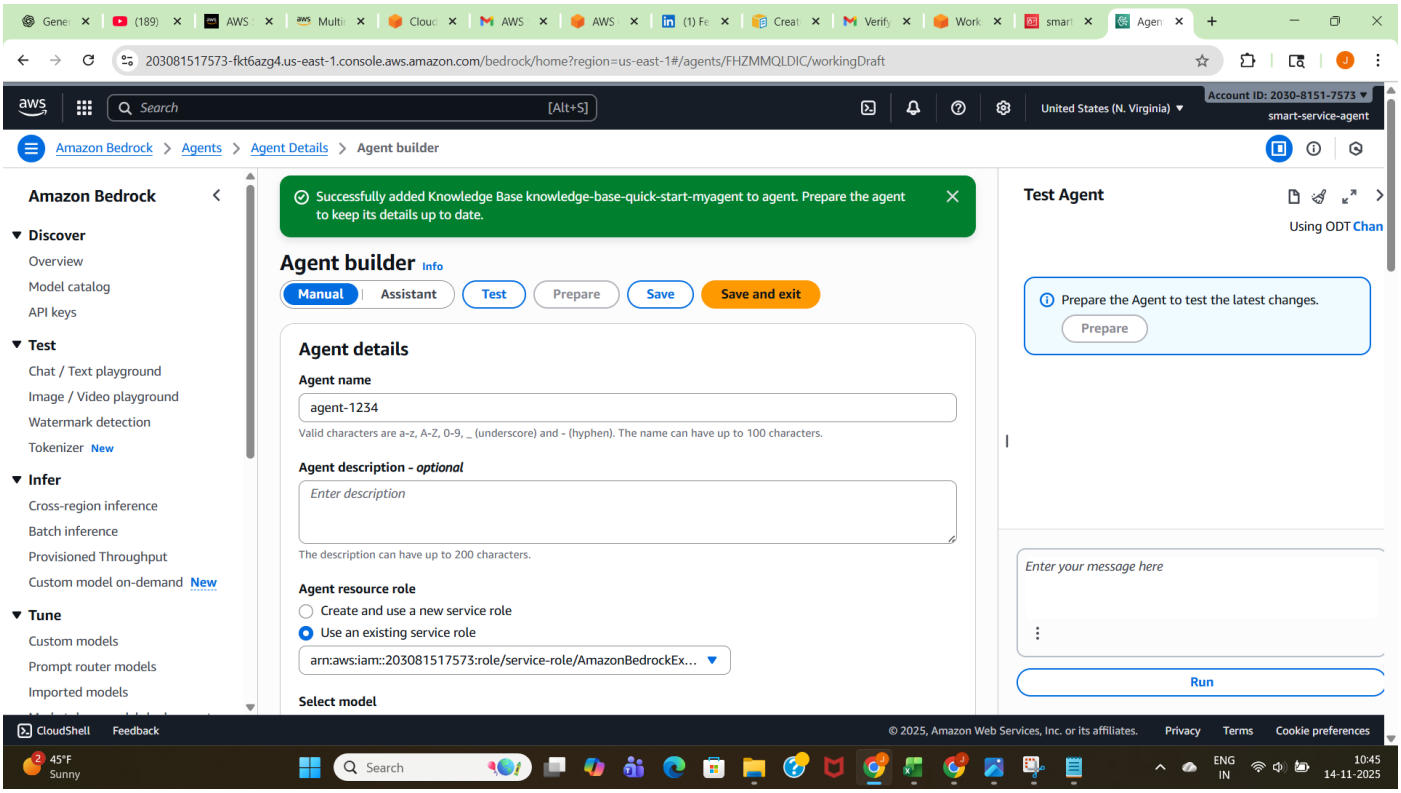


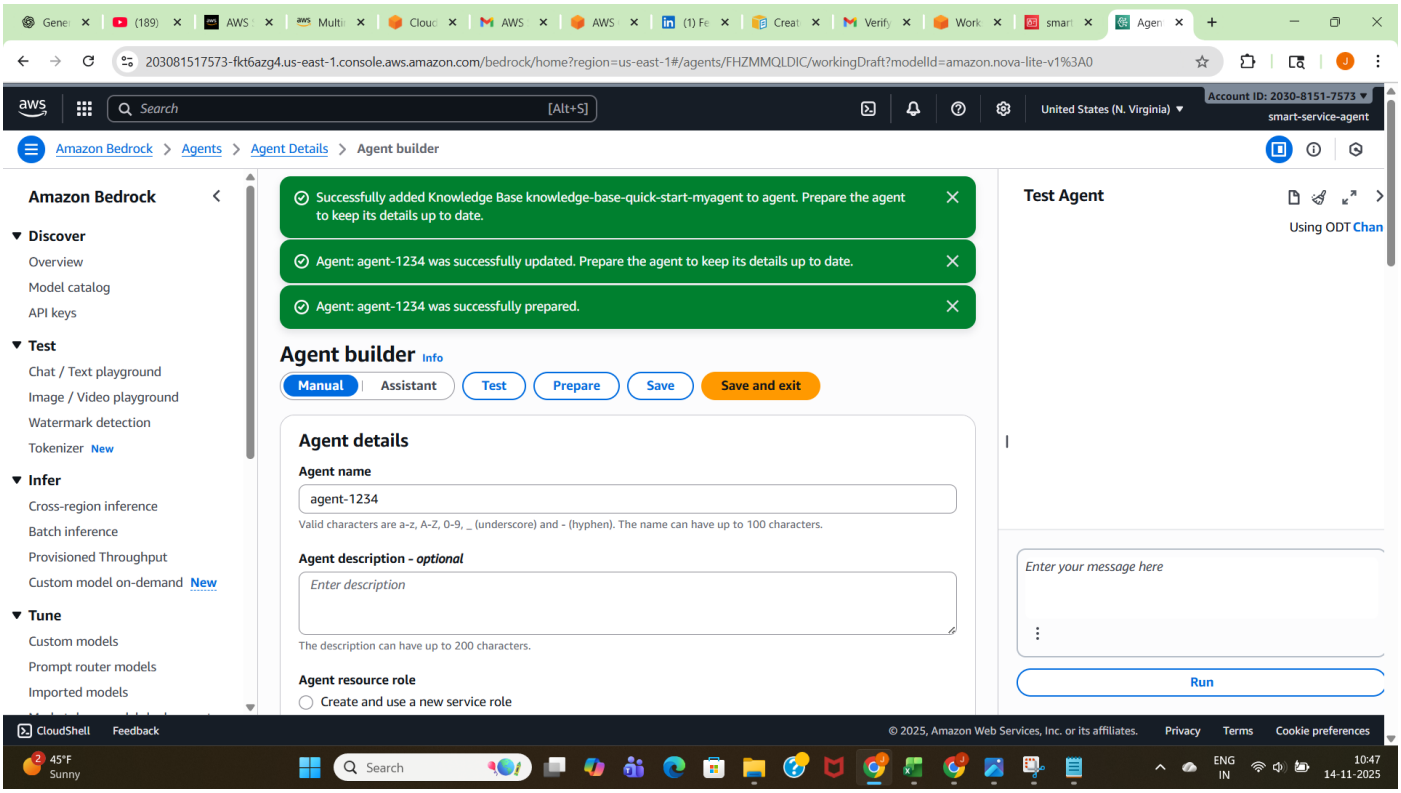

203081517573-fkt6azg4.us-east-1.console.aws.amazon.com/bedrock/home?region=us-east-1#/knowledge-bases/knowledge-base-quick-start-myagent/ABE4KTBA25/0
Search [Alt+S]
United States (N. Virginia)
Account ID: 2030-8151-7573
smart-service-agent
Amazon Bedrock > Knowledge Bases > knowledge-base-quick-start-myagent
Prompt router models
Imported models
Marketplace model deployment
Build
Agents
Flows
Knowledge Bases
Automated Reasoning
Guardrails
Prompt Management
Data Automation
AgentCore
Assess
Evaluations
Configure and learn
Settings New
Model access
User guide
Bedrock Service Terms
Knowledge Base overview
Edit
Knowledge Base name
knowledge-base-quick-start-myagent
Knowledge Base description
—
Service Role
AmazonBedrockExecutionRoleForKnowledgeBase_ju6bv
Knowledge Base ID
ABE4KTBA25
Status
Available
Created date
November 14, 2025, 10:33 (UTC-05:00)
Log Deliveries
Configure log deliveries and event logs in the Edit page.
Retrieval-Augmented Generation (RAG) type
Vector store
Data source (1)
Sync
Stop sync
Add
Add documents from S3
Data sources contain information returned when querying a Knowledge Base.
Find data source
Data so...
Status
Data sour...
Account ID
Source Link
Last sync ...
Last sync ...
Chunking...
Parsing st...
knowledg...
Available
S3
20308151...
s3://myag...
Default
Default
Tags
0
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
WMT -1.44%
Search
ENG IN
10:37
14-11-2025

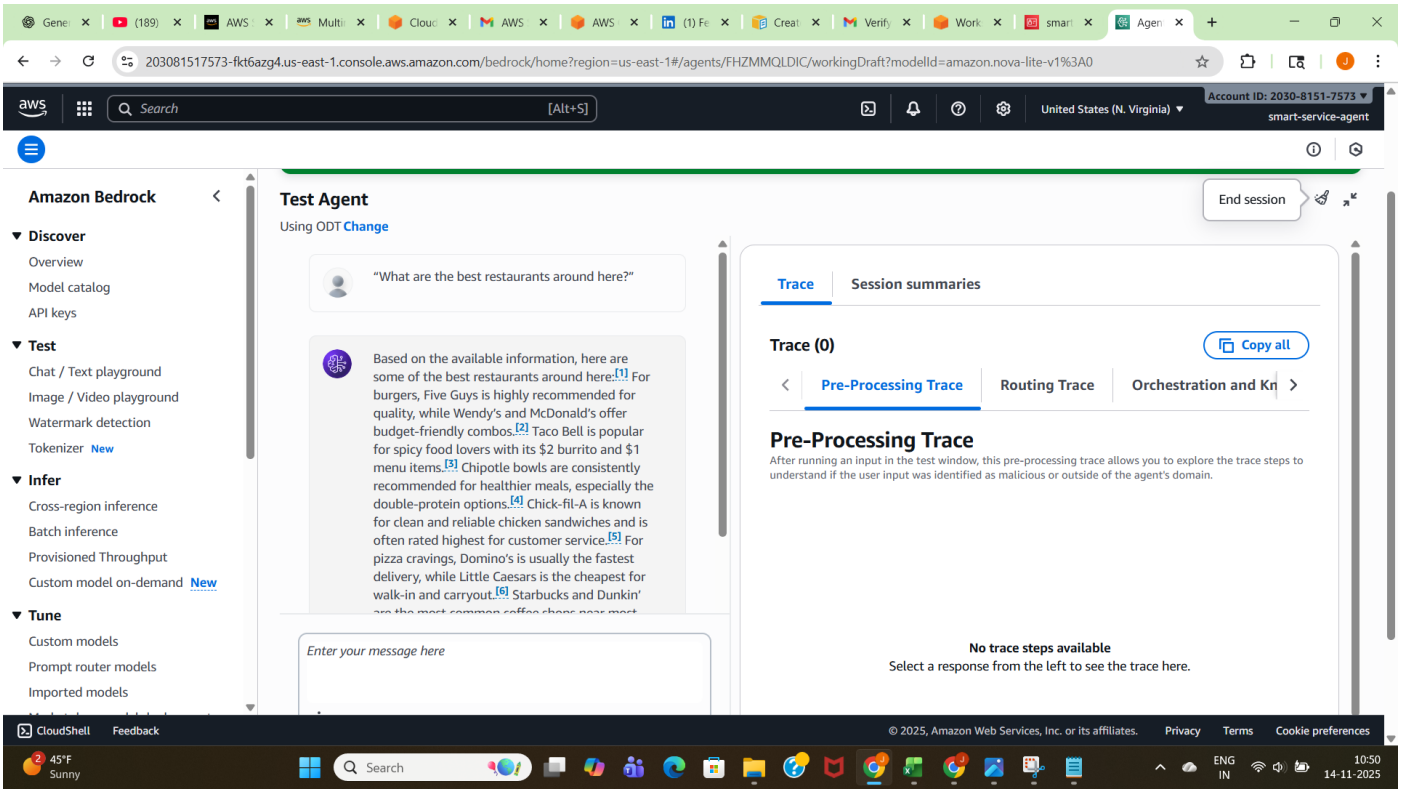

203081517573-fkt6azg4.us-east-1.console.aws.amazon.com/bedrock/home?region=us-east-1#/knowledge-bases/knowledge-base-quick-start-myagent/ABE4KTBA25/0
Search [Alt+S]
United States (N. Virginia)
Account ID: 2030-8151-7573
smart-service-agent
Amazon Bedrock > Knowledge Bases > knowledge-base-quick-start-myagent
Sync completed for data source - 'knowledge-base-myagent-data-source'
knowledge-base-quick-start-myagent
Test Knowledge Base
Delete
Prompt router models
Imported models
Marketplace model deployment
Build
Agents
Flows
Knowledge Bases
Automated Reasoning
Guardrails
Prompt Management
Data Automation
AgentCore
Assess
Evaluations
Configure and learn
Settings New
Model access
User guide
Bedrock Service Terms
Knowledge Base overview
Edit
Knowledge Base name
knowledge-base-quick-start-myagent
Knowledge Base description
—
Service Role
AmazonBedrockExecutionRoleForKnowledgeBase_ju6bv
Knowledge Base ID
ABE4KTBA25
Status
Available
Created date
November 14, 2025, 10:33 (UTC-05:00)
Log Deliveries
Configure log deliveries and event logs in the Edit page.
Retrieval-Augmented Generation (RAG) type
Vector store
Data source (1)
Sync
Stop sync
Add
Add documents from S3
Data sources contain information returned when querying a Knowledge Base.
Find data source
Data so...
Status
Data sour...
Account ID
Source Link
Last sync ...
Last sync ...
Chunking...
Parsing st...
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
WMT -1.44%
Search
ENG IN
10:38
14-11-2025

