Telecom Resilience Platform: Serverless Billing, Fraud Detection & Disaster Recovery.

MUKESH GOND (B037) | JUBI RANKA (B039) |
TANVI SHINDE (B042) | ISHA LAD (B045) | SHRIYA NAIR (B056)

First-Year M.Sc. Data Science Students, Nilkamal School of Mathematics, Applied Statistics & Analytics (NSOMASA), SVKM's NMIMS (Deemed-to-be) University, Mumbai, Maharashtra, India

MENTOR :

Prof. Pratik Desai

Nilkamal School of Mathematics, Applied Statistics & Analytics (NSOMASA), SVKM's NMIMS (Deemed-to-be) University, Mumbai, Maharashtra, India

TABLE OF CONTENTS

Executive Summary

The telecom industry requires a scalable, highly available, and secure platform to monitor real-time network usage, automate customer billing, prevent fraud, and provide a self-service portal for customers. This document outlines the AWS-based architecture and detailed implementation plan for such a system, ensuring operational efficiency and cost optimization.

Business Use Case

Challenges Faced by Telecom Companies

1. **Real-Time Call & Data Usage Monitoring** – Telecom companies need to track millions of call logs, data usage, and network health metrics in real-time to ensure seamless operations and customer satisfaction.

2. **Automated Customer Billing** – Manual billing processes can lead to errors and delays; thus, a highly available automated billing system ensures customers are charged correctly and on time.

3. **Fraud Detection & Prevention** – Suspicious activity such as SIM cloning, spam calls, and fraudulent transactions must be detected and mitigated in real-time to protect users and revenue.

4. **Customer Self-Service Portal** – A user-friendly web portal enables customers to manage their accounts, recharge their plans, and resolve queries without needing customer support intervention.

5. **Disaster Recovery & Failover** – Ensuring service continuity during unexpected outages is crucial, requiring robust backup, failover, and disaster recovery mechanisms.

Solution Overview

Uniqueness of the Solution

- **Serverless & Scalable** – Optimized performance with AWS Lambda, Kinesis, and DynamoDB, eliminating the need for maintaining physical infrastructure.

- **Real-Time Processing** – Instant tracking and monitoring using AWS streaming services ensures up-to-date network insights.

- **AI-Driven Fraud Prevention** – Amazon SageMaker provides predictive analysis and fraud detection, reducing security risks.

- **Disaster-Resilient** – Multi-region replication and automated backups ensure business continuity even in system failures.

Proposed AWS Services

- **Data Ingestion & Processing:** Amazon Kinesis, AWS Lambda, AWS Step Functions.

- **Data Storage & Management:** Amazon DynamoDB, Amazon S3, Amazon Glacier.

- **Security & Compliance:** AWS WAF, AWS Shield, AWS IAM.

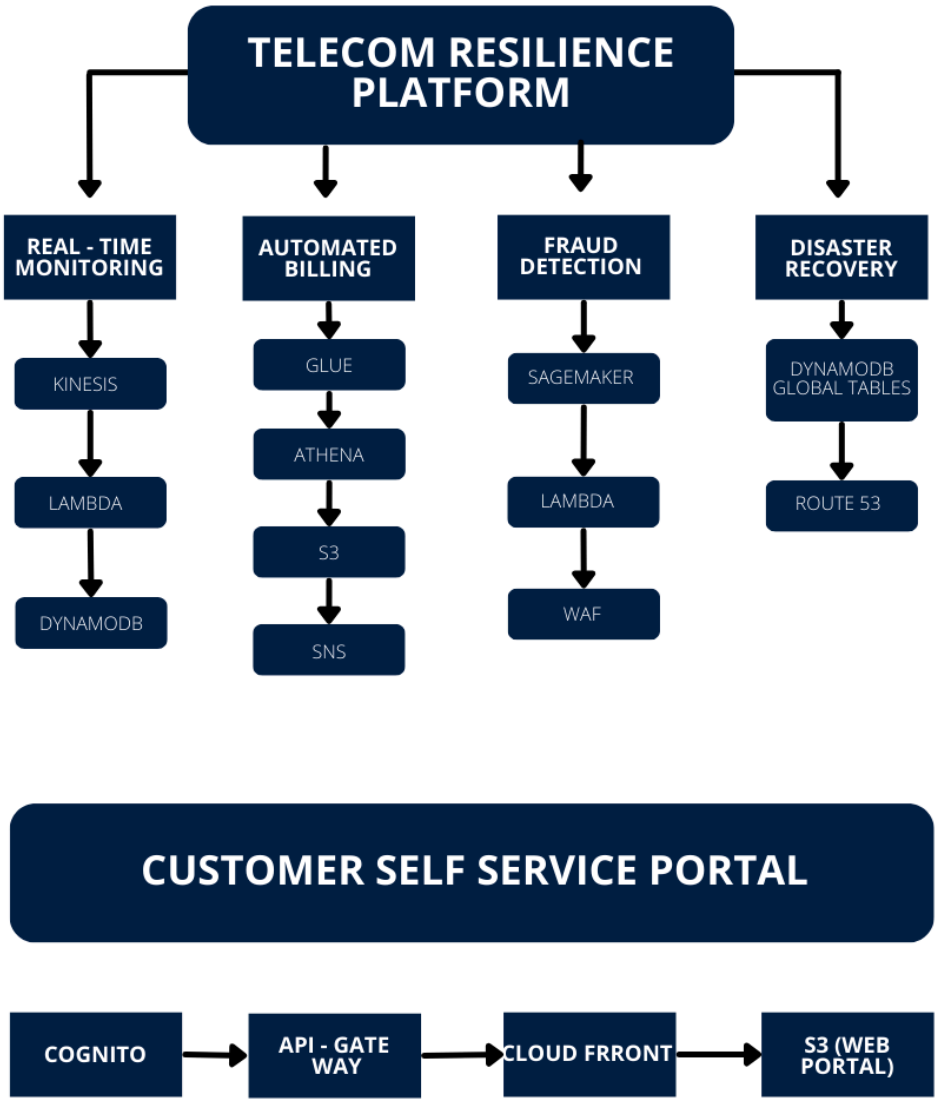
- **Billing & Fraud Detection:** AWS Glue, Amazon SageMaker, Amazon Athena.

- **Customer Self-Service:** Amazon API Gateway, Amazon Cognito, CloudFront.

- **Disaster Recovery:** AWS Backup, Multi-Region Replication, AWS Route 53.

AWS Architecture

Complete System Architecture Diagram



Implementation Plan

Step 1: Setting Up Real-Time Data Monitoring

1. **Create Amazon Kinesis Data Stream** – Configure Kinesis Data Streams to ingest real-time telecom data, such as call logs and network usage.

2. **Deploy AWS Lambda Functions** – Write serverless Lambda functions to process and filter the streaming data for further analysis.

3. **Store Data in Amazon DynamoDB** – Store processed data in a NoSQL DynamoDB table for efficient querying and retrieval.

4. **Enable Amazon CloudWatch Monitoring** – Set up CloudWatch to track system performance and alert on anomalies.

Step 2: Implementing Automated Billing System

1. **Extract Usage Data with AWS Glue** – Use AWS Glue to crawl and extract call and data usage details from DynamoDB and logs.

2. **Analyze Billing Data with Amazon Athena** – Query structured billing data using Amazon Athena to generate invoices dynamically.

3. **Store Generated Invoices in Amazon S3** – Store customer invoices securely in Amazon S3 for easy access and retrieval.

4. **Notify Customers via Amazon SNS** – Send automated SMS and email notifications to customers with billing summaries.

Step 3: Deploying Fraud Detection System

1. **Stream Call Metadata Using Kinesis Firehose** – Collect call patterns and metadata to detect suspicious behavior.

2. **Train AI Fraud Models with Amazon SageMaker** – Use historical fraud data to train machine learning models that identify fraudulent transactions.

3. **Trigger AWS Lambda for Instant Fraud Prevention** – Deploy Lambda functions to block fraudulent transactions in real-time.

4. **Implement AWS WAF for Threat Mitigation** – Use AWS WAF to prevent bot attacks, spam calls, and brute-force attempts.

Step 4: Developing Customer Self-Service Portal

1. **Host Portal on Amazon S3 & CloudFront** – Deploy a static web application on S3 with CloudFront for global content delivery.

2. **Use Amazon API Gateway for Service Requests** – Implement REST APIs to handle user account management, plan recharges, and support requests.

3. **Enable User Authentication with Amazon Cognito** – Secure customer accounts with multi-factor authentication and OAuth2 login.

4. **Store Customer Preferences in DynamoDB** – Store customer preferences and historical transactions in a scalable database.

Step 5: Configuring Disaster Recovery & Failover

1. **Enable AWS Backup for Regular Snapshots** – Schedule automated snapshots of databases and storage to prevent data loss.

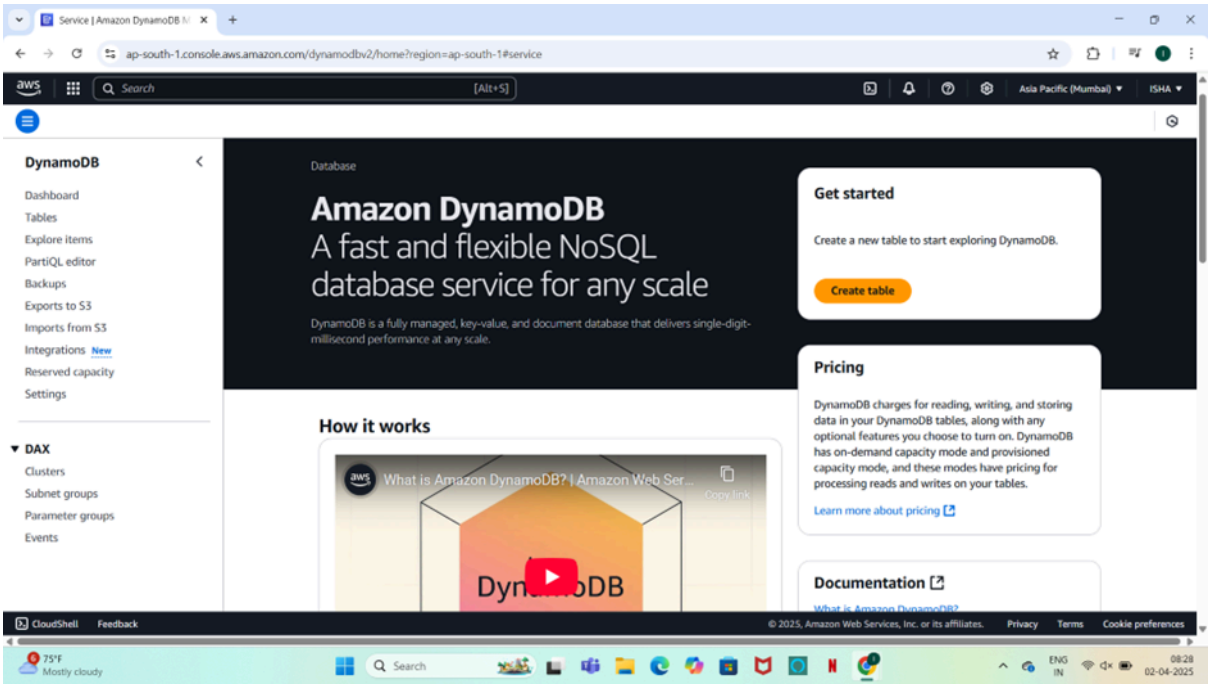
2. **Set Up Multi-Region Replication for DynamoDB & S3** – Ensure high availability by replicating data across different AWS regions.

3. **Use AWS Route 53 for Automatic Failover** – Configure Route 53 to switch traffic to backup servers in case of an outage.

4. **Conduct Disaster Recovery Drills Regularly** – Perform scheduled failover tests to validate the resilience of the system.

Step-by-Step Implementation

Step 1: Set Up DynamoDB Global Tables



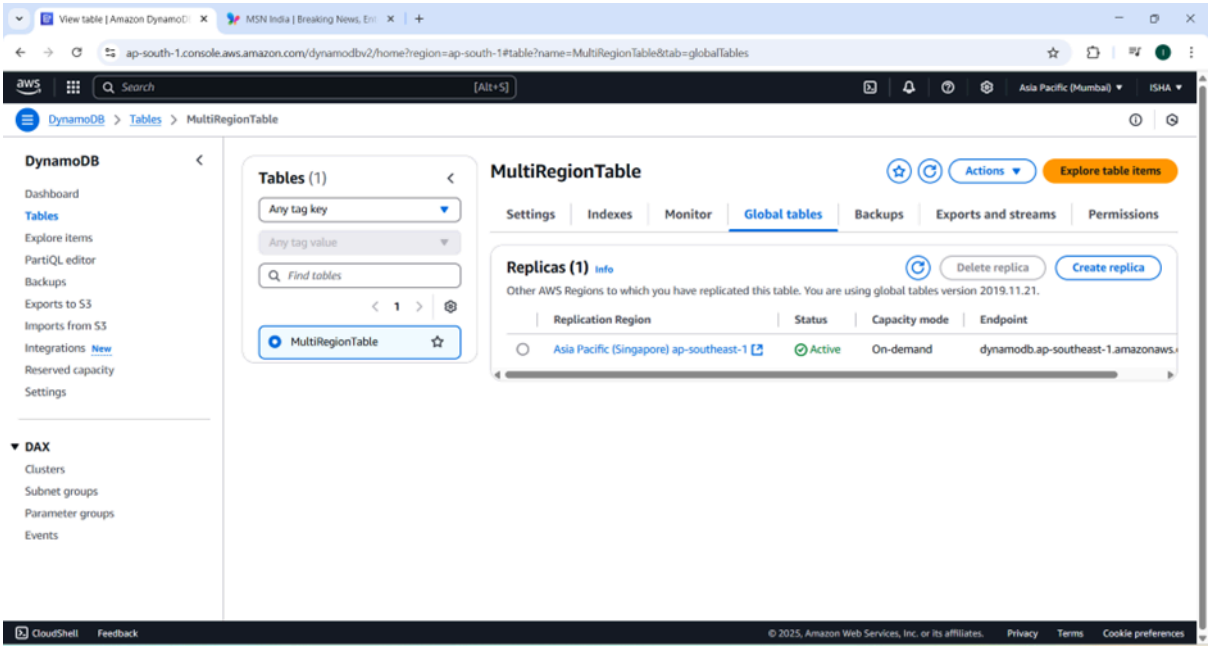
Primary Region (Mumbai/ap-south-1)

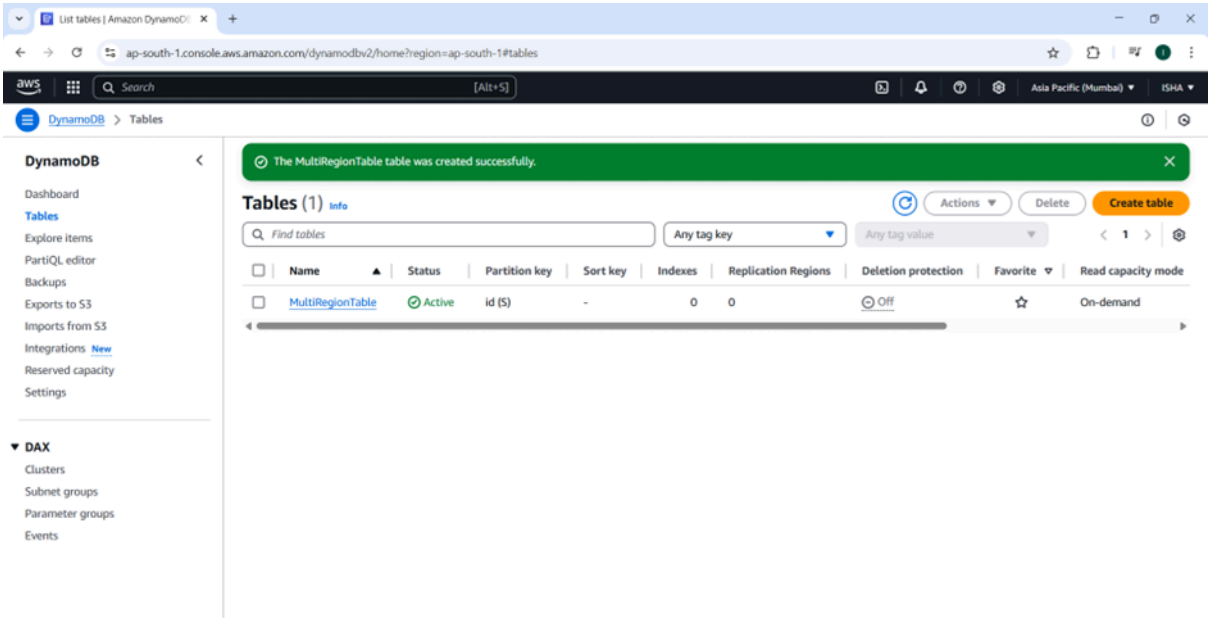
Since Mumbai is chosen as the primary region, all data will be first written here. The setup ensures high availability and quick access for users in this region.

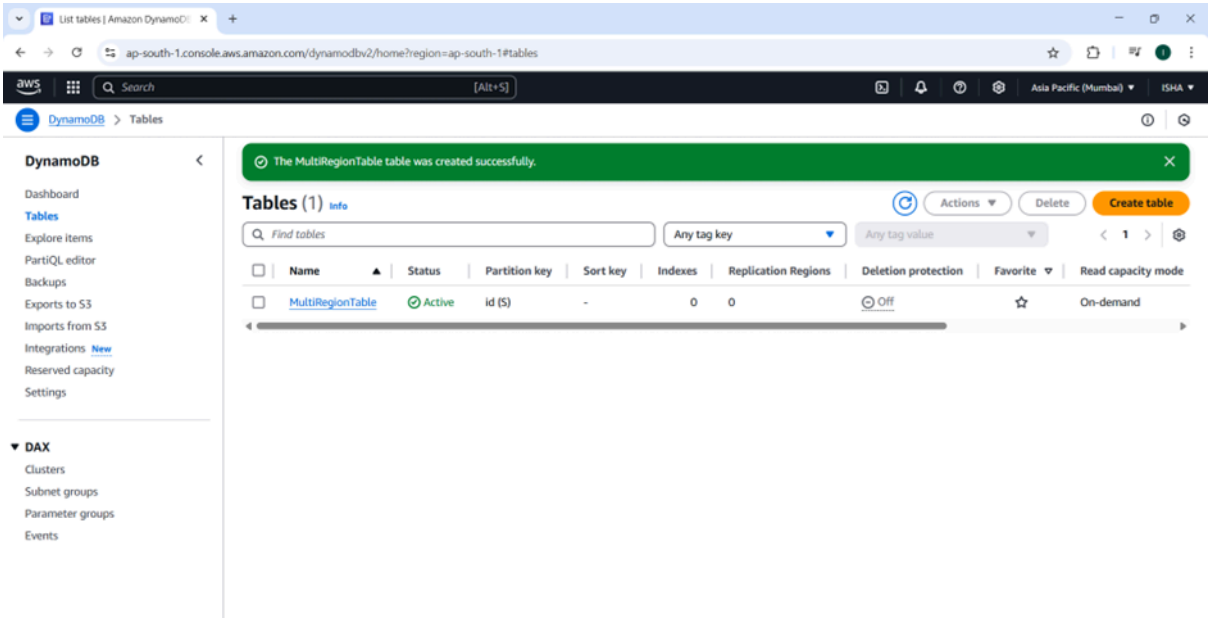
1. Create DynamoDB Table: The table stores application data that needs to be replicated across multiple regions.
    - Table name: MultiRegionTable
    - Partition key: id (String) -

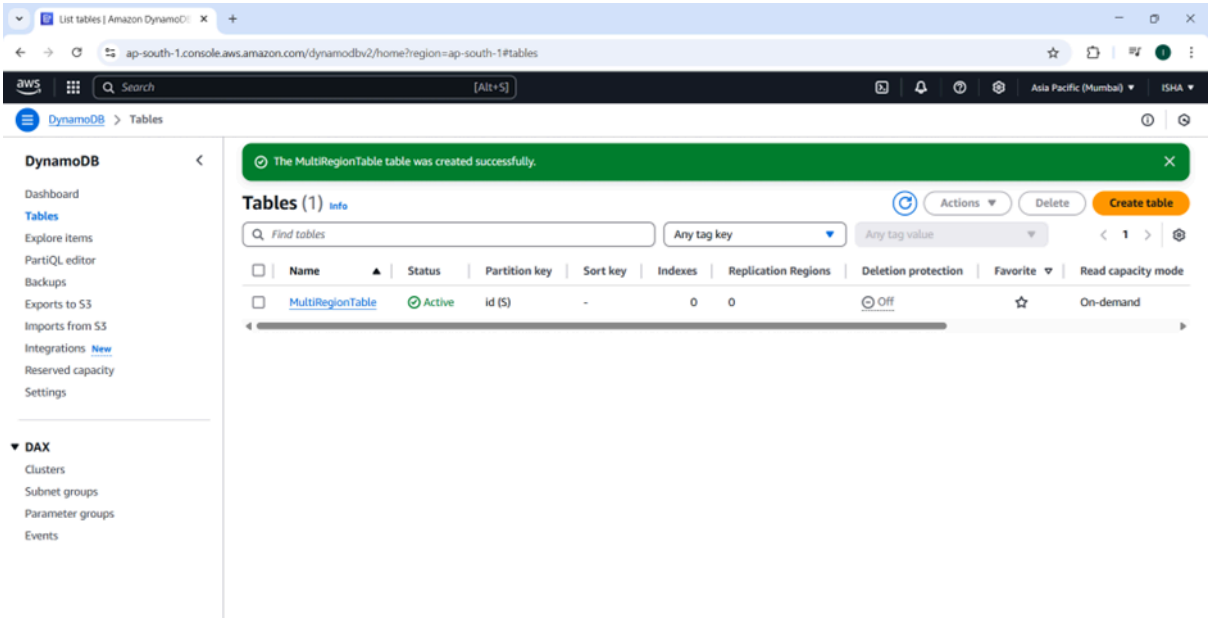
- The **partition key** uniquely identifies each item in the table.
- It ensures that data is distributed efficiently for fast lookups.
- Choosing `id` as a string makes it flexible for different types of data entries.

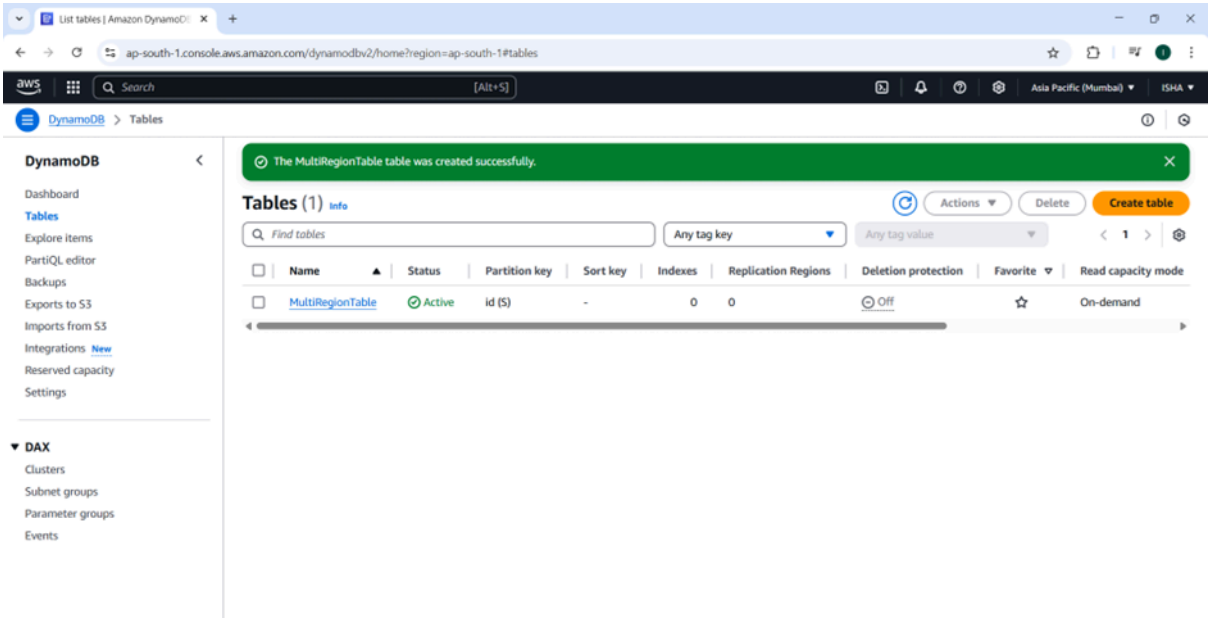
- Enable Point-in-Time Recovery (PITR)
  Why PITR?
  - PITR allows automatic backups of data **every second** for up to **35 days**.
  - In case of accidental deletion or corruption, you can restore data from any point in time.
  - This adds an extra layer of **data protection** and helps recover from human errors or system failures.