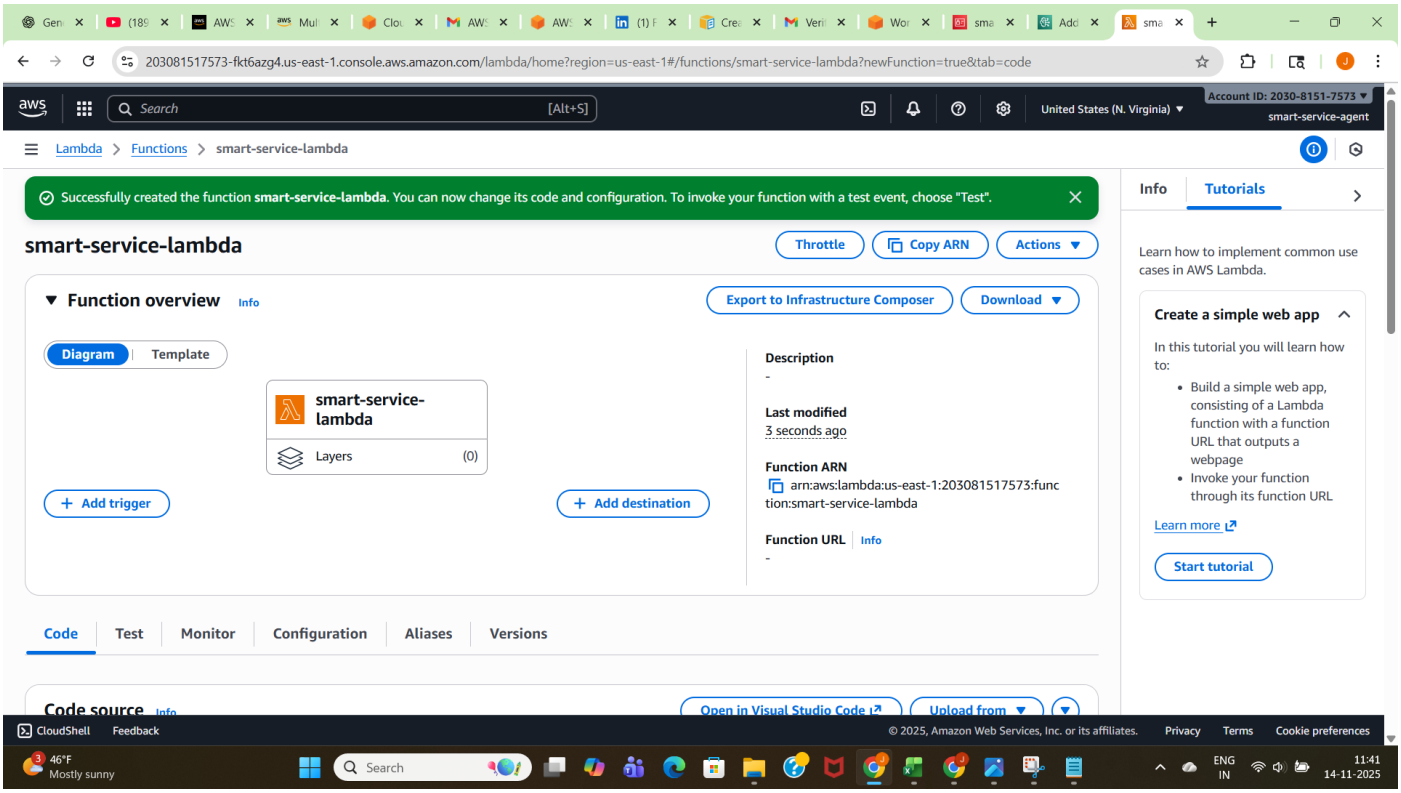


This screenshot shows the Bedrock Agent being configured with the newly added Knowledge Base, enabling the agent to retrieve context using RAG during conversation. The agent is assigned the required execution role and is ready for preparation and testing.

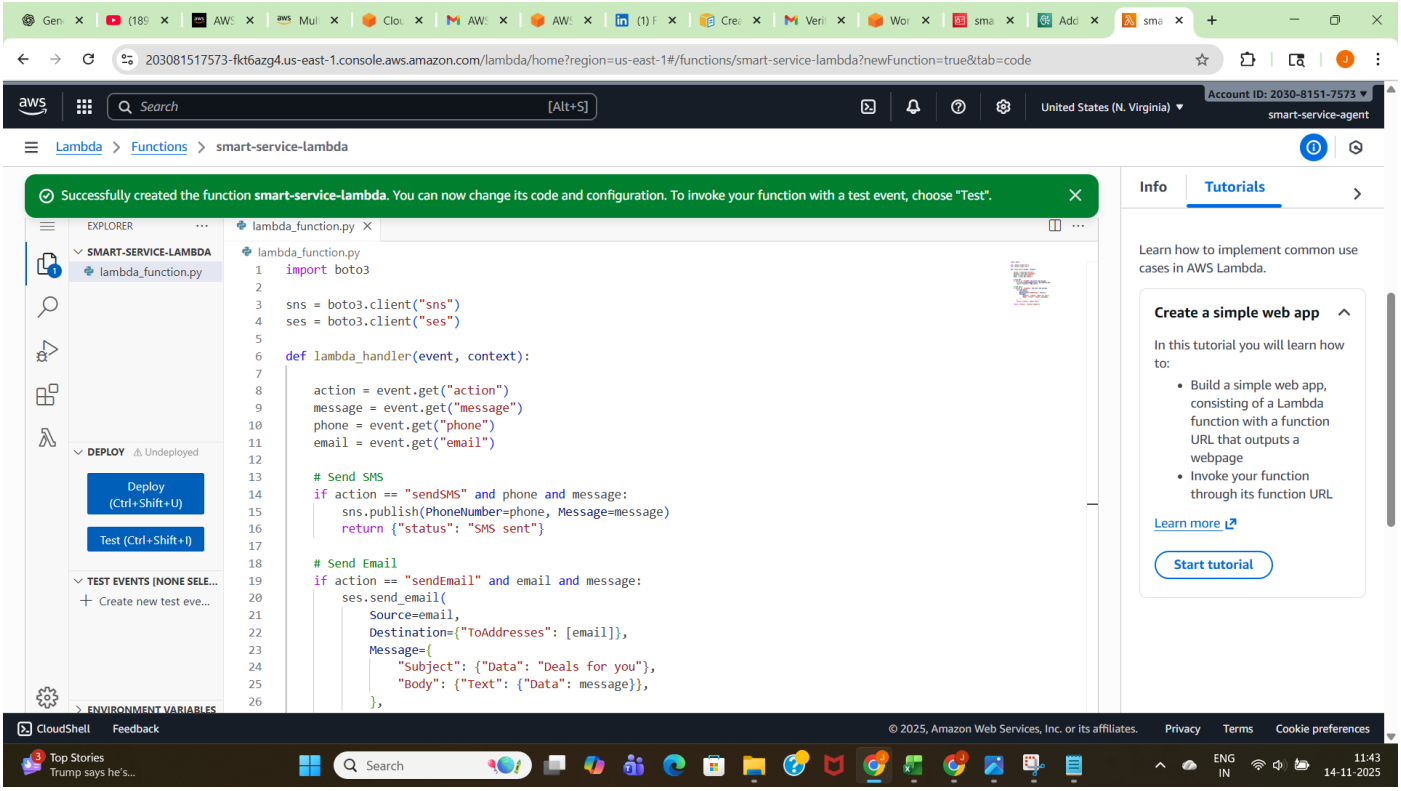


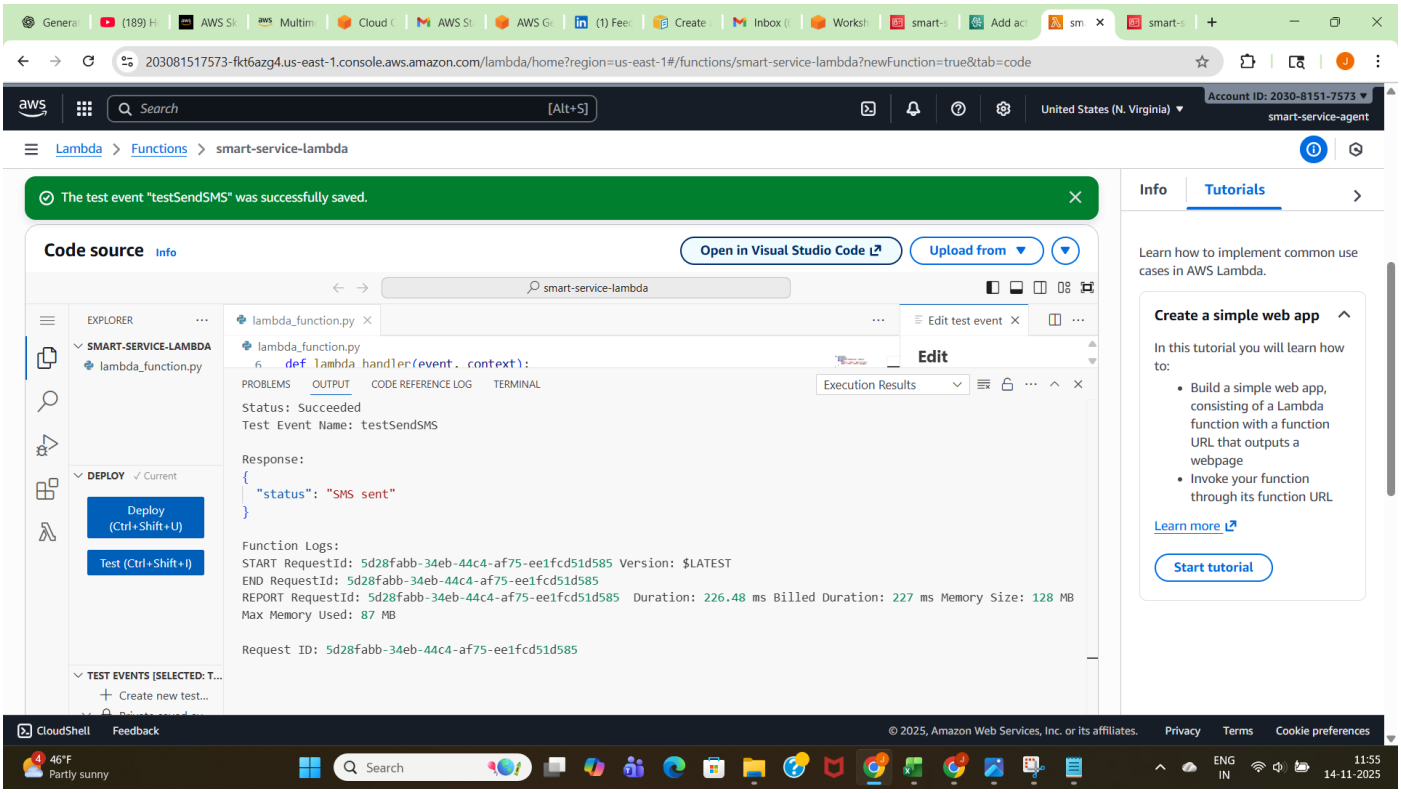

203081517573-fkt6azg4.us-east-1.console.aws.amazon.com/bedrock/home?region=us-east-1#/agents/FHZMMQLDIC/workingDraft?modelId=amazon.nova-lite-v1%3A0
Search [Alt+S]
United States (N. Virginia)
Account ID: 2030-8151-7573
smart-service-agent
Amazon Bedrock > Agents > Agent Details > Agent builder
Amazon Bedrock
Discover
Overview
Model catalog
API keys
Test
Chat / Text playground
Image / Video playground
Watermark detection
Tokenizer New
Infer
Cross-region inference
Batch inference
Provisioned Throughput
Custom model on-demand New
Tune
Custom models
Prompt router models
Imported models
Successfully added Knowledge Base knowledge-base-quick-start-myagent to agent. Prepare the agent to keep its details up to date.
Agent: agent-1234 was successfully updated. Prepare the agent to keep its details up to date.
Agent: agent-1234 was successfully prepared.
Agent builder Info
Manual
Assistant
Test
Prepare
Save
Save and exit
Agent details
Agent name
agent-1234
Valid characters are a-z, A-Z, 0-9, _ (underscore) and - (hyphen). The name can have up to 100 characters.
Agent description - optional
Enter description
The description can have up to 200 characters.
Agent resource role
Create and use a new service role
Test Agent
Using ODT Chan
Enter your message here
Run
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
45°F
Sunny
Search
ENG
IN
10:47
14-11-2025