List tables | Amazon DynamoDB
ap-south-1.console.aws.amazon.com/dynamodbv2/home?region=ap-south-1#tables
Search
[Alt+S]
Asia Pacific (Mumbai)
ISHA
DynamoDB > Tables
DynamoDB
Dashboard
Tables
Explore items
PartiQL editor
Backups
Exports to S3
Imports from S3
Integrations New
Reserved capacity
Settings
DAX
Clusters
Subnet groups
Parameter groups
Events
The MultiRegionTable table was created successfully.
Tables (1) Info
Actions
Delete
Create table
Find tables
Any tag key
Any tag value
Name
Status
Partition key
Sort key
Indexes
Replication Regions
Deletion protection
Favorite
Read capacity mode
MultiRegionTable
Active
id (S)
-
0
0
Off
On-demand


List tables | Amazon DynamoDB
ap-south-1.console.aws.amazon.com/dynamodbv2/home?region=ap-south-1#tables
Search
[Alt+S]
Asia Pacific (Mumbai)
ISHA
DynamoDB > Tables
DynamoDB
Dashboard
Tables
Explore items
PartiQL editor
Backups
Exports to S3
Imports from S3
Integrations New
Reserved capacity
Settings
DAX
Clusters
Subnet groups
Parameter groups
Events
The MultiRegionTable table was created successfully.
Tables (1) Info
Actions
Delete
Create table
Find tables
Any tag key
Any tag value
Name
Status
Partition key
Sort key
Indexes
Replication Regions
Deletion protection
Favorite
Read capacity mode
MultiRegionTable
Active
id (S)
-
0
0
Off
On-demand





1. Enable Global Tables:

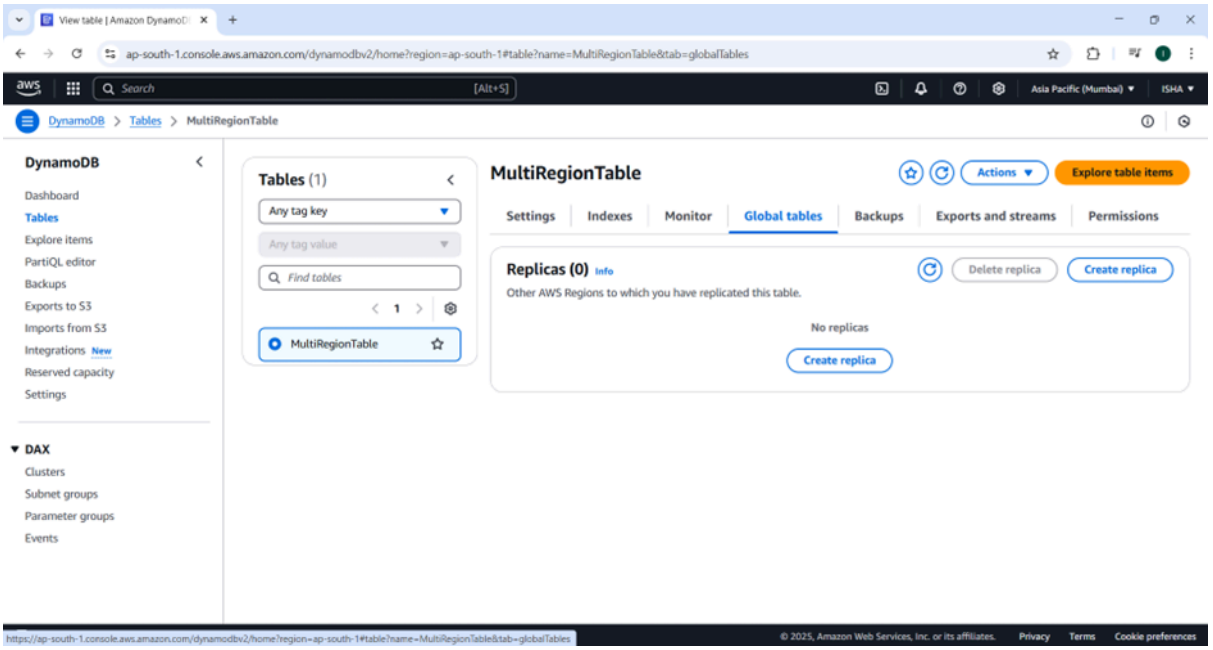
   Why do we need Global Tables?

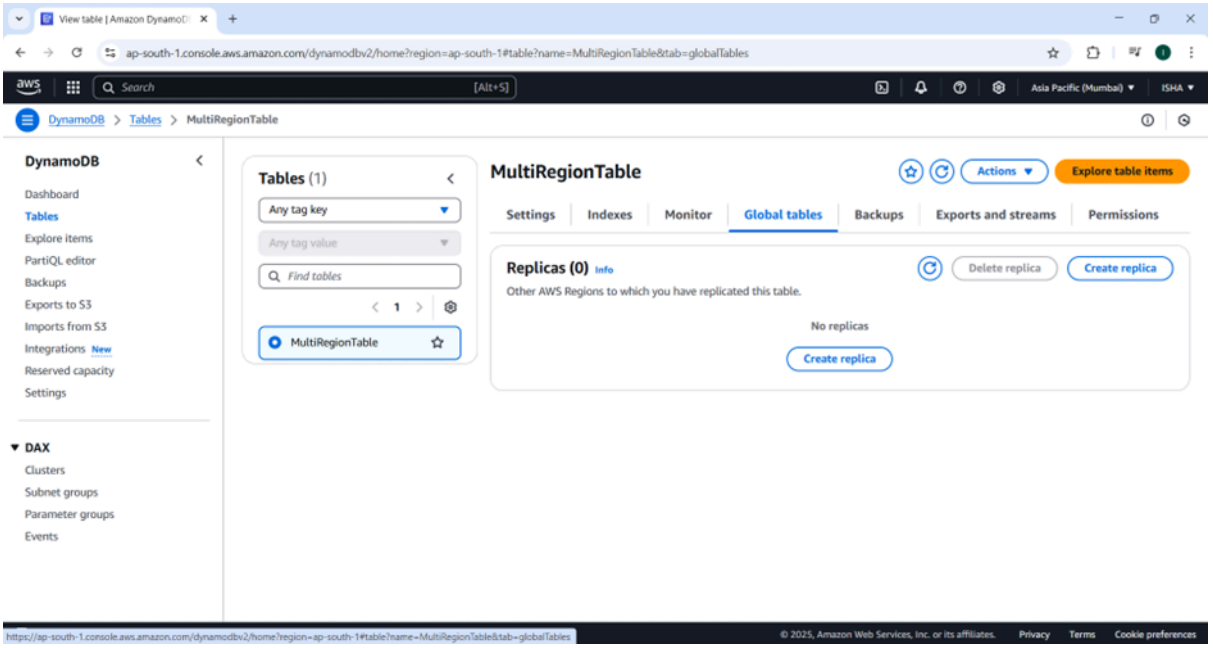
   - If Mumbai (ap-south-1) faces an outage, data should still be available in another AWS region (Singapore in this case).

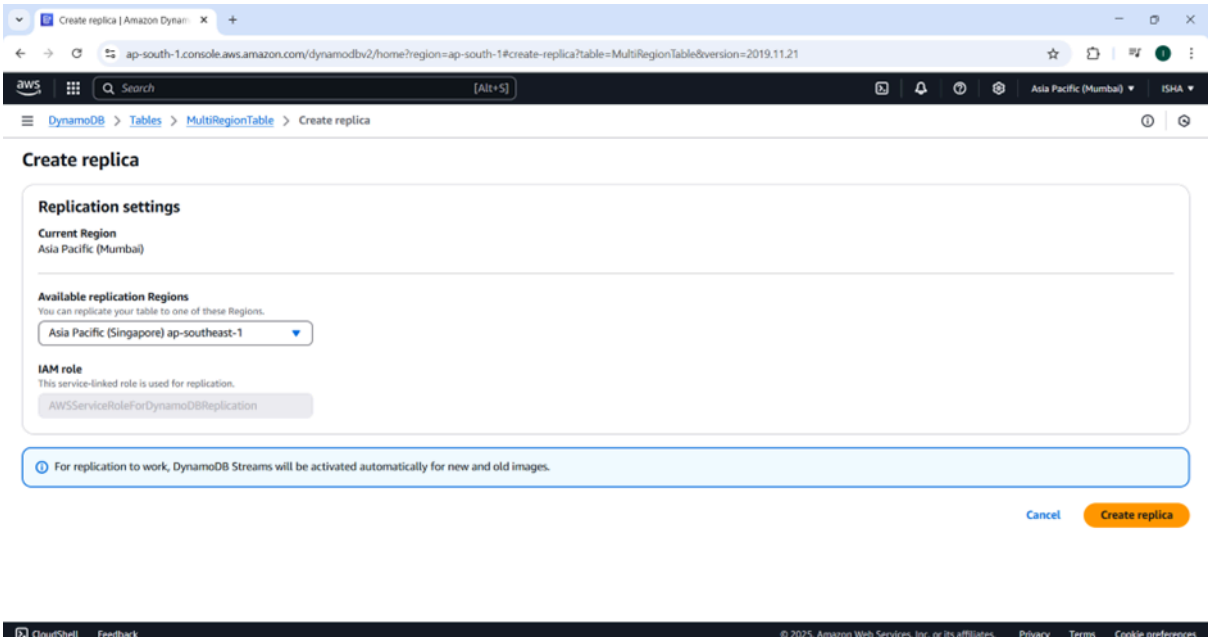
- Global Tables automatically replicate data across multiple AWS regions to keep backups updated in real-time.
- It ensures low-latency access for users in different geographic locations.

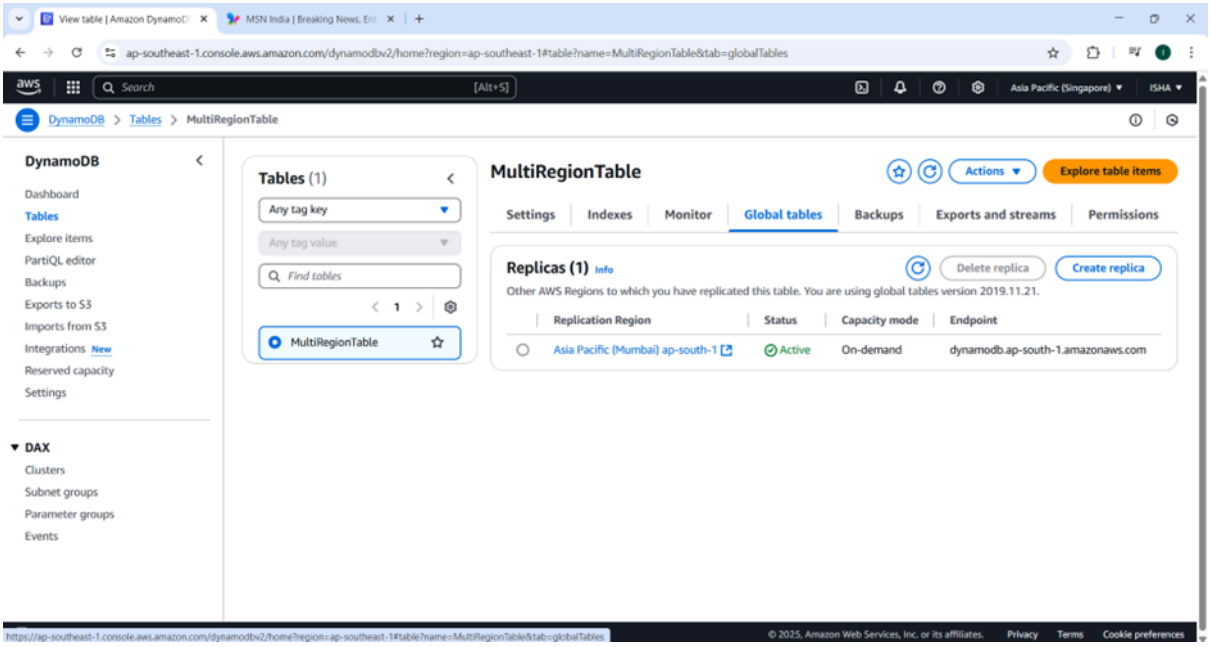
Adding Replica Region: Singapore (ap-southeast-1)

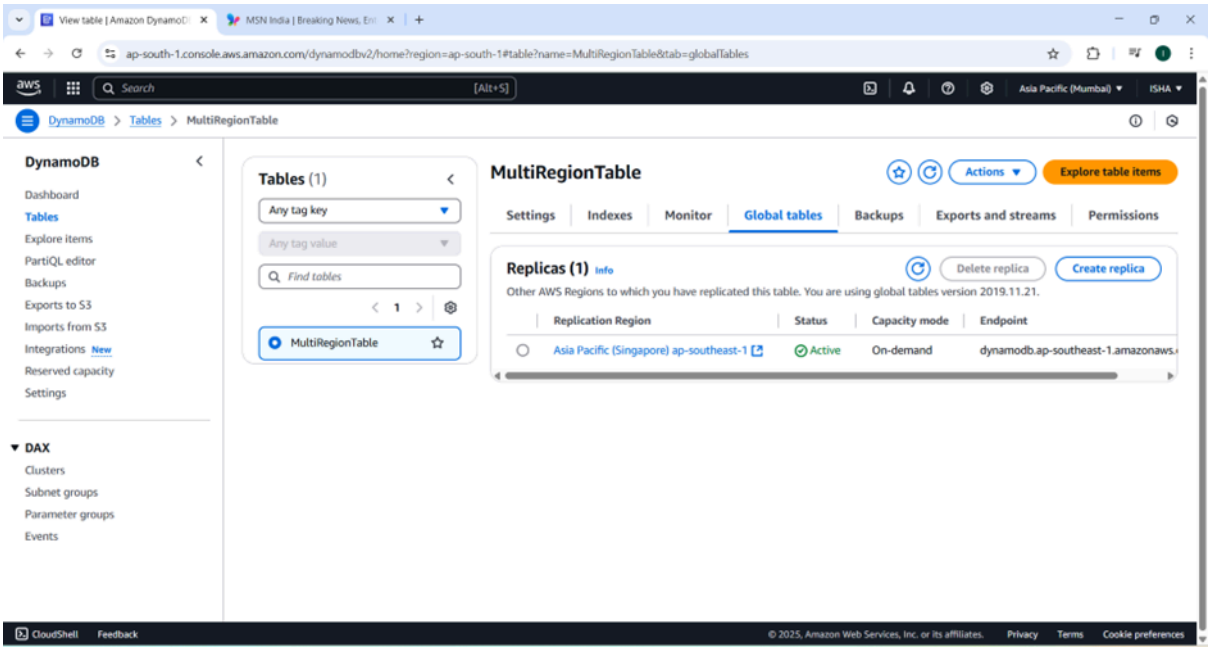
- Singapore is chosen as the **backup region** to ensure that data remains accessible even if Mumbai goes down.
- This enables a **multi-region disaster recovery setup**, where users can switch to Singapore if needed.
- Go to Global Tables → Add replica region: Singapore (ap-southeast-1).
- Wait ~10 mins for replication to activate.