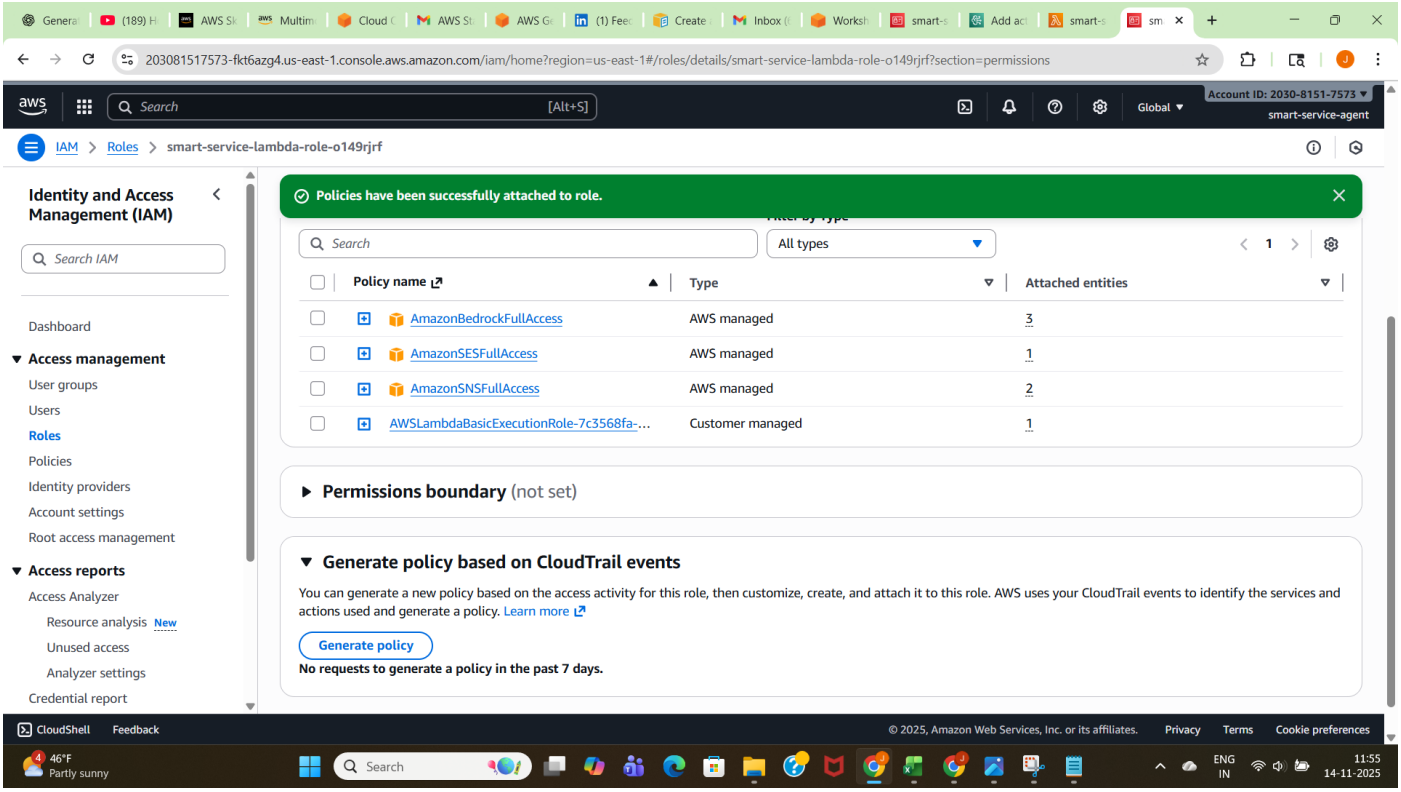

203081517573-fkt6azg4.us-east-1.console.aws.amazon.com/bedrock/home?region=us-east-1#/agents/FHZMMQLDIC/workingDraft?modelId=amazon.nova-lite-v1%3A0
Search [Alt+S]
United States (N. Virginia)
Account ID: 2030-8151-7573
smart-service-agent
Amazon Bedrock
Discover
Overview
Model catalog
API keys
Test
Chat / Text playground
Image / Video playground
Watermark detection
Tokenizer New
Infer
Cross-region inference
Batch inference
Provisioned Throughput
Custom model on-demand New
Tune
Custom models
Prompt router models
Imported models
Test Agent
Using ODT Change
End session
"What are the best restaurants around here?"
Based on the available information, here are some of the best restaurants around here:[1] For burgers, Five Guys is highly recommended for quality, while Wendy's and McDonald's offer budget-friendly combos.[2] Taco Bell is popular for spicy food lovers with its $2 burrito and $1 menu items.[3] Chipotle bowls are consistently recommended for healthier meals, especially the double-protein options.[4] Chick-fil-A is known for clean and reliable chicken sandwiches and is often rated highest for customer service.[5] For pizza cravings, Domino's is usually the fastest delivery, while Little Caesars is the cheapest for walk-in and carryout.[6] Starbucks and Dunkin'
Enter your message here
Trace
Session summaries
Trace (0)
Copy all
Pre-Processing Trace
Routing Trace
Orchestration and Kn
Pre-Processing Trace
After running an input in the test window, this pre-processing trace allows you to explore the trace steps to understand if the user input was identified as malicious or outside of the agent's domain.
No trace steps available
Select a response from the left to see the trace here.
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
45°F
Sunny
Search
ENG
IN
10:50
14-11-2025

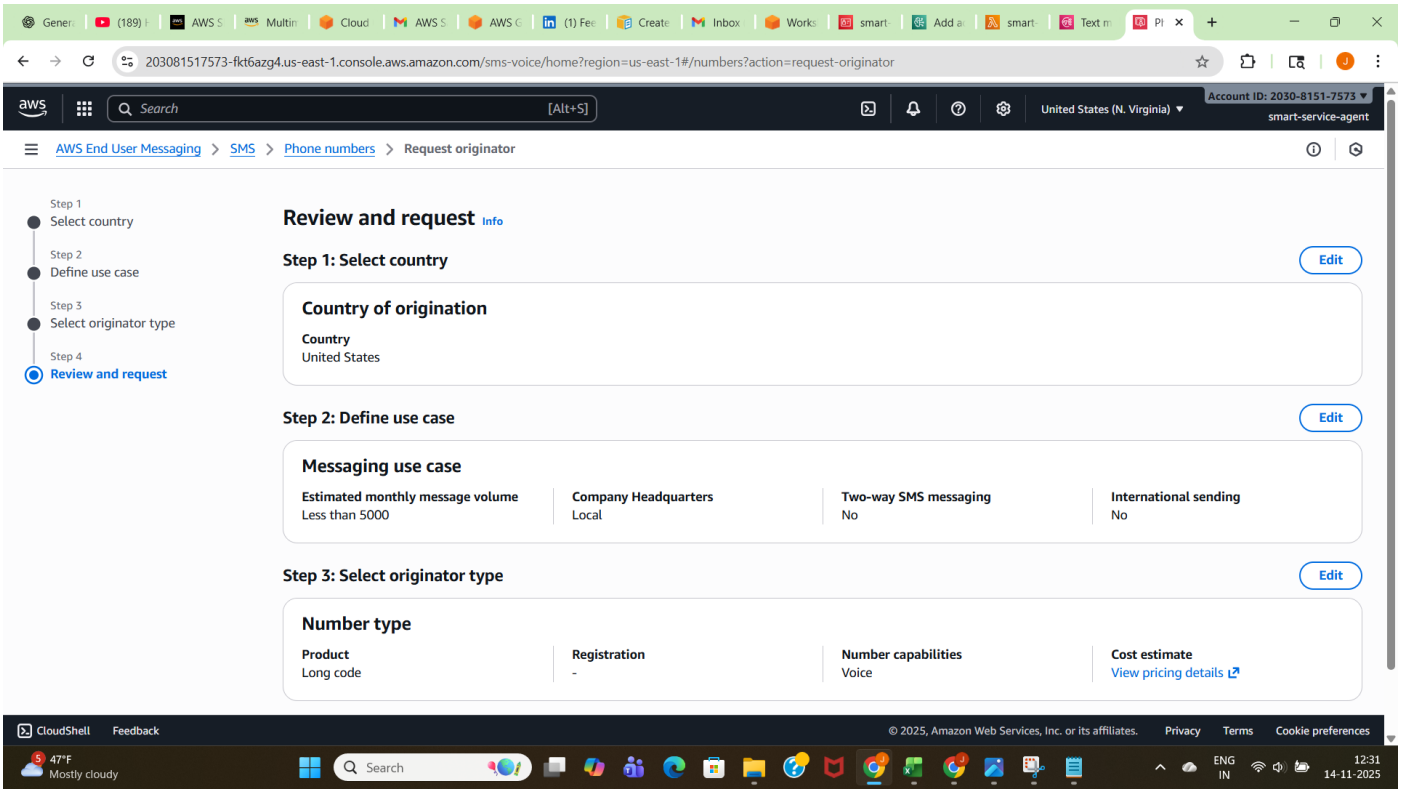


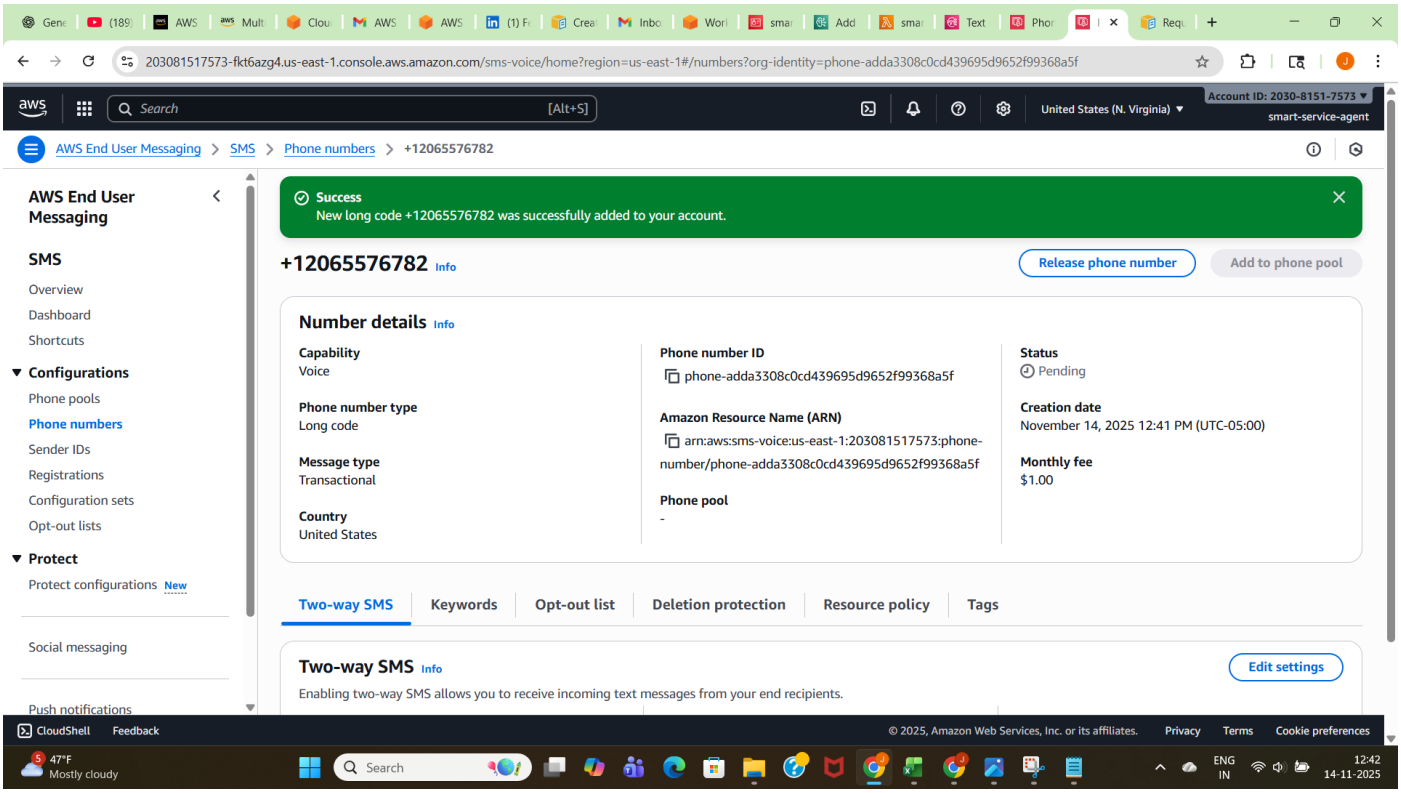
This screenshot shows the Lambda function **smart-service-lambda**, which processes Bedrock Agent actions to send emails or SMS based on the user request. The function uses boto3 to interact with Amazon SES and SNS, enabling automated message delivery as part of the agent workflow.