View table | Amazon DynamoDB
ap-south-1.console.aws.amazon.com/dynamodbv2/home?region=ap-south-1#table?name=MultiRegionTable&tab=globalTables
Search
[Alt+S]
Asia Pacific (Mumbai)
ISHA
DynamoDB > Tables > MultiRegionTable
DynamoDB
Dashboard
Tables
Explore items
PartiQL editor
Backups
Exports to S3
Imports from S3
Integrations New
Reserved capacity
Settings
DAX
Clusters
Subnet groups
Parameter groups
Events
Tables (1)
Any tag key
Any tag value
Find tables
1
MultiRegionTable
MultiRegionTable
Actions
Explore table items
Settings
Indexes
Monitor
Global tables
Backups
Exports and streams
Permissions
Replicas (0) Info
Other AWS Regions to which you have replicated this table.
Delete replica
Create replica
No replicas
Create replica
https://ap-south-1.console.aws.amazon.com/dynamodbv2/home?region=ap-south-1#table?name=MultiRegionTable&tab=globalTables
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


Create replica | Amazon DynamoDB
ap-south-1.console.aws.amazon.com/dynamodbv2/home?region=ap-south-1#create-replica?table=MultiRegionTable&version=2019.11.21
Search
[Alt+S]
Asia Pacific (Mumbai)
ISHA
DynamoDB > Tables > MultiRegionTable > Create replica
Create replica
Replication settings
Current Region
Asia Pacific (Mumbai)
Available replication Regions
You can replicate your table to one of these Regions.
Asia Pacific (Singapore) ap-southeast-1
IAM role
This service-linked role is used for replication.
AWSServiceRoleForDynamoDBReplication
For replication to work, DynamoDB Streams will be activated automatically for new and old images.
Cancel
Create replica
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

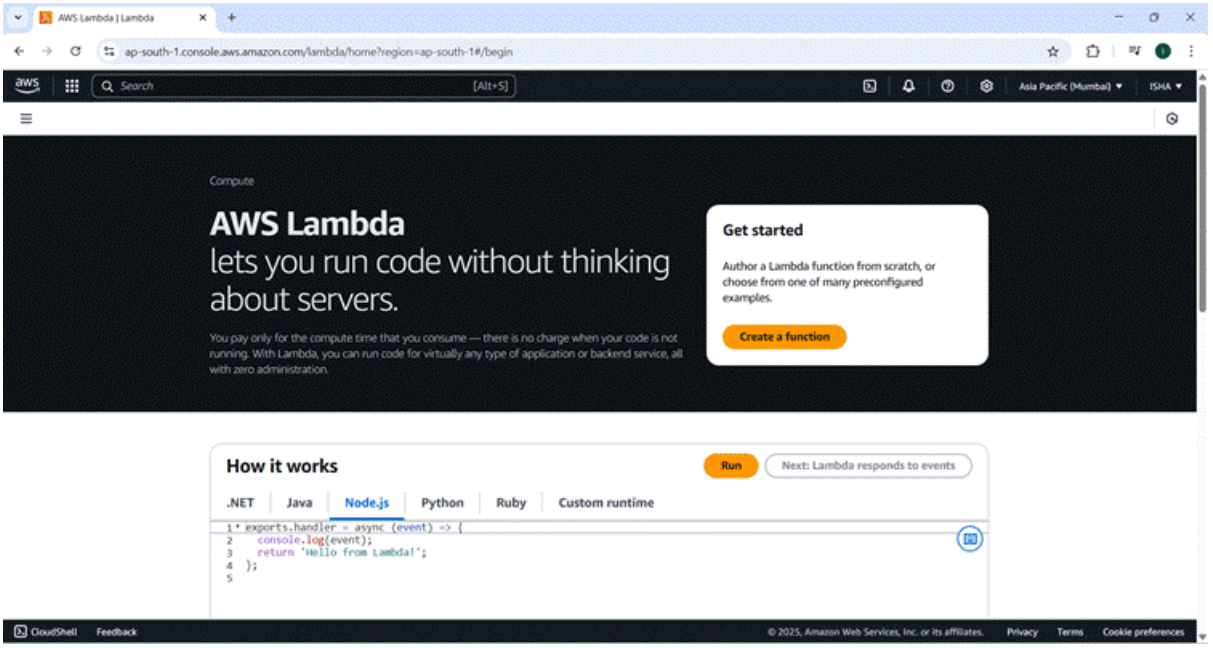




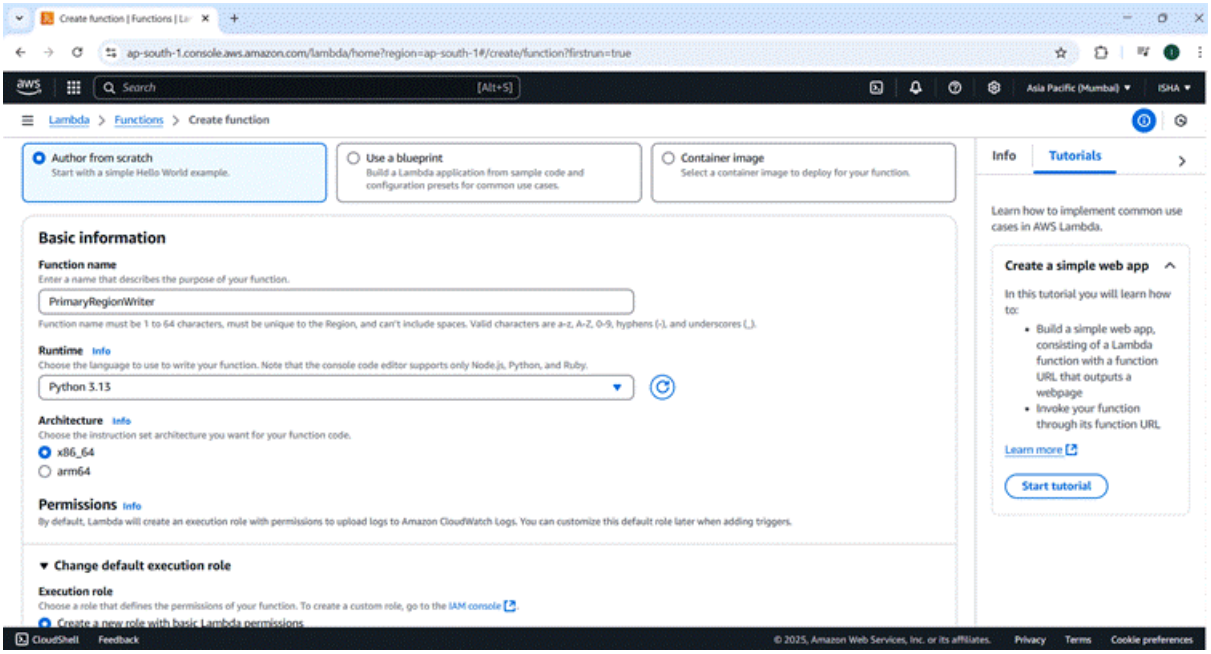
Step 2: Deploy Lambda Functions

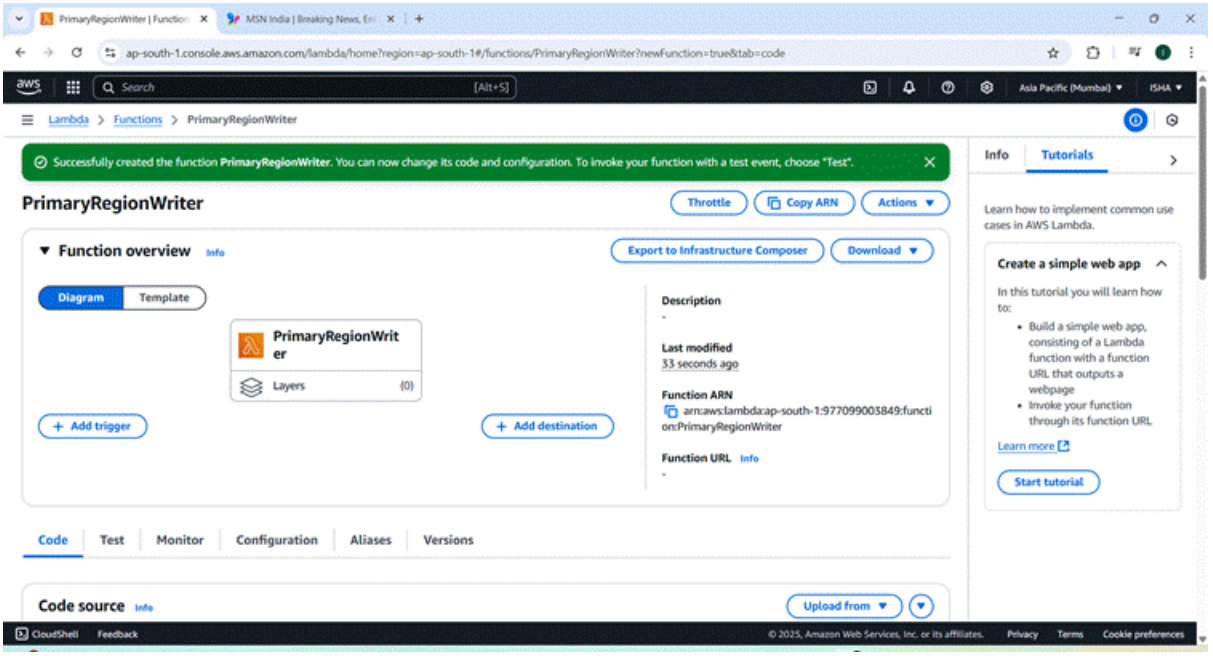
Primary Region (Mumbai)

1. Create Lambda Function:
    - This function automatically writes data to DynamoDB in the Mumbai region (ap-south-1).
    - Helps store important data so that it can be used later.