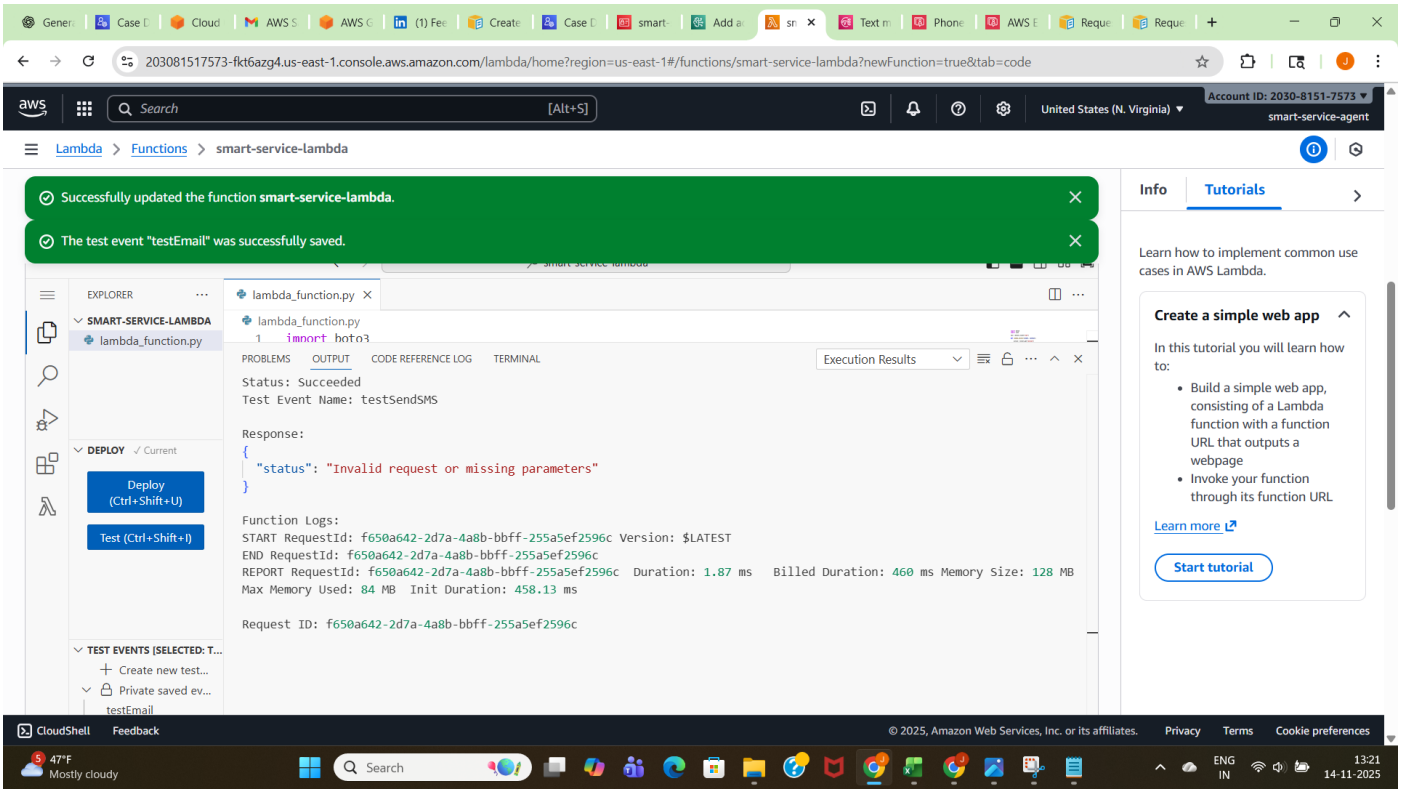


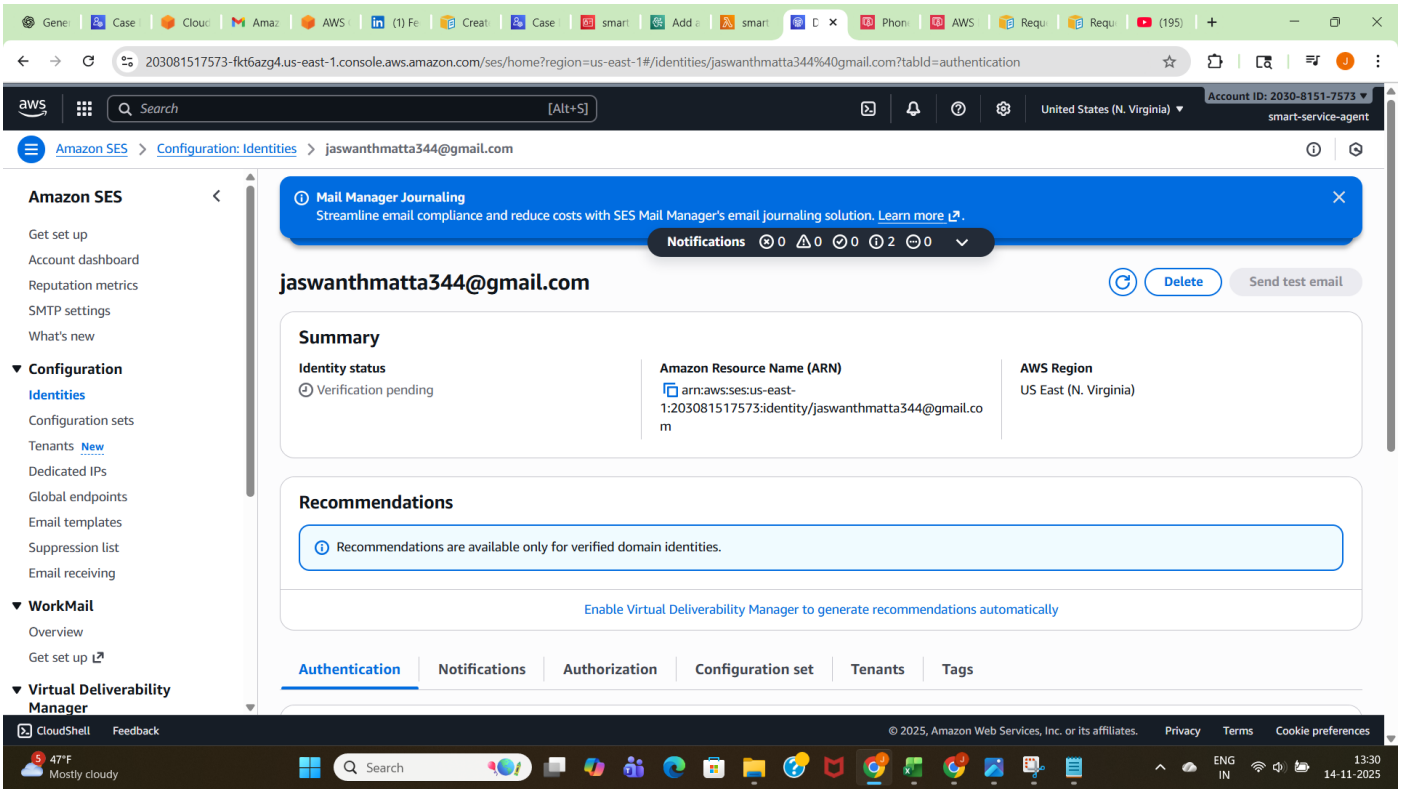

The test event "testSendSMS" was successfully saved.
Code source Info
Open in Visual Studio Code
Upload from
Status: Succeeded
Test Event Name: testSendSMS
Response:
{
"status": "SMS sent"
}
Function Logs:
START RequestId: 5d28fabb-34eb-44c4-af75-ee1fcd51d585 Version: $LATEST
END RequestId: 5d28fabb-34eb-44c4-af75-ee1fcd51d585
REPORT RequestId: 5d28fabb-34eb-44c4-af75-ee1fcd51d585 Duration: 226.48 ms Billed Duration: 227 ms Memory Size: 128 MB
Max Memory Used: 87 MB
Request ID: 5d28fabb-34eb-44c4-af75-ee1fcd51d585
Info
Tutorials
Learn how to implement common use cases in AWS Lambda.
Create a simple web app
In this tutorial you will learn how to:
Build a simple web app, consisting of a Lambda function with a function URL that outputs a webpage
Invoke your function through its function URL
Learn more
Start tutorial

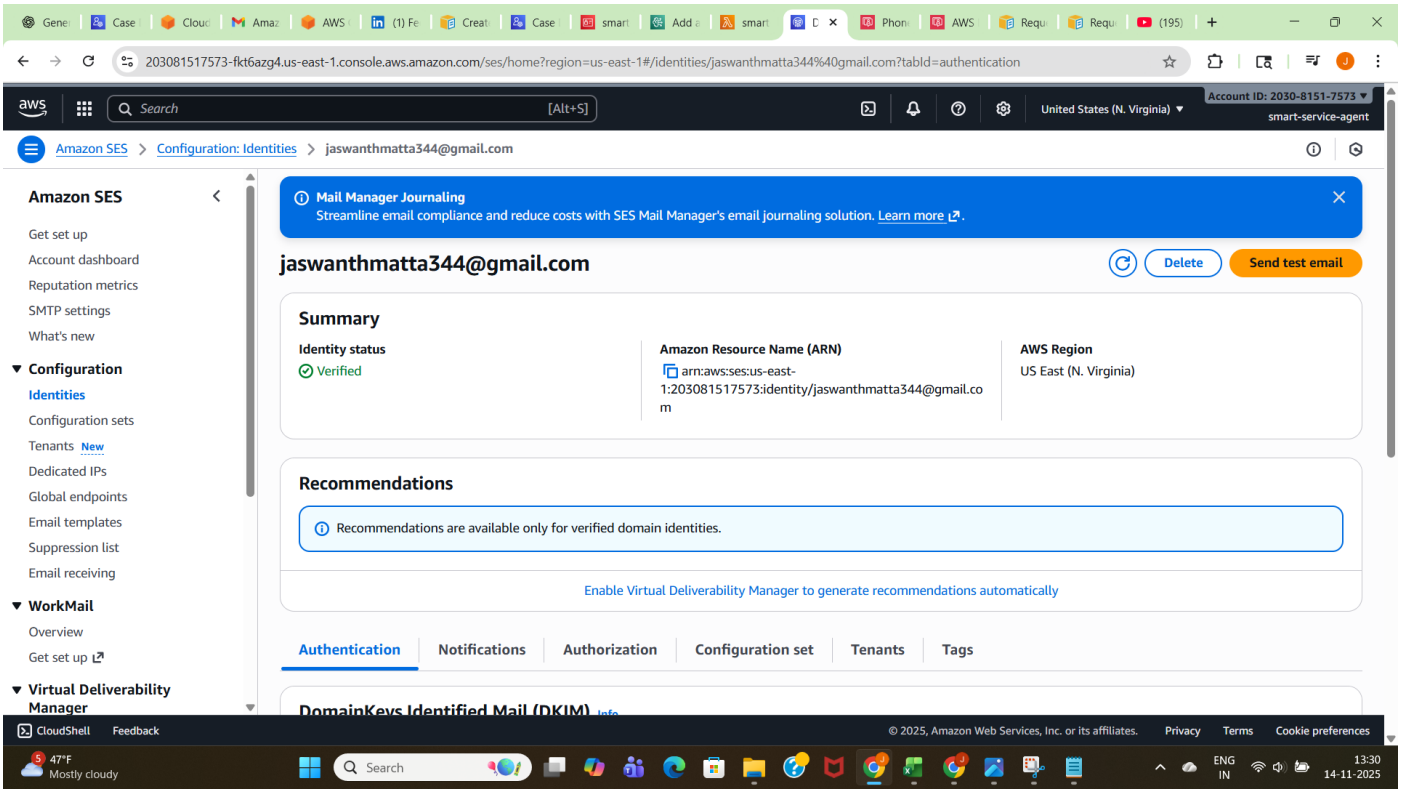

Policies have been successfully attached to role.
Identity and Access Management (IAM)
Search IAM
Dashboard
Access management
User groups
Users
Roles
Policies
Identity providers
Account settings
Root access management
Access reports
Access Analyzer
Resource analysis New
Unused access
Analyzer settings
Credential report
Policy name
Type
Attached entities
AmazonBedrockFullAccess AWS managed 3
AmazonSESFullAccess AWS managed 1
AmazonSNSFullAccess AWS managed 2
AWSLambdaBasicExecutionRole-7c3568fa-... Customer managed 1
Permissions boundary (not set)
Generate policy based on CloudTrail events
You can generate a new policy based on the access activity for this role, then customize, create, and attach it to this role. AWS uses your CloudTrail events to identify the services and actions used and generate a policy. Learn more
Generate policy
No requests to generate a policy in the past 7 days.

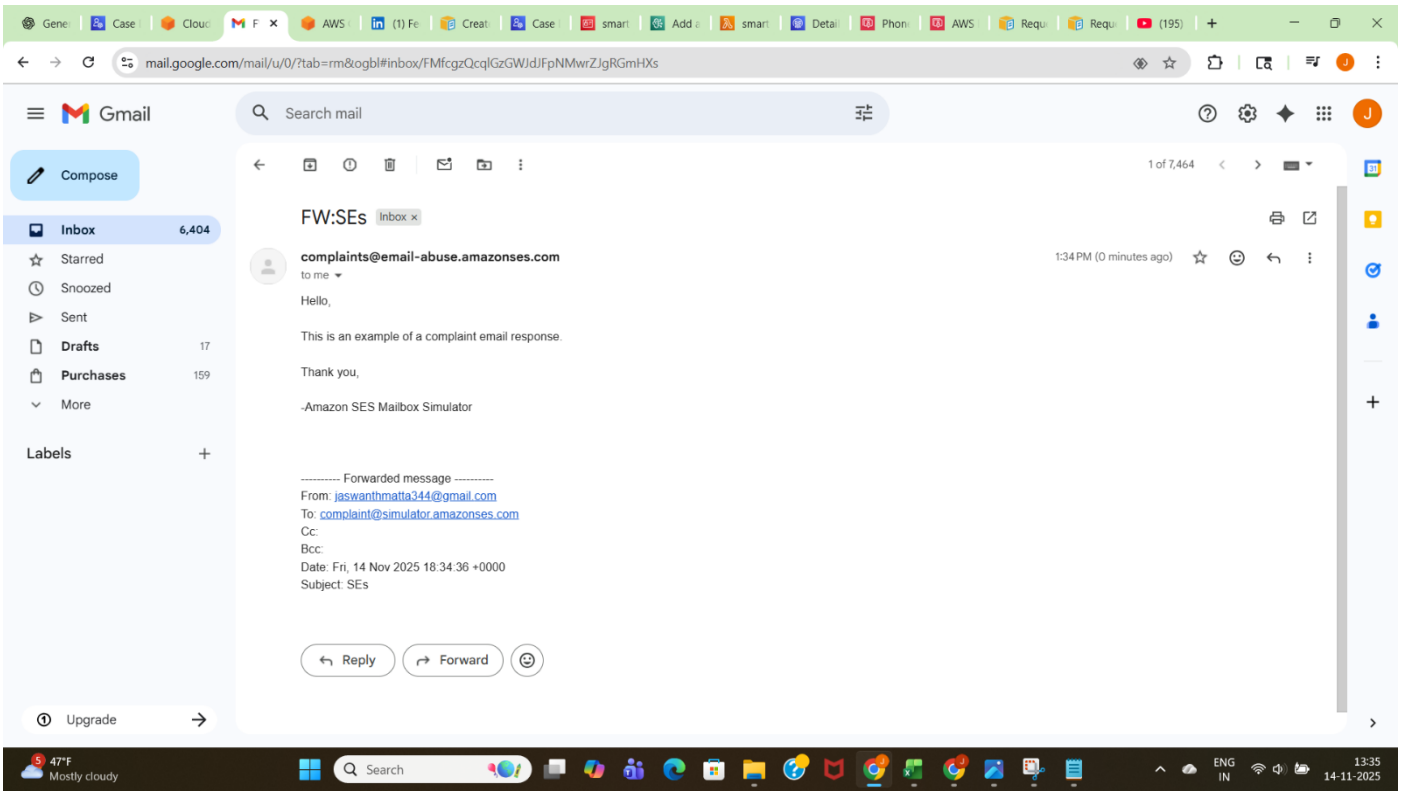

203081517573-fkt6azg4.us-east-1.console.aws.amazon.com/sms-voice/home?region=us-east-1#/numbers?action=request-originator
Search [Alt+S]
United States (N. Virginia)
Account ID: 2030-8151-7573
smart-service-agent
AWS End User Messaging > SMS > Phone numbers > Request originator
Step 1
Select country
Step 2
Define use case
Step 3
Select originator type
Step 4
Review and request
Review and request Info
Step 1: Select country
Edit
Country of origination
Country
United States
Step 2: Define use case
Edit
Messaging use case
Estimated monthly message volume
Less than 5000
Company Headquarters
Local
Two-way SMS messaging
No
International sending
No
Step 3: Select originator type
Edit
Number type
Product
Long code
Registration
-
Number capabilities
Voice
Cost estimate
View pricing details
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
47°F
Mostly cloudy
Search
ENG
IN
12:31
14-11-2025

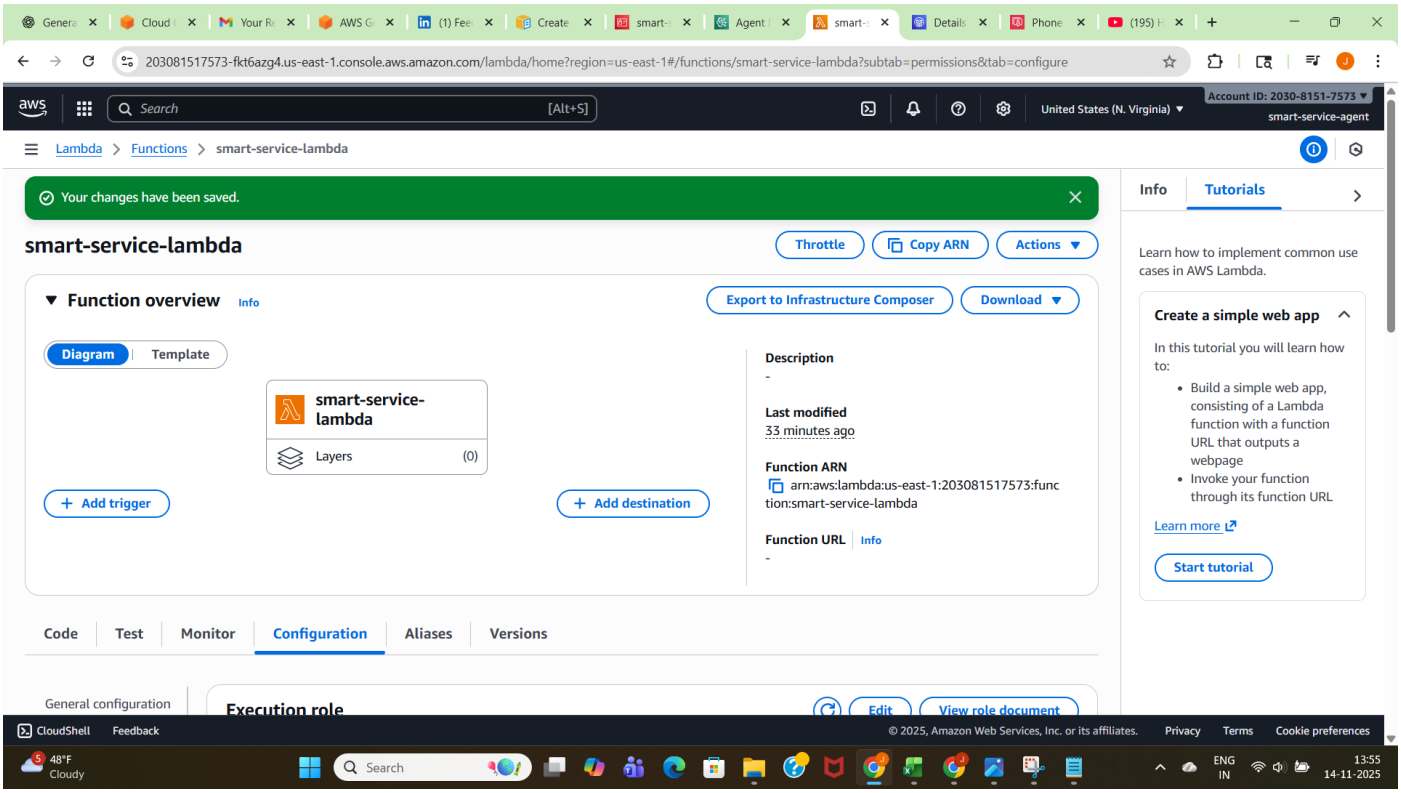

203081517573-fkt6azg4.us-east-1.console.aws.amazon.com/sms-voice/home?region=us-east-1#/numbers?org-identity=phone-adda3308c0cd439695d9652f99368a5f
Search [Alt+S]
United States (N. Virginia)
Account ID: 2030-8151-7573
smart-service-agent
AWS End User Messaging > SMS > Phone numbers > +12065576782
AWS End User Messaging
SMS
Overview
Dashboard
Shortcuts
Configurations
Phone pools
Phone numbers
Sender IDs
Registrations
Configuration sets
Opt-out lists
Protect
Protect configurations New
Social messaging
Push notifications
Success
New long code +12065576782 was successfully added to your account.
+12065576782 Info
Release phone number
Add to phone pool
Number details Info
Capability
Voice
Phone number type
Long code
Message type
Transactional
Country
United States
Phone number ID
phone-adda3308c0cd439695d9652f99368a5f
Amazon Resource Name (ARN)
arn:aws:sms-voice:us-east-1:203081517573:phone-number/phone-adda3308c0cd439695d9652f99368a5f
Phone pool
-
Status
Pending
Creation date
November 14, 2025 12:41 PM (UTC-05:00)
Monthly fee
$1.00
Two-way SMS
Keywords
Opt-out list
Deletion protection
Resource policy
Tags
Two-way SMS Info
Edit settings
Enabling two-way SMS allows you to receive incoming text messages from your end recipients.
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
47°F
Mostly cloudy
Search
ENG
IN
12:42
14-11-2025

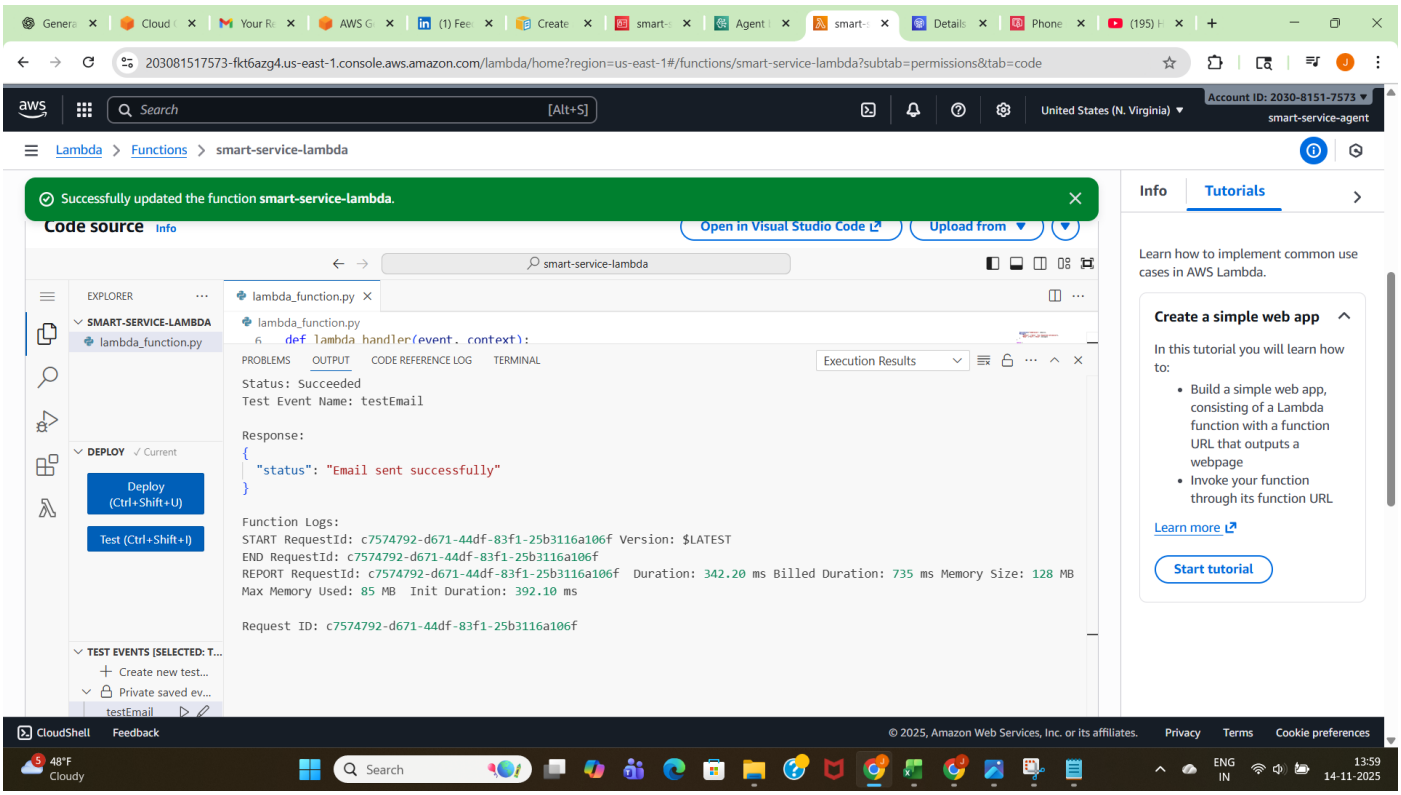

Successfully updated the function smart-service-lambda.
The test event "testEmail" was successfully saved.
lambda_function.py
Status: Succeeded
Test Event Name: testSendSMS
Response:
{
"status": "Invalid request or missing parameters"
}
Function Logs:
START RequestId: f650a642-2d7a-4a8b-bbff-255a5ef2596c Version: $LATEST
END RequestId: f650a642-2d7a-4a8b-bbff-255a5ef2596c
REPORT RequestId: f650a642-2d7a-4a8b-bbff-255a5ef2596c Duration: 1.87 ms Billed Duration: 460 ms Memory Size: 128 MB
Max Memory Used: 84 MB Init Duration: 458.13 ms
Request ID: f650a642-2d7a-4a8b-bbff-255a5ef2596c
Info
Tutorials
Learn how to implement common use cases in AWS Lambda.
Create a simple web app
In this tutorial you will learn how to:
Build a simple web app, consisting of a Lambda function with a function URL that outputs a webpage
Invoke your function through its function URL
Learn more
Start tutorial

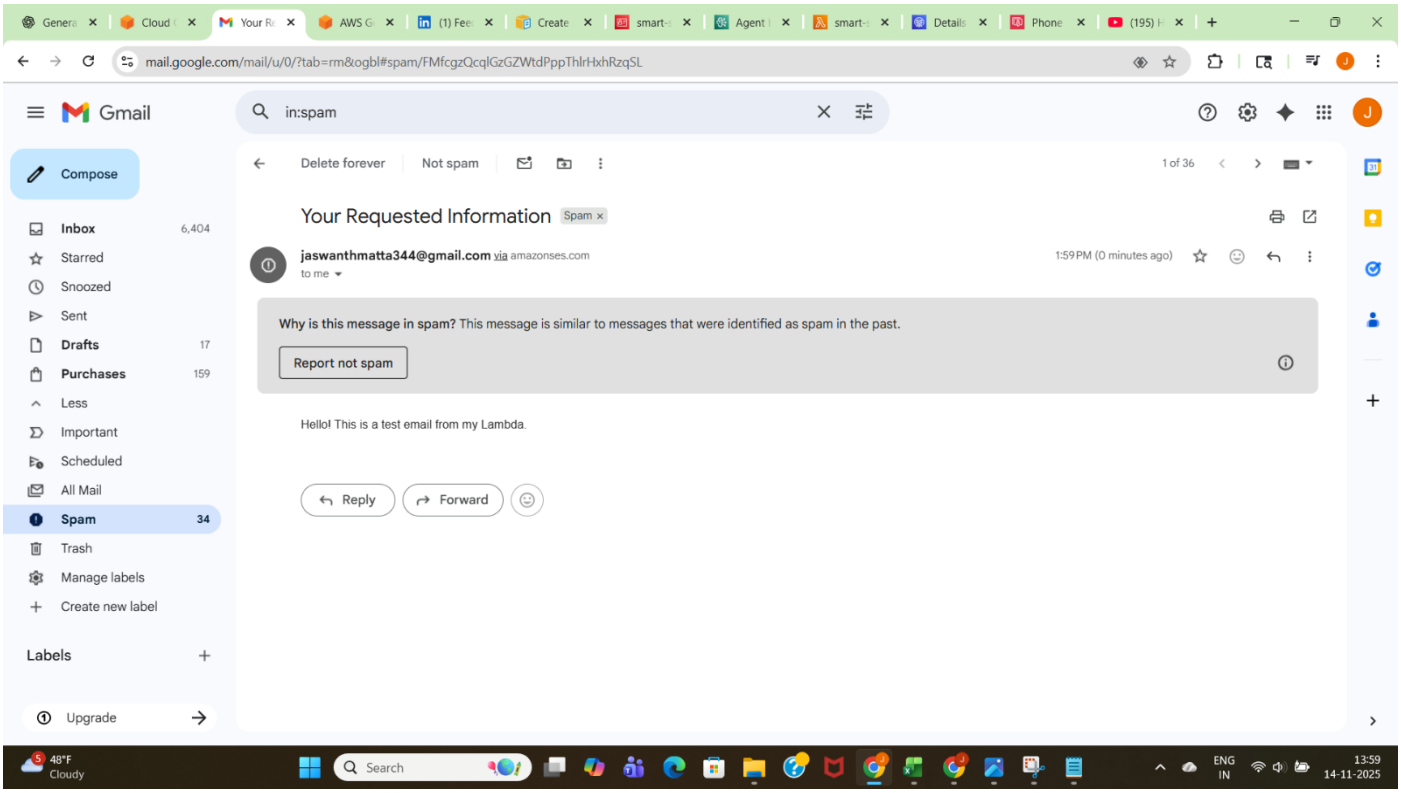

Amazon SES
Configuration: Identities
jaswanthmatta344@gmail.com
Mail Manager Journaling
Streamline email compliance and reduce costs with SES Mail Manager's email journaling solution. Learn more
jaswanthmatta344@gmail.com
Delete
Send test email
Summary
Identity status
Verification pending
Amazon Resource Name (ARN)
arn:aws:ses:us-east-1:203081517573:identity/jaswanthmatta344@gmail.com
AWS Region
US East (N. Virginia)
Recommendations
Recommendations are available only for verified domain identities.
Enable Virtual Deliverability Manager to generate recommendations automatically
Authentication
Notifications
Authorization
Configuration set
Tenants
Tags