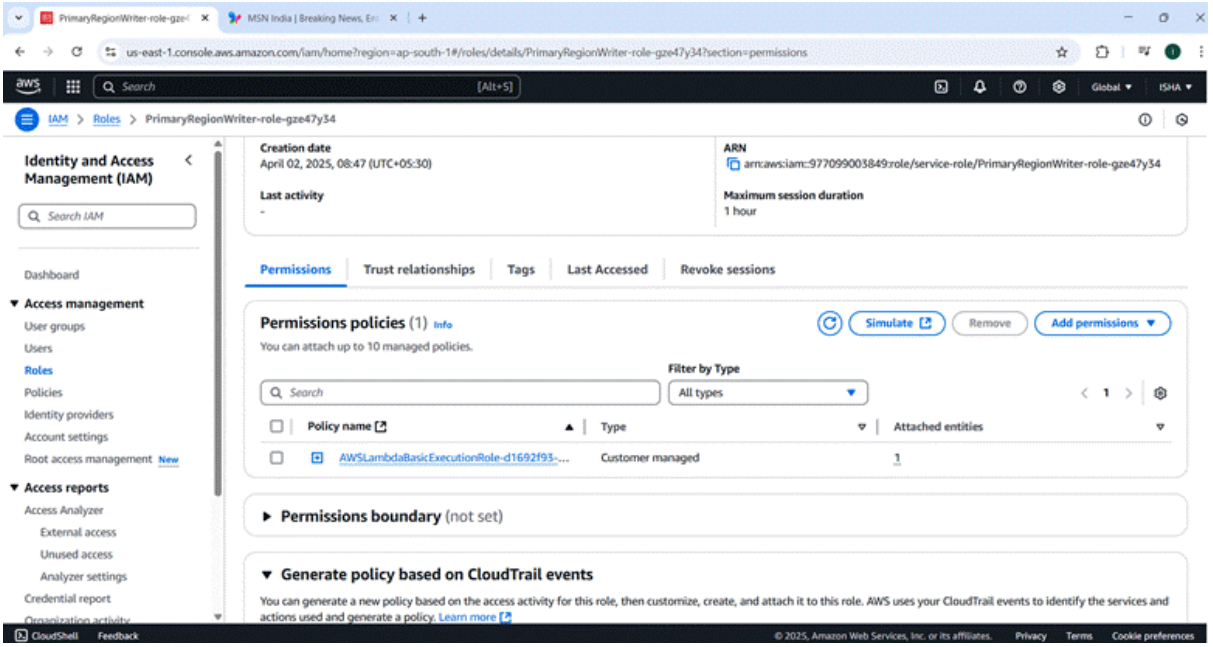


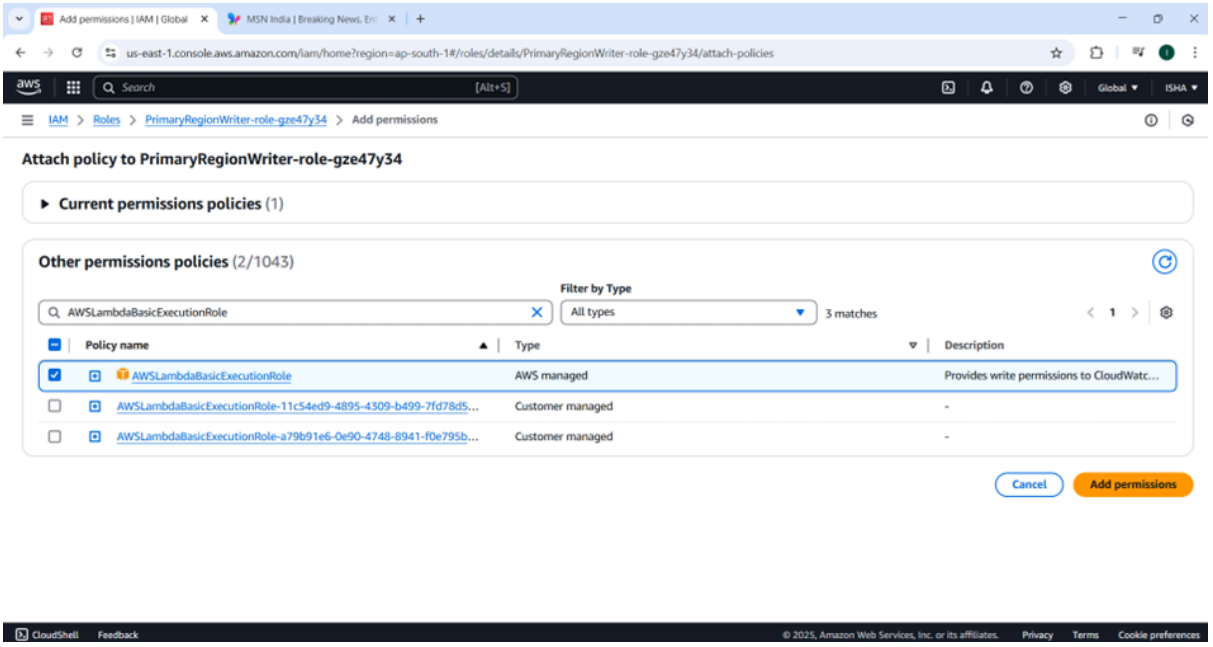
- Name: PrimaryRegionWriter
- Runtime: Python 3.12

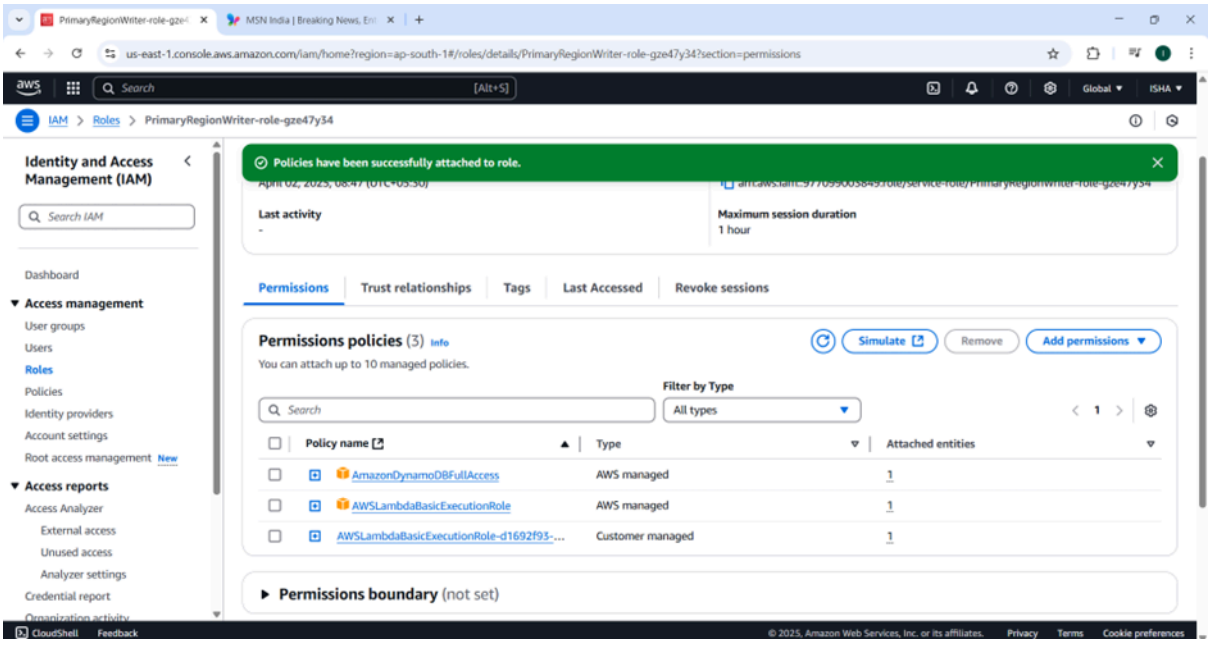




- IAM Role:
  - Attach AmazonDynamoDBFullAccess (for testing):Allows the function to store data in DynamoDB.
  - Attach AWSLambdaBasicExecutionRole (for logs):Allows logging and monitoring in CloudWatch (helps in debugging).







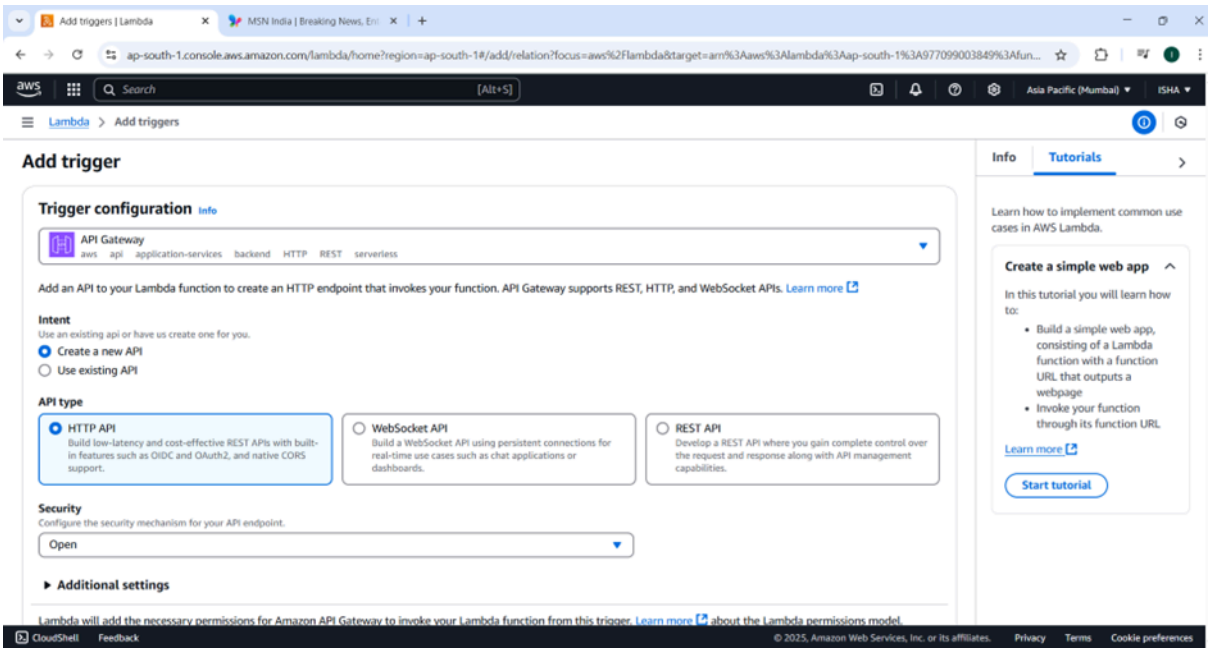
1. Lambda Code:
    - Uses boto3 (AWS SDK for Python) to connect to DynamoDB in Mumbai.
    - Generates a unique ID using `uuid` and saves a message to the table.
    - Returns a success message confirming the data is stored.

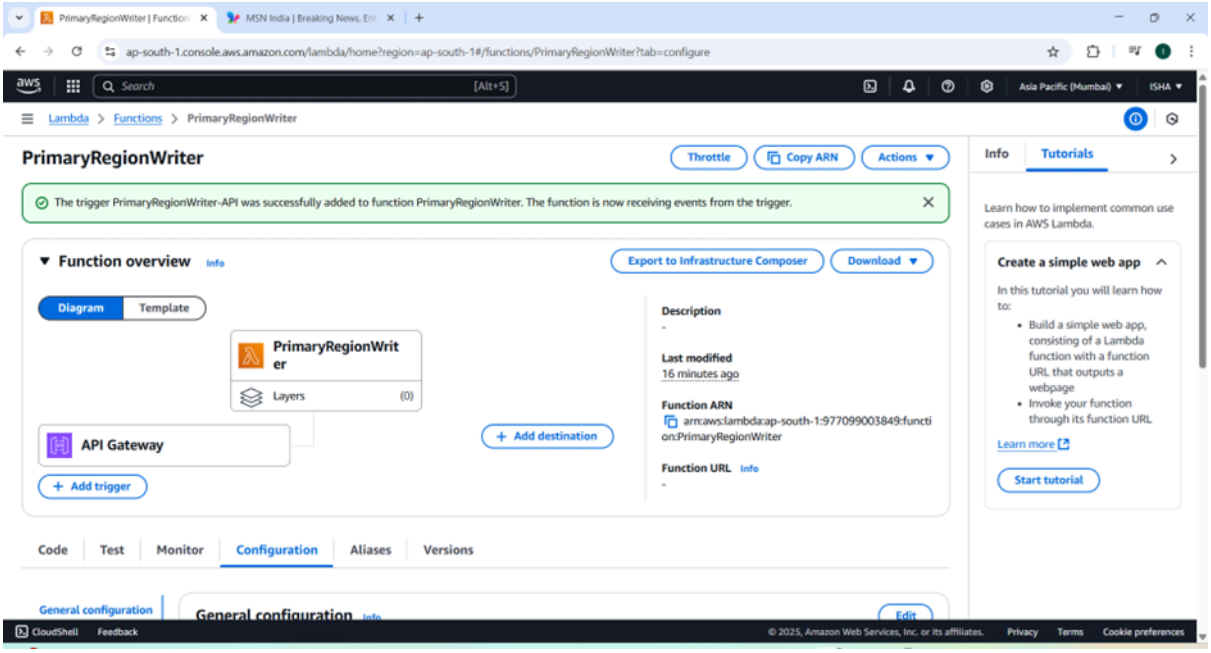
```
import boto3

import uuid

def lambda_handler(event, context):


    dynamodb=boto3.resource('dynamodb',region_name='ap-so
    uth-1')

    table = dynamodb.Table('MultiRegionTable')

    item_id = str(uuid.uuid4())

    table.put_item(Item={

        'id': item_id,

        'region': 'ap-south-1',

        'message': 'Data from Mumbai Lambda'

    })

    return {

     'statusCode': 200,

      'body': f'Success! Inserted item {item_id} into DynamoDB'

     }
```

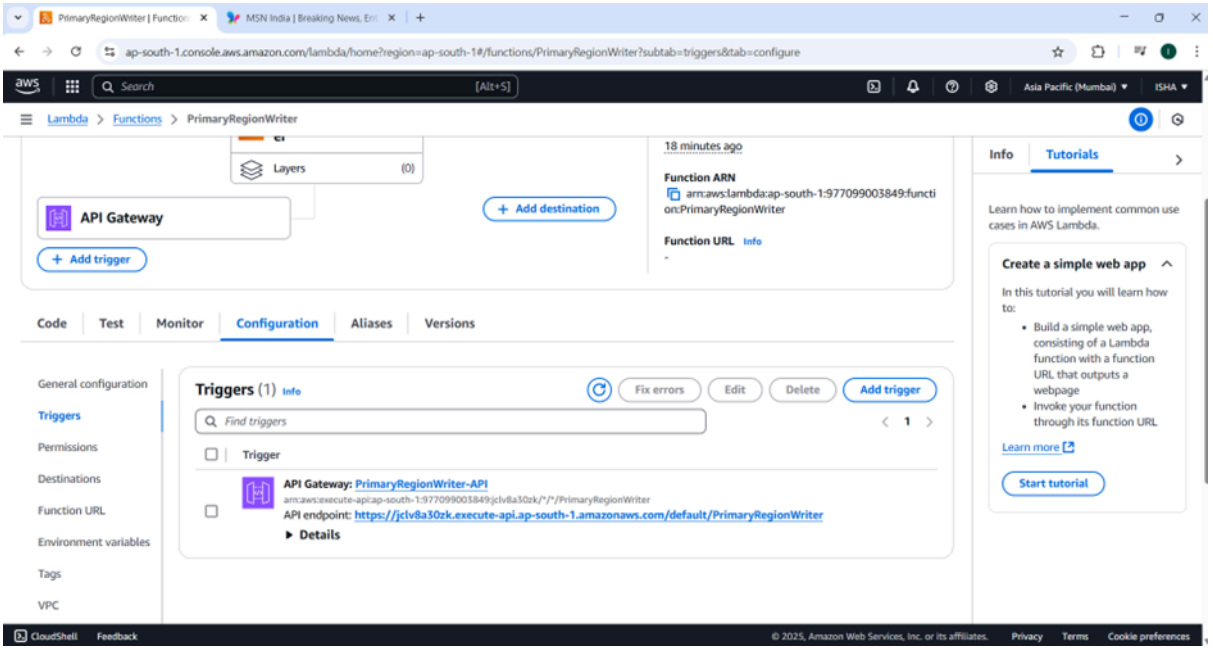
2. Add API Gateway Trigger:
    - **Purpose**: Allows external applications to trigger the Lambda function via an **HTTP request**.
    - **Why?** So users or other systems can **send data to Mumbai's database** through a simple API call.

API Type: HTTP API

Endpoint:
https://[api-id].execute-api.ap-south-1.amazonaws.com/default/PrimaryRegionWriter







Secondary Region (Singapore)

This acts as a backup in case the Mumbai region fails.

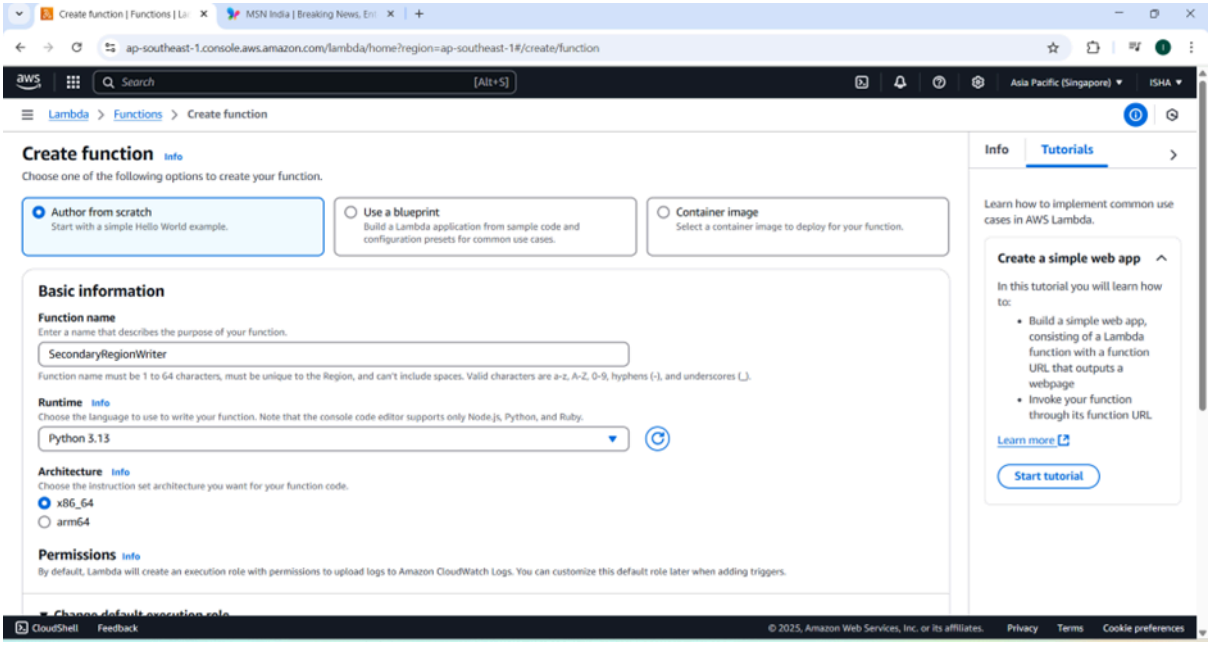
The function works the same way but stores data in DynamoDB in Singapore (ap-southeast-1) instead.

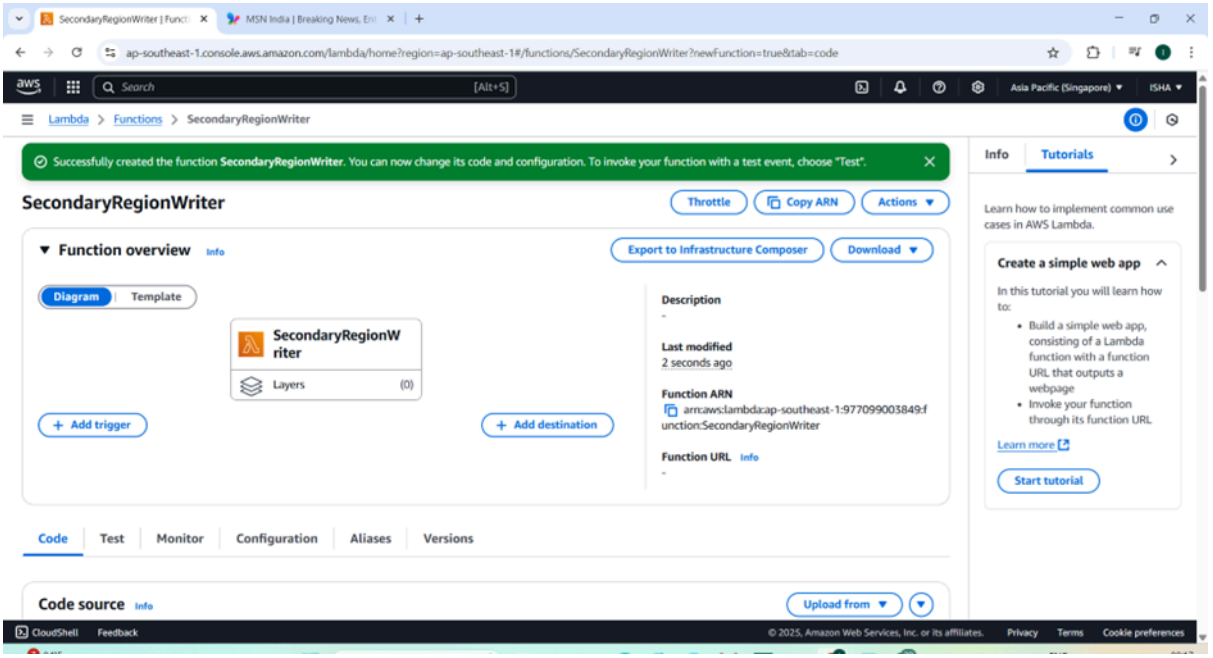
The DynamoDB connection is updated to point to Singapore (`ap-southeast-1`).

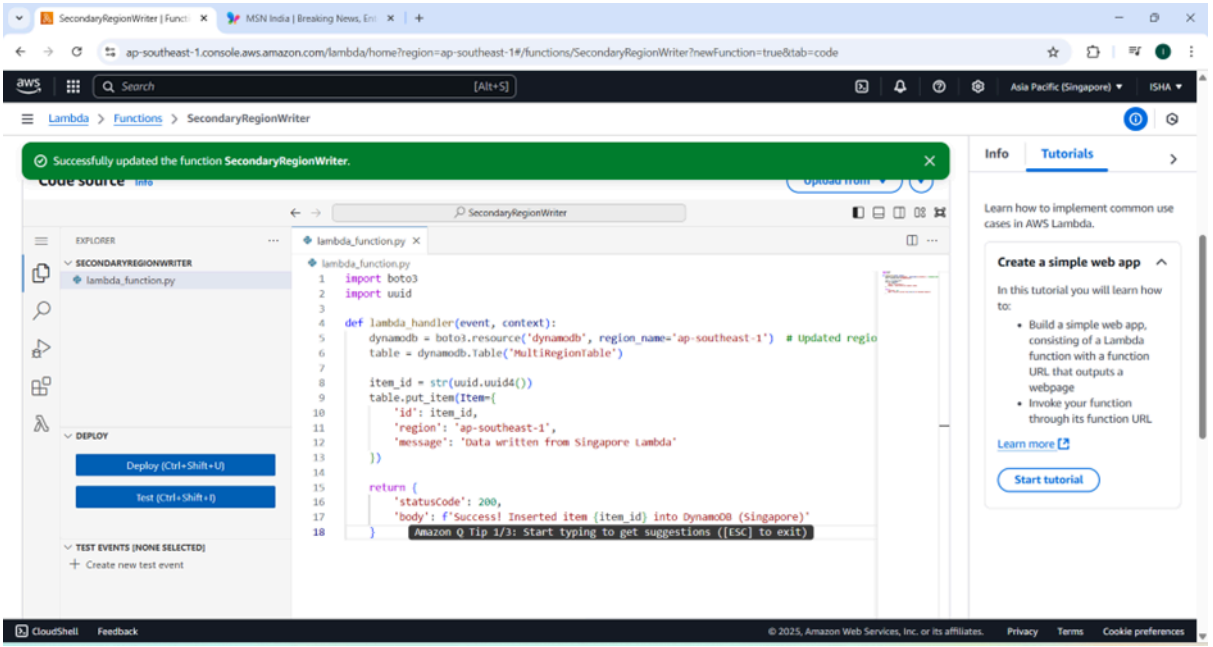
This ensures the data is stored in the secondary region for disaster recovery.

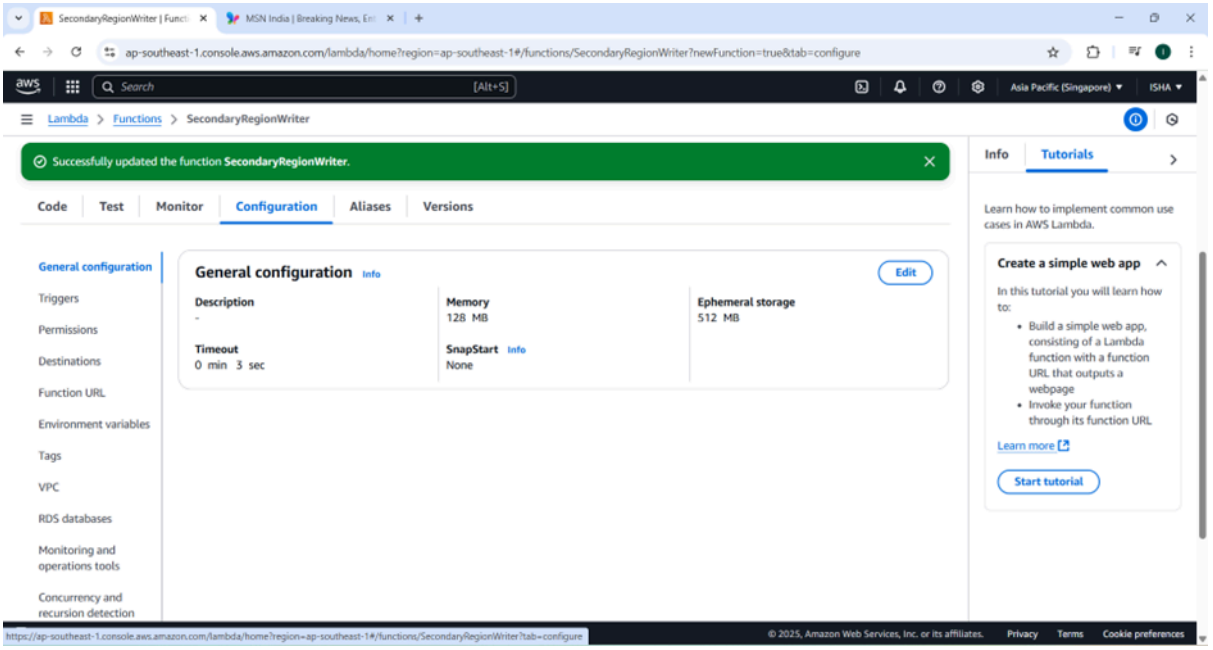
1. Repeat the same steps with:
    - Lambda Name: SecondaryRegionWriter
    - Region: ap-southeast-1
    - Code Update:

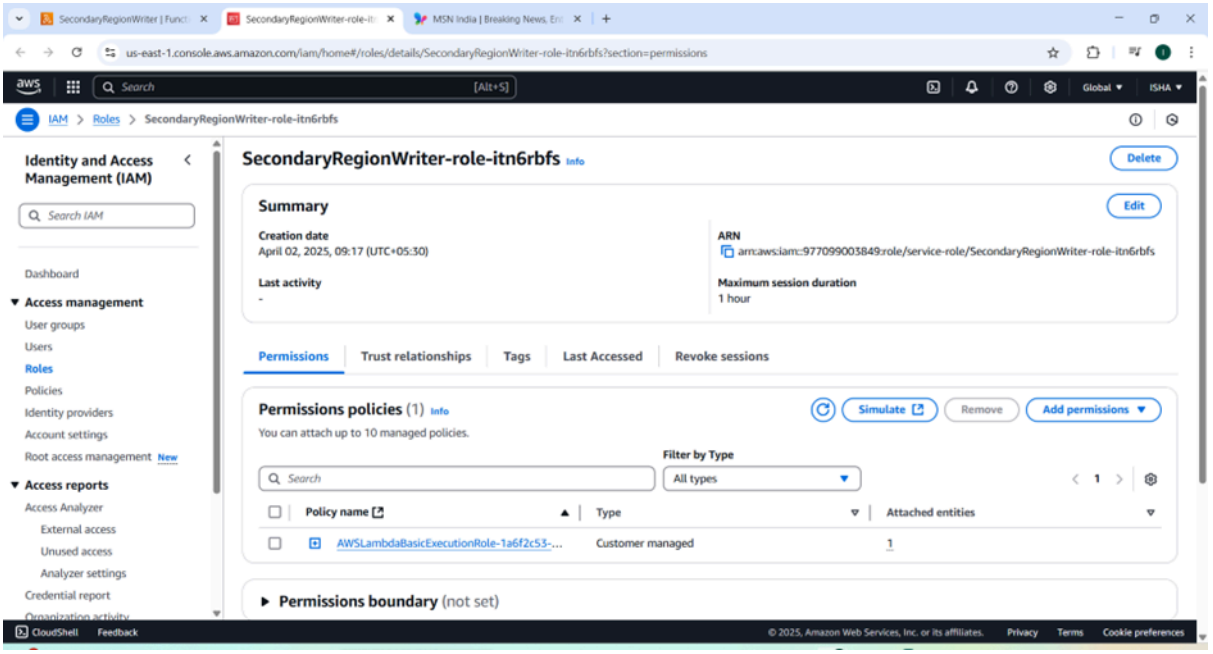
    dynamodb = boto3.resource('dynamodb', region_name='ap-southeast-1')

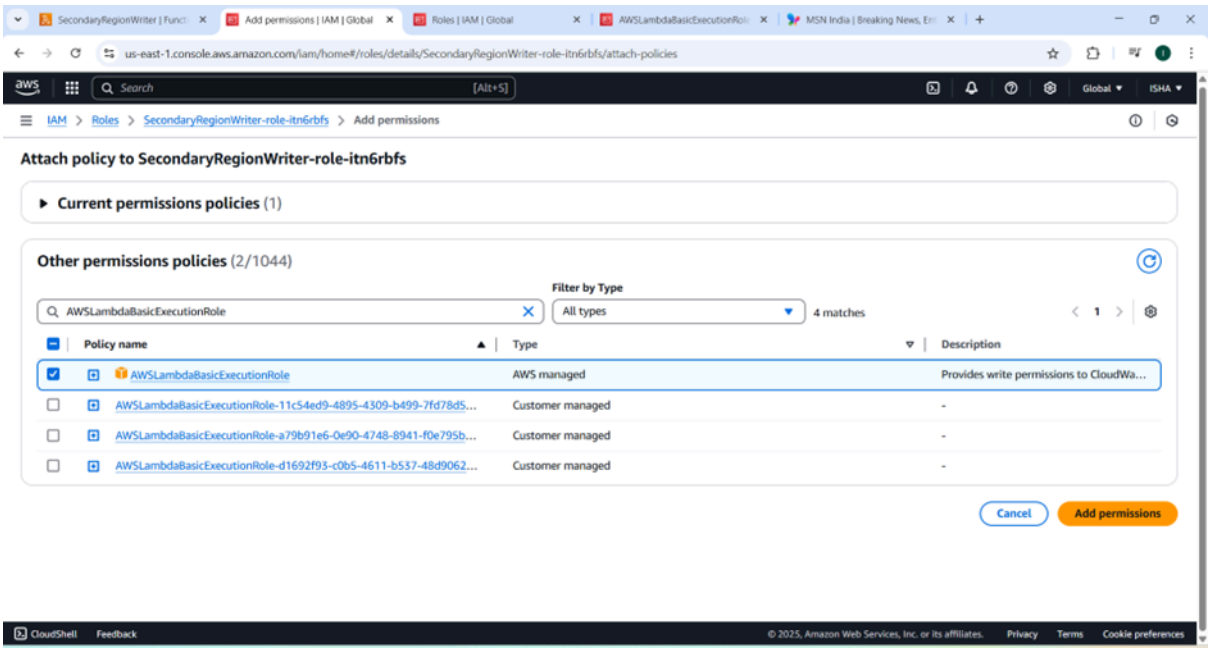

Create function | Functions | La
MSN India | Breaking News, En
ap-southeast-1.console.aws.amazon.com/lambda/home?region=ap-southeast-1#/create/function
Search
[Alt+S]
Asia Pacific (Singapore)
ISHA
Lambda > Functions > Create function
Create function Info
Choose one of the following options to create 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.
SecondaryRegionWriter
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 3.13
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.
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

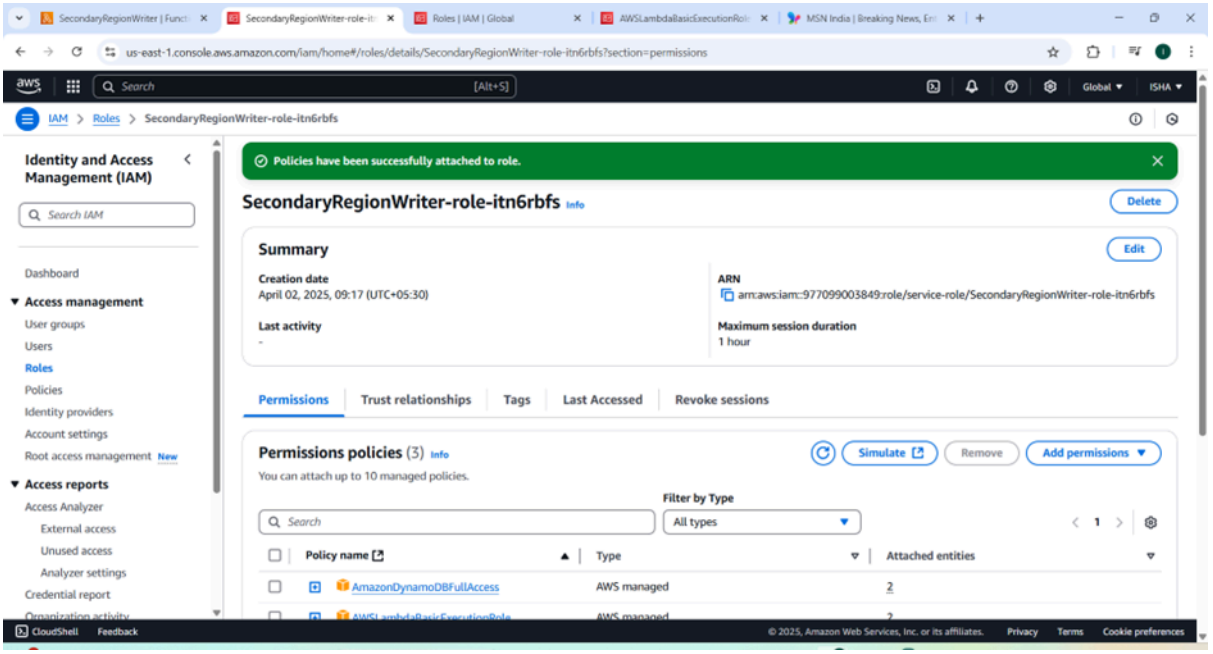

SecondaryRegionWriter | Funct
MSN India | Breaking News, En
ap-southeast-1.console.aws.amazon.com/lambda/home?region=ap-southeast-1#/functions/SecondaryRegionWriter?newFunction=true&tab=code
Search
[Alt+S]
Asia Pacific (Singapore)
ISHA
Lambda > Functions > SecondaryRegionWriter
Successfully created the function SecondaryRegionWriter. You can now change its code and configuration. To invoke your function with a test event, choose "Test".
SecondaryRegionWriter
Throttle
Copy ARN
Actions
Function overview Info
Export to Infrastructure Composer
Download
Diagram
Template
SecondaryRegionWriter
Layers
(0)
Add trigger
Add destination
Description
-
Last modified
2 seconds ago
Function ARN
arn:aws:lambda:ap-southeast-1:977099003849:function:SecondaryRegionWriter
Function URL Info
-
Code
Test
Monitor
Configuration
Aliases
Versions
Code source Info
Upload from
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

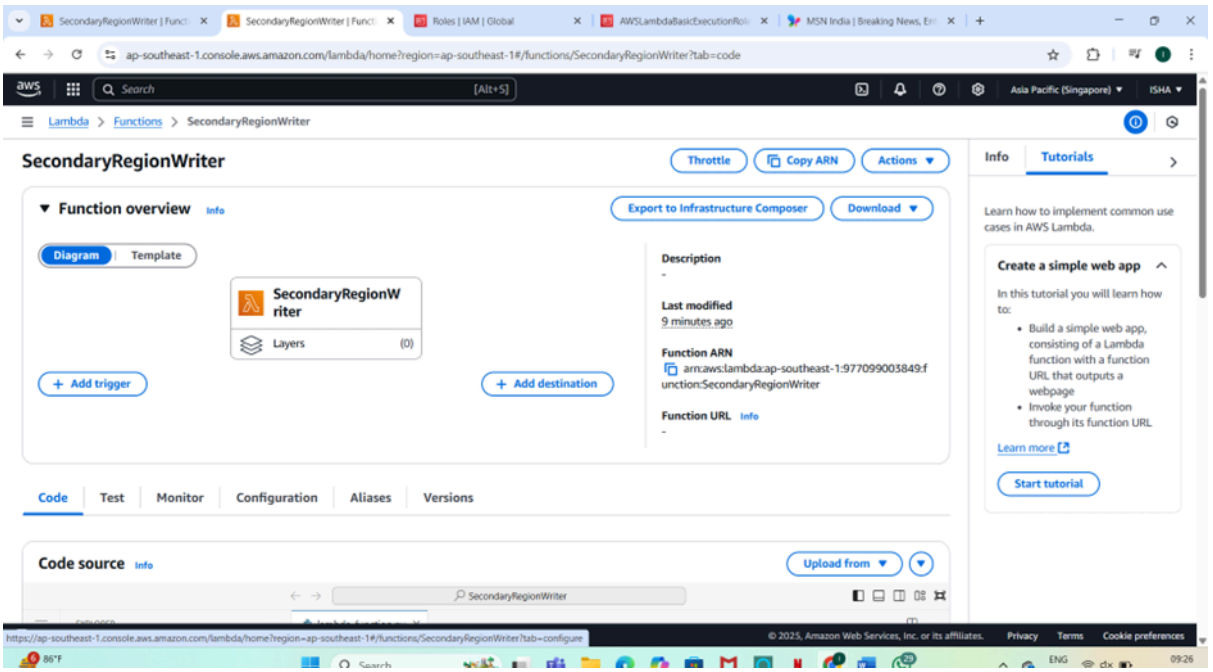

Successfully updated the function SecondaryRegionWriter.
lambda_function.py
import boto3
import uuid
def lambda_handler(event, context):
dynamodb = boto3.resource('dynamodb', region_name='ap-southeast-1') # Updated regio
table = dynamodb.Table('MultiRegionTable')
item_id = str(uuid.uuid4())
table.put_item(Item={
'id': item_id,
'region': 'ap-southeast-1',
'message': 'Data written from Singapore Lambda'
})
return {
'statusCode': 200,
'body': f'Success! Inserted item {item_id} into DynamoDB (Singapore)'
}
Amazon Q Tip 1/3: Start typing to get suggestions ([ESC] to exit)
Deploy (Ctrl+Shift+U)
Test (Ctrl+Shift+I)
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