203081517573-fkt6azg4.us-east-1.console.aws.amazon.com/ses/home?region=us-east-1#/identities/jaswanthmatta344%40gmail.com?tabId=authentication
Search [Alt+S]
United States (N. Virginia)
Account ID: 2030-8151-7573
smart-service-agent
Amazon SES > Configuration: Identities > jaswanthmatta344@gmail.com
Amazon SES
Get set up
Account dashboard
Reputation metrics
SMTP settings
What's new
Configuration
Identities
Configuration sets
Tenants New
Dedicated IPs
Global endpoints
Email templates
Suppression list
Email receiving
WorkMail
Overview
Get set up
Virtual Deliverability Manager
Mail Manager Journaling
Streamline email compliance and reduce costs with SES Mail Manager's email journaling solution. Learn more.
jaswanthmatta344@gmail.com
Delete
Send test email
Summary
Identity status
Verified
Amazon Resource Name (ARN)
arn:aws:ses:us-east-1:203081517573:identity/jaswanthmatta344@gmail.com
AWS Region
US East (N. Virginia)
Recommendations
Recommendations are available only for verified domain identities.
Enable Virtual Deliverability Manager to generate recommendations automatically
Authentication
Notifications
Authorization
Configuration set
Tenants
Tags
DomainKeys Identified Mail (DKIM)
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
47°F
Mostly cloudy
Search
ENG
IN
13:30
14-11-2025

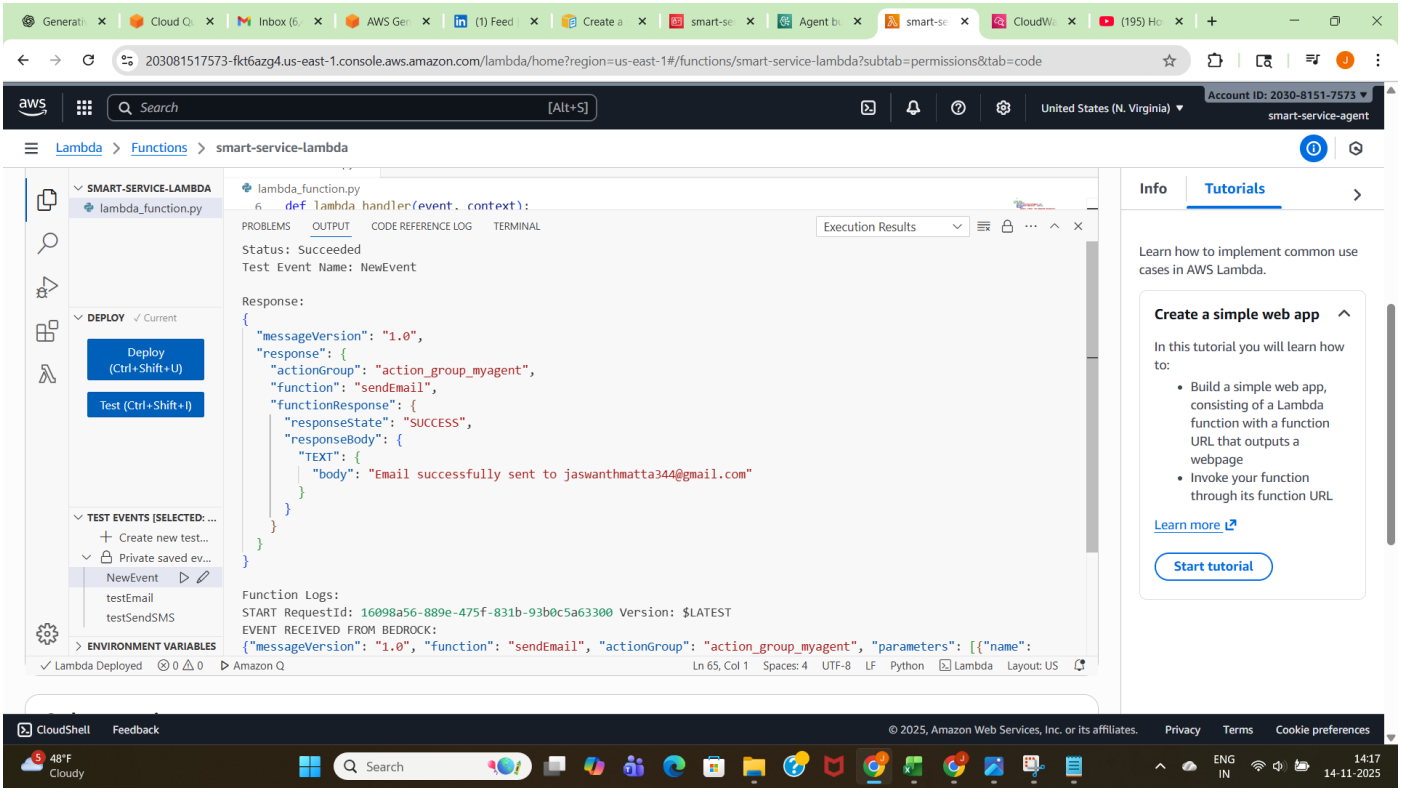

mail.google.com/mail/u/0/?tab=rm&ogbl#inbox/FMfcgzQcqlGzGWJdJFpNMwrZJgRGmHXs
Gmail
Search mail
Compose
Inbox 6,404
Starred
Snoozed
Sent
Drafts 17
Purchases 159
More
Labels
Upgrade
1 of 7,464
FW:SEs
Inbox
complaints@email-abuse.amazonses.com
to me
1:34 PM (0 minutes ago)
Hello,
This is an example of a complaint email response.
Thank you,
-Amazon SES Mailbox Simulator
---------- Forwarded message ---------
From: jaswanthmatta344@gmail.com
To: complaint@simulator.amazonses.com
Cc:
Bcc:
Date: Fri, 14 Nov 2025 18:34:36 +0000
Subject: SEs
Reply
Forward
47°F
Mostly cloudy
Search
ENG
IN
13:35
14-11-2025

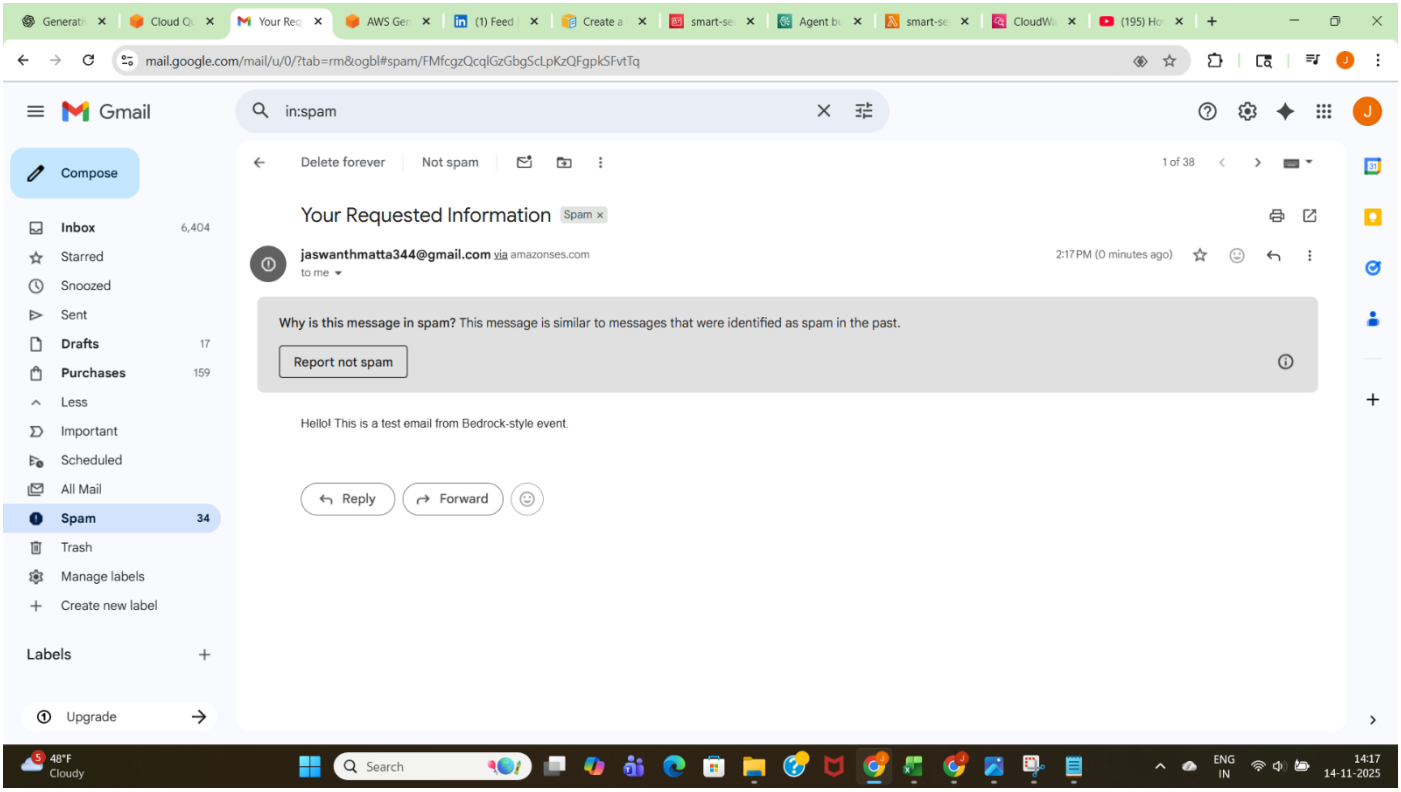

203081517573-fkt6azg4.us-east-1.console.aws.amazon.com/lambda/home?region=us-east-1#/functions/smart-service-lambda?subtab=permissions&tab=configure
Search [Alt+S]
United States (N. Virginia)
Account ID: 2030-8151-7573
smart-service-agent
Lambda > Functions > smart-service-lambda
Your changes have been saved.
smart-service-lambda
Throttle
Copy ARN
Actions
Function overview Info
Export to Infrastructure Composer
Download
Diagram
Template
smart-service-lambda
Layers (0)
+ Add trigger
+ Add destination
Description
-
Last modified
33 minutes ago
Function ARN
arn:aws:lambda:us-east-1:203081517573:function:smart-service-lambda
Function URL Info
-
Code
Test
Monitor
Configuration
Aliases
Versions
General configuration
Execution role
Edit
View role document
Info
Tutorials
Learn how to implement common use cases in AWS Lambda.
Create a simple web app
In this tutorial you will learn how to:
Build a simple web app, consisting of a Lambda function with a function URL that outputs a webpage
Invoke your function through its function URL
Learn more
Start tutorial
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
48°F Cloudy
Search
ENG IN
13:55
14-11-2025

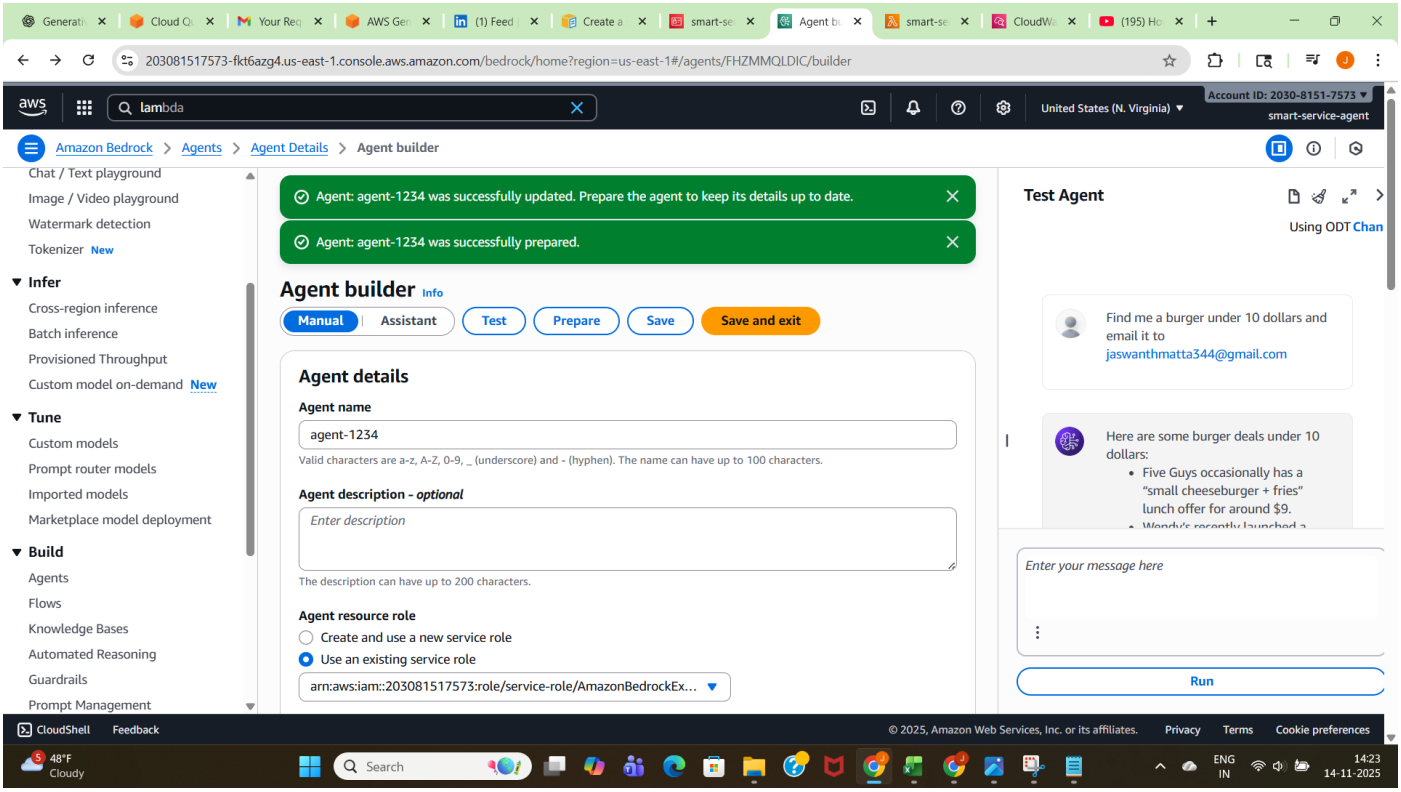

203081517573-fkt6azg4.us-east-1.console.aws.amazon.com/lambda/home?region=us-east-1#/functions/smart-service-lambda?subtab=permissions&tab=code
Search [Alt+S]
United States (N. Virginia)
Account ID: 2030-8151-7573
smart-service-agent
Lambda > Functions > smart-service-lambda
Successfully updated the function smart-service-lambda.
Code source Info
Open in Visual Studio Code
Upload from
smart-service-lambda
EXPLORER
SMART-SERVICE-LAMBDA
lambda_function.py
DEPLOY
Deploy (Ctrl+Shift+U)
Test (Ctrl+Shift+I)
TEST EVENTS (SELECTED: T...
Create new test...
Private saved ev...
testEmail
PROBLEMS
OUTPUT
CODE REFERENCE LOG
TERMINAL
Execution Results
Status: Succeeded
Test Event Name: testEmail
Response:
{
"status": "Email sent successfully"
}
Function Logs:
START RequestId: c7574792-d671-44df-83f1-25b3116a106f Version: $LATEST
END RequestId: c7574792-d671-44df-83f1-25b3116a106f
REPORT RequestId: c7574792-d671-44df-83f1-25b3116a106f Duration: 342.20 ms Billed Duration: 735 ms Memory Size: 128 MB
Max Memory Used: 85 MB Init Duration: 392.10 ms
Request ID: c7574792-d671-44df-83f1-25b3116a106f
Info
Tutorials
Learn how to implement common use cases in AWS Lambda.
Create a simple web app
In this tutorial you will learn how to:
Build a simple web app, consisting of a Lambda function with a function URL that outputs a webpage
Invoke your function through its function URL
Learn more
Start tutorial
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
48°F Cloudy
Search
ENG IN
13:59
14-11-2025

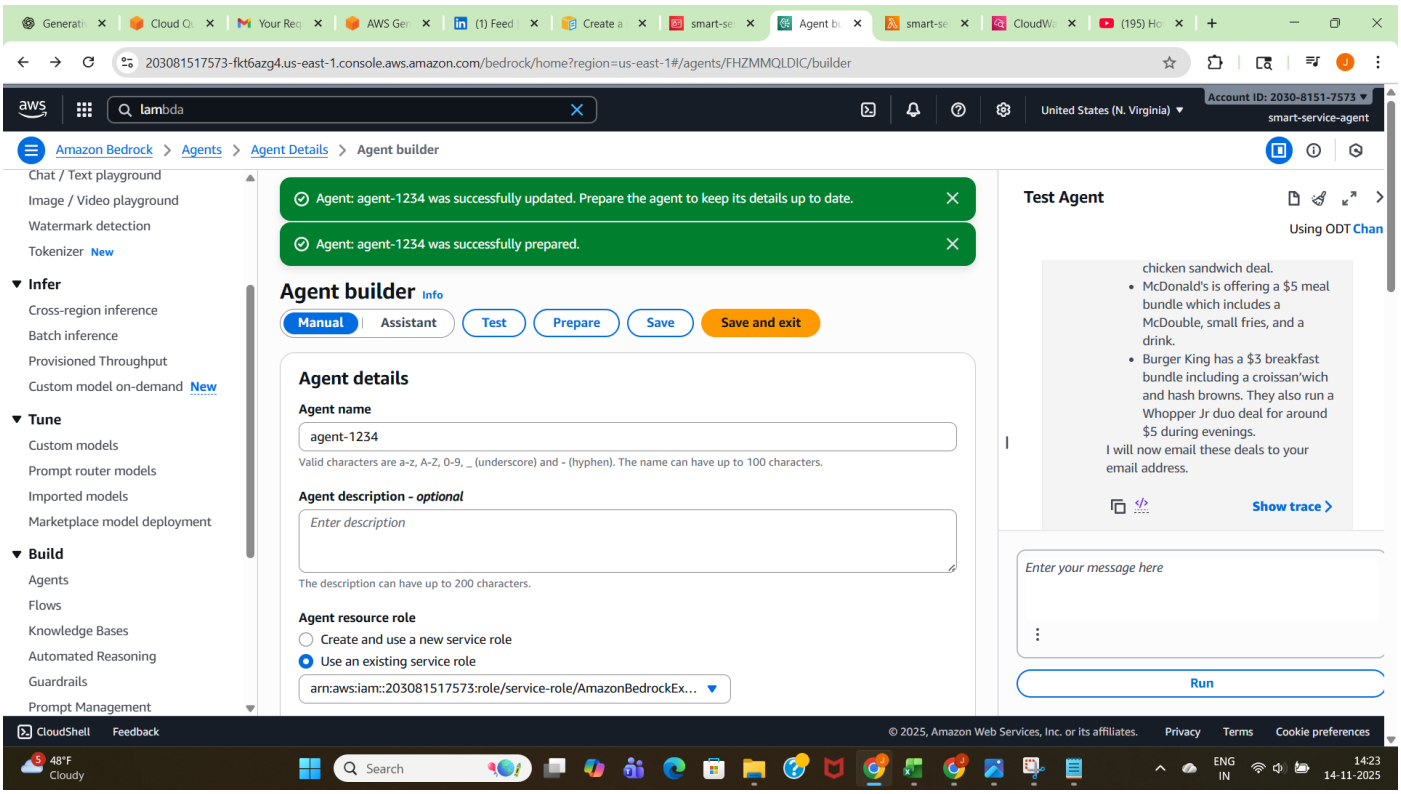


This screenshot shows the successful execution of the Lambda function triggered by the Bedrock Agent's **sendEmail** action, confirming that the email workflow is functioning correctly. The response output and logs indicate that the message was delivered to the specified email address without errors.

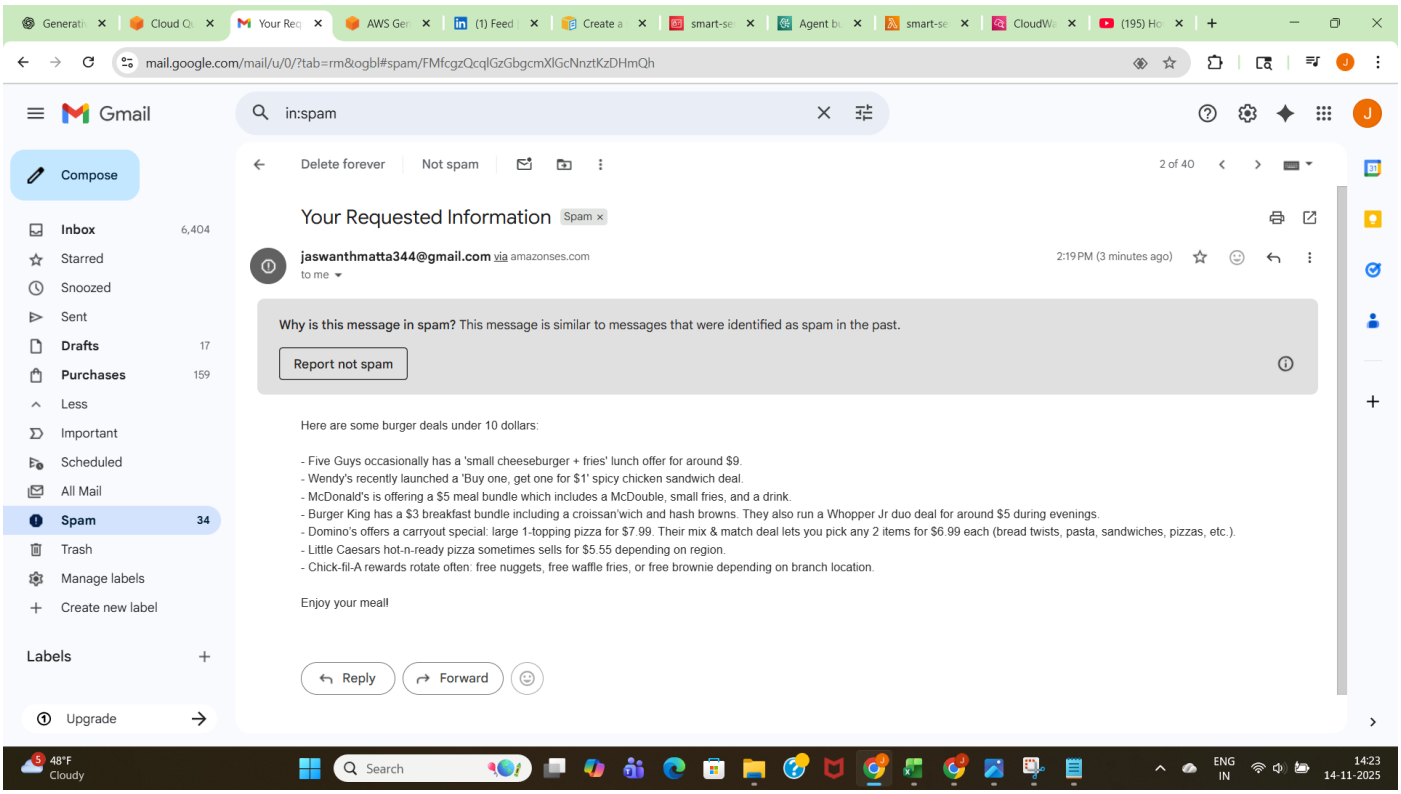


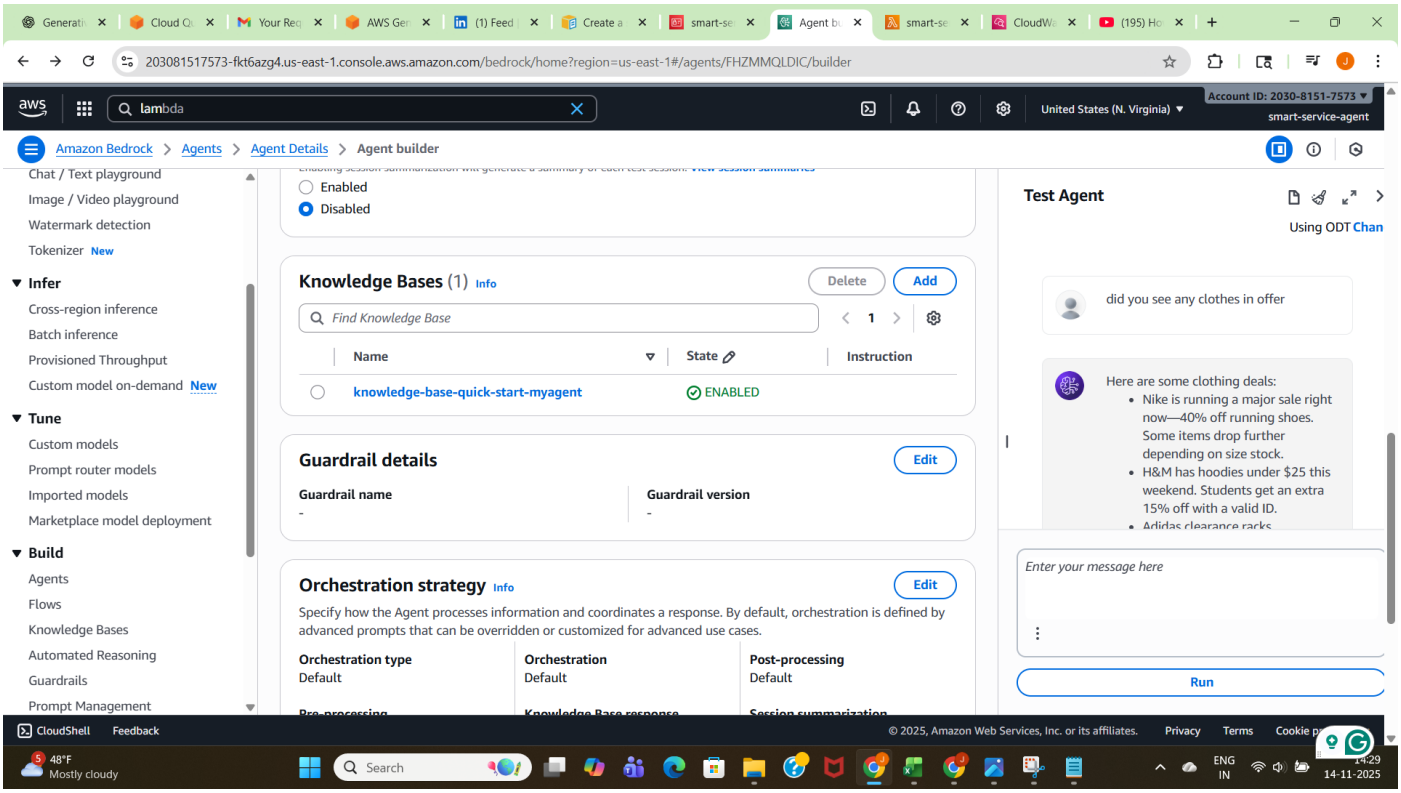

mail.google.com/mail/u/0/?tab=rm&ogbl#spam/FMfcgzQcqlGzGbgScLpKzQFgpkSFvtTq
Gmail
in:spam
Compose
Inbox 6,404
Starred
Snoozed
Sent
Drafts 17
Purchases 159
Less
Important
Scheduled
All Mail
Spam 34
Trash
Manage labels
Create new label
Labels
Upgrade
Delete forever
Not spam
1 of 38
Your Requested Information
Spam ×
jaswanthmatta344@gmail.com via amazonses.com
to me
2:17 PM (0 minutes ago)
Why is this message in spam? This message is similar to messages that were identified as spam in the past.
Report not spam
Hello! This is a test email from Bedrock-style event.
Reply
Forward
48°F
Cloudy
Search
ENG
IN
14:17
14-11-2025

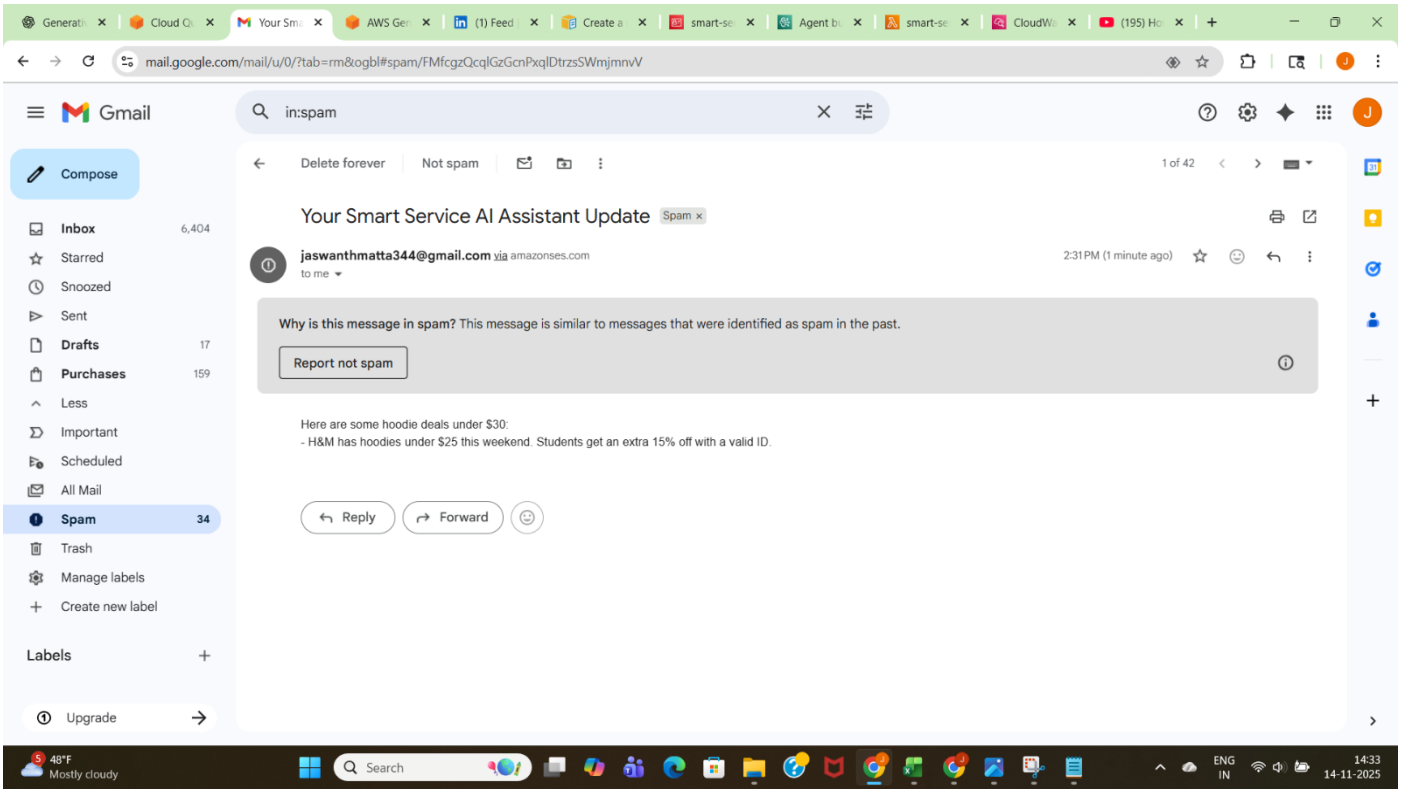

203081517573-fkt6azg4.us-east-1.console.aws.amazon.com/bedrock/home?region=us-east-1#/agents/FHZMMQLDIC/builder
aws
lambda
United States (N. Virginia)
Account ID: 2030-8151-7573
smart-service-agent
Amazon Bedrock > Agents > Agent Details > Agent builder
Chat / Text playground
Image / Video playground
Watermark detection
Tokenizer New
Infer
Cross-region inference
Batch inference
Provisioned Throughput
Custom model on-demand New
Tune
Custom models
Prompt router models
Imported models
Marketplace model deployment
Build
Agents
Flows
Knowledge Bases
Automated Reasoning
Guardrails
Prompt Management
Agent: agent-1234 was successfully updated. Prepare the agent to keep its details up to date.
Agent: agent-1234 was successfully prepared.
Agent builder Info
Manual
Assistant
Test
Prepare
Save
Save and exit
Agent details
Agent name
agent-1234
Valid characters are a-z, A-Z, 0-9, _ (underscore) and - (hyphen). The name can have up to 100 characters.
Agent description - optional
Enter description
The description can have up to 200 characters.
Agent resource role
Create and use a new service role
Use an existing service role
arn:aws:iam::203081517573:role/service-role/AmazonBedrockEx...
Test Agent
Using ODT Chan
Find me a burger under 10 dollars and email it to jaswanthmatta344@gmail.com
Here are some burger deals under 10 dollars:
Five Guys occasionally has a "small cheeseburger + fries" lunch offer for around $9.
Enter your message here
Run
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
48°F
Cloudy
Search
ENG
IN
14:23
14-11-2025

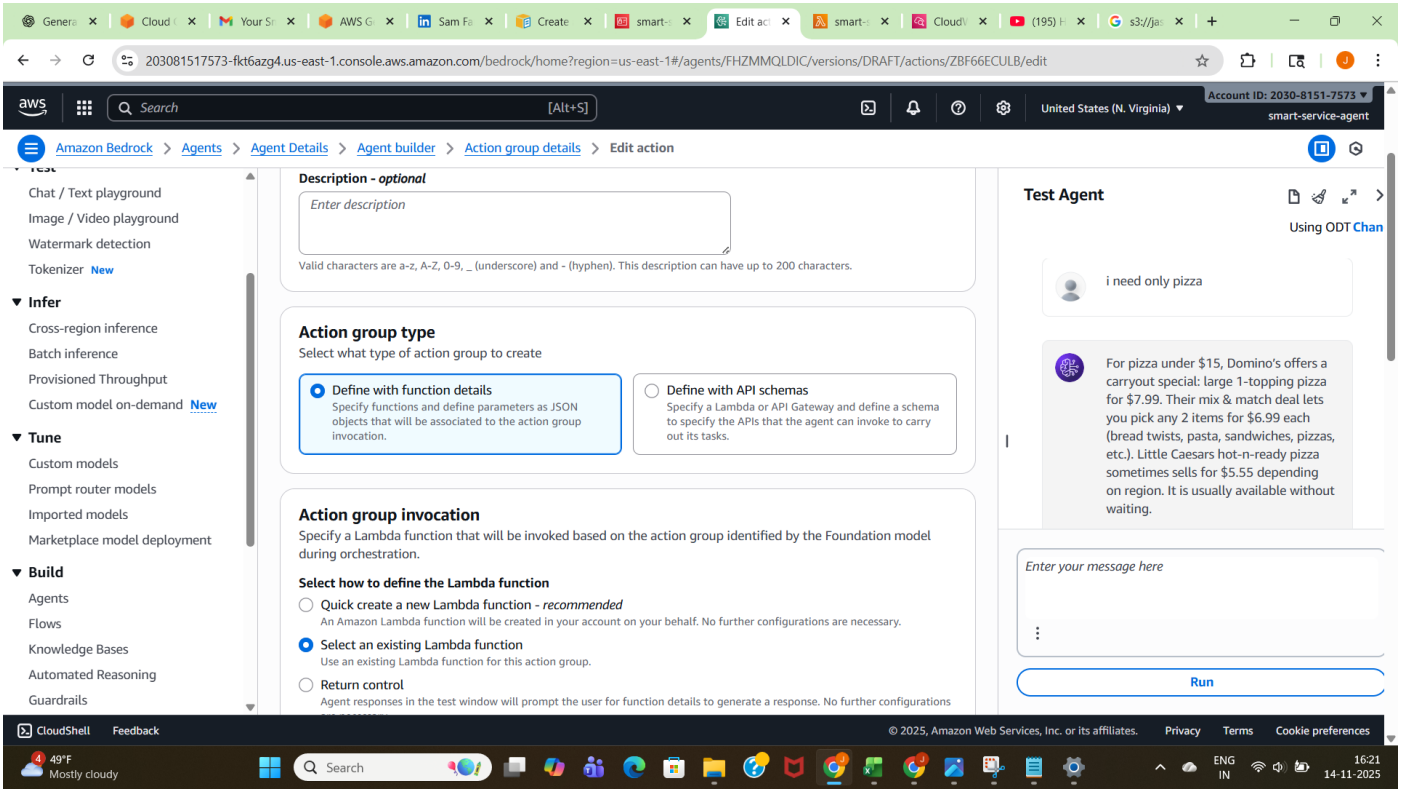


This screenshot shows the email successfully delivered by Amazon SES, containing AI-generated deal recommendations retrieved through the Bedrock Knowledge Base. Although Gmail placed it in the spam folder, the content confirms that the end-to-end workflow—from query to retrieval to email delivery—is working correctly




203081517573-fkt6azg4.us-east-1.console.aws.amazon.com/bedrock/home?region=us-east-1#/agents/FHZMMQLDIC/builder
lambda
United States (N. Virginia)
Account ID: 2030-8151-7573
smart-service-agent
Amazon Bedrock > Agents > Agent Details > Agent builder
Chat / Text playground
Image / Video playground
Watermark detection
Tokenizer New
Infer
Cross-region inference
Batch inference
Provisioned Throughput
Custom model on-demand New
Tune
Custom models
Prompt router models
Imported models
Marketplace model deployment
Build
Agents
Flows
Knowledge Bases
Automated Reasoning
Guardrails
Prompt Management
Enabled
Disabled
Knowledge Bases (1) Info
Delete
Add
Find Knowledge Base
Name
State
Instruction
knowledge-base-quick-start-myagent
ENABLED
Guardrail details
Edit
Guardrail name
-
Guardrail version
-
Orchestration strategy Info
Edit
Specify how the Agent processes information and coordinates a response. By default, orchestration is defined by advanced prompts that can be overridden or customized for advanced use cases.
Orchestration type
Default
Orchestration
Default
Post-processing
Default
Test Agent
Using ODT Chan
did you see any clothes in offer
Here are some clothing deals:
Nike is running a major sale right now—40% off running shoes. Some items drop further depending on size stock.
H&M has hoodies under $25 this weekend. Students get an extra 15% off with a valid ID.
Enter your message here
Run
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
48°F
Mostly cloudy
Search
ENG
IN
14-11-2025


mail.google.com/mail/u/0/?tab=rm&ogbl#spam/FMfcgzQcqlGzGcnPxqlDtrzsSWmjmnvV
Gmail
in:spam
Compose
Inbox 6,404
Starred
Snoozed
Sent
Drafts 17
Purchases 159
Less
Important
Scheduled
All Mail
Spam 34
Trash
Manage labels
Create new label
Labels
Upgrade
Delete forever
Not spam
1 of 42
Your Smart Service AI Assistant Update
Spam
jaswanthmatta344@gmail.com via amazonses.com
to me
2:31 PM (1 minute ago)
Why is this message in spam? This message is similar to messages that were identified as spam in the past.
Report not spam
Here are some hoodie deals under $30:
- H&M has hoodies under $25 this weekend. Students get an extra 15% off with a valid ID.
Reply
Forward
48°F
Mostly cloudy
Search
ENG
IN
14:33
14-11-2025



8.What I Learned

Through this project, I gained hands-on experience with:

Cloud Engineering

- Designing fully serverless architectures
- Securing service-to-service communication with IAM
- Leveraging managed services for scale and cost efficiency

Generative AI & RAG

- Building Bedrock Agents with custom tools
- Designing Knowledge Bases for semantic retrieval

- Using embeddings and vector search for unstructured data

Serverless Patterns

- Integrating Lambda with AI workflows
- Event-driven execution through agent action groups
- Using SES for automated notifications

This project significantly strengthened my skills in building **real-world AI systems** on AWS.

9.Conclusion

- The Smart Service AI Assistant demonstrates how **Generative AI**, **RAG**, and **AWS serverless services** can be combined to deliver real, actionable intelligence.It retrieves unstructured content, synthesizes it into natural responses, and performs real-world actions all without managing any infrastructure.
- This architecture can be extended to enterprise customer support, e-commerce applications, marketing automation, hospitality, or any domain requiring intelligent and automated interactions.