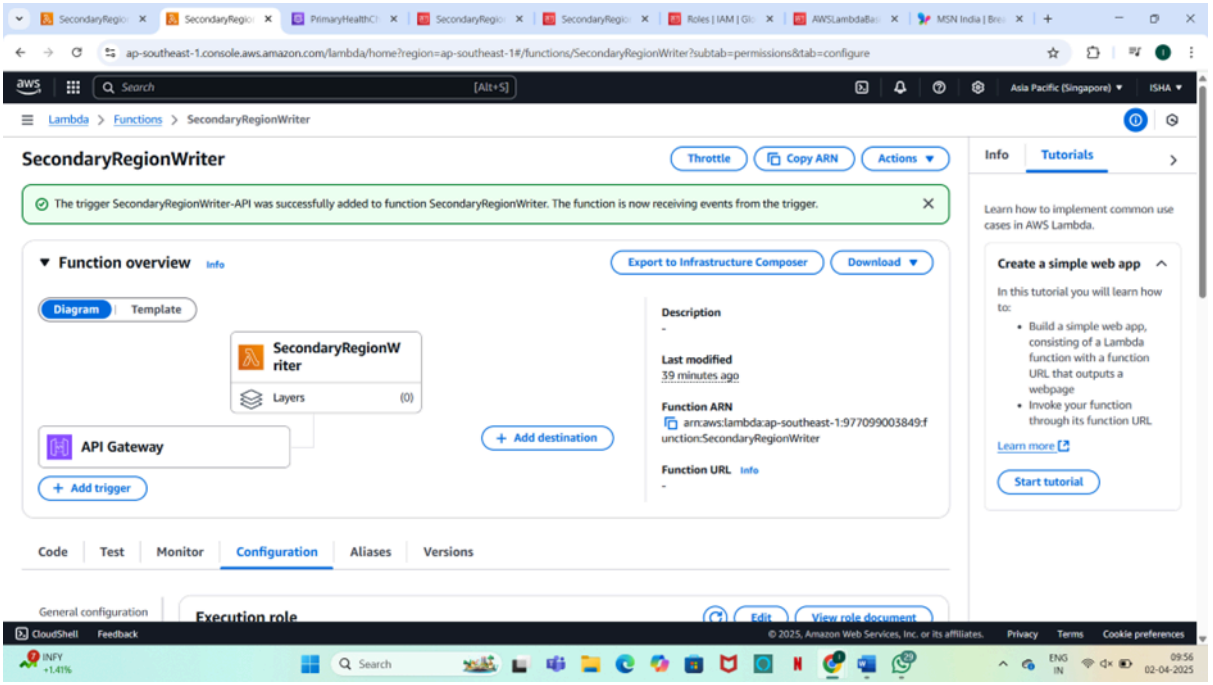

Successfully updated the function SecondaryRegionWriter.
Code
Test
Monitor
Configuration
Aliases
Versions
General configuration
Triggers
Permissions
Destinations
Function URL
Environment variables
Tags
VPC
RDS databases
Monitoring and operations tools
Concurrency and recursion detection
General configuration Info
Edit
Description
-
Memory
128 MB
Ephemeral storage
512 MB
Timeout
0 min 3 sec
SnapStart Info
None
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

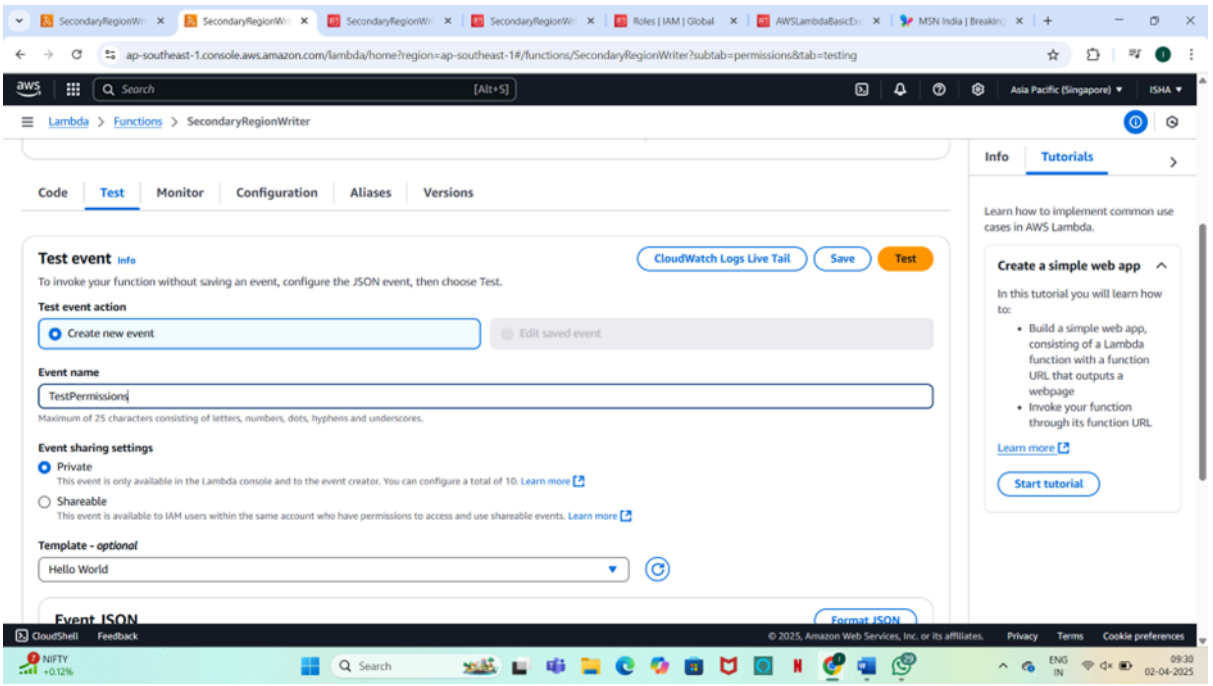

SecondaryRegionWriter-role-itn6rbfs
Delete
Summary
Edit
Creation date
April 02, 2025, 09:17 (UTC+05:30)
ARN
arn:aws:iam::977099003849:role/service-role/SecondaryRegionWriter-role-itn6rbfs
Last activity
-
Maximum session duration
1 hour
Permissions
Trust relationships
Tags
Last Accessed
Revoke sessions
Permissions policies (1)
You can attach up to 10 managed policies.
Simulate
Remove
Add permissions
Filter by Type
All types
Policy name
Type
Attached entities
AWSLambdaBasicExecutionRole-1a6f2c53-...
Customer managed
1
Permissions boundary (not set)

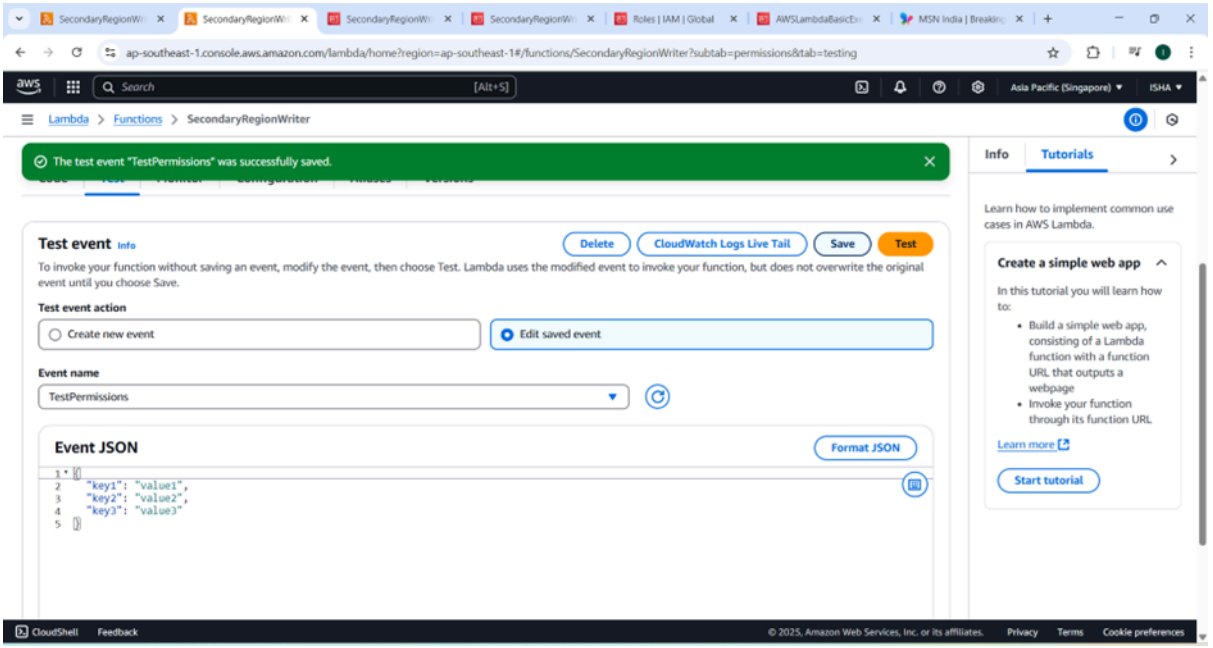

Attach policy to SecondaryRegionWriter-role-itn6rbfs
Current permissions policies (1)
Other permissions policies (2/1044)
Filter by Type
AWSLambdaBasicExecutionRole
All types
4 matches
Policy name
Type
Description
AWSLambdaBasicExecutionRole
AWS managed
Provides write permissions to CloudWa...
AWSLambdaBasicExecutionRole-11c54ed9-4895-4309-b499-7fd78d5...
Customer managed
AWSLambdaBasicExecutionRole-a79b91e6-0e90-4748-8941-f0e795b...
Customer managed
AWSLambdaBasicExecutionRole-d1692f93-c0b5-4611-b537-48d9062...
Customer managed
Cancel
Add permissions

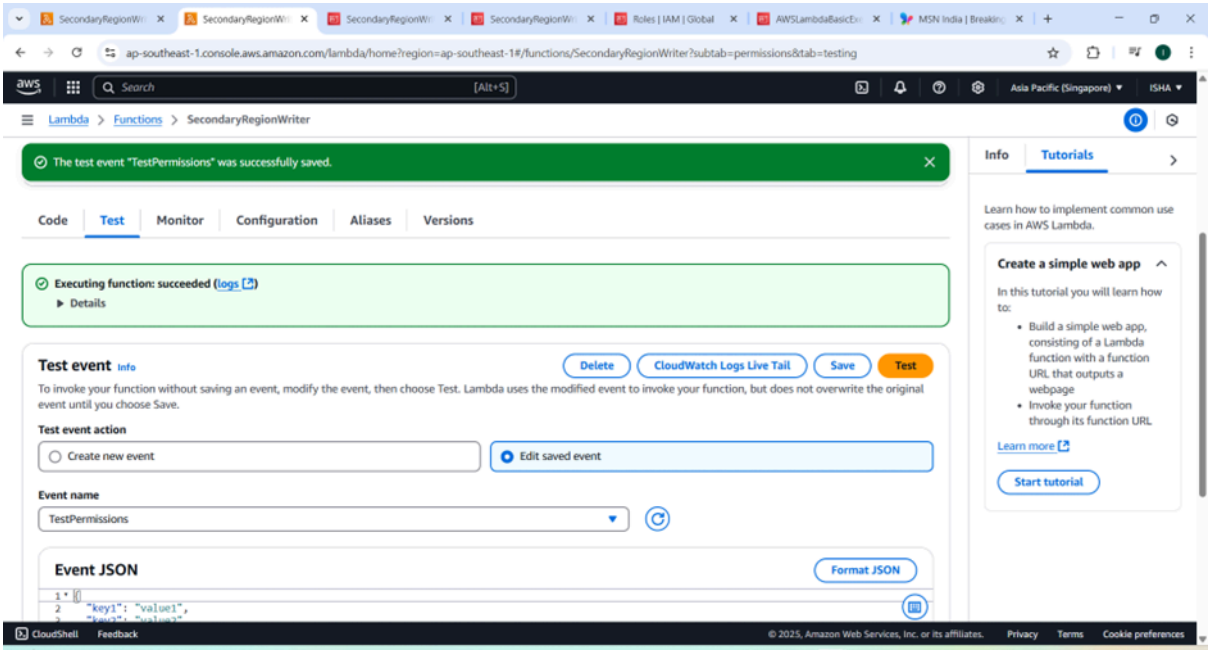

us-east-1.console.aws.amazon.com/iam/home#/roles/details/SecondaryRegionWriter-role-itn6rbfs?section=permissions
IAM > Roles > SecondaryRegionWriter-role-itn6rbfs
Identity and Access Management (IAM)
Search IAM
Dashboard
Access management
User groups
Users
Roles
Policies
Identity providers
Account settings
Root access management New
Access reports
Access Analyzer
External access
Unused access
Analyzer settings
Credential report
Policies have been successfully attached to role.
SecondaryRegionWriter-role-itn6rbfs Info
Delete
Summary
Edit
Creation date
April 02, 2025, 09:17 (UTC+05:30)
ARN
arn:aws:iam::977099003849:role/service-role/SecondaryRegionWriter-role-itn6rbfs
Last activity
-
Maximum session duration
1 hour
Permissions
Trust relationships
Tags
Last Accessed
Revoke sessions
Permissions policies (3) Info
You can attach up to 10 managed policies.
Simulate
Remove
Add permissions
Filter by Type
Search
All types
Policy name
Type
Attached entities
AmazonDynamoDBFullAccess
AWS managed
2
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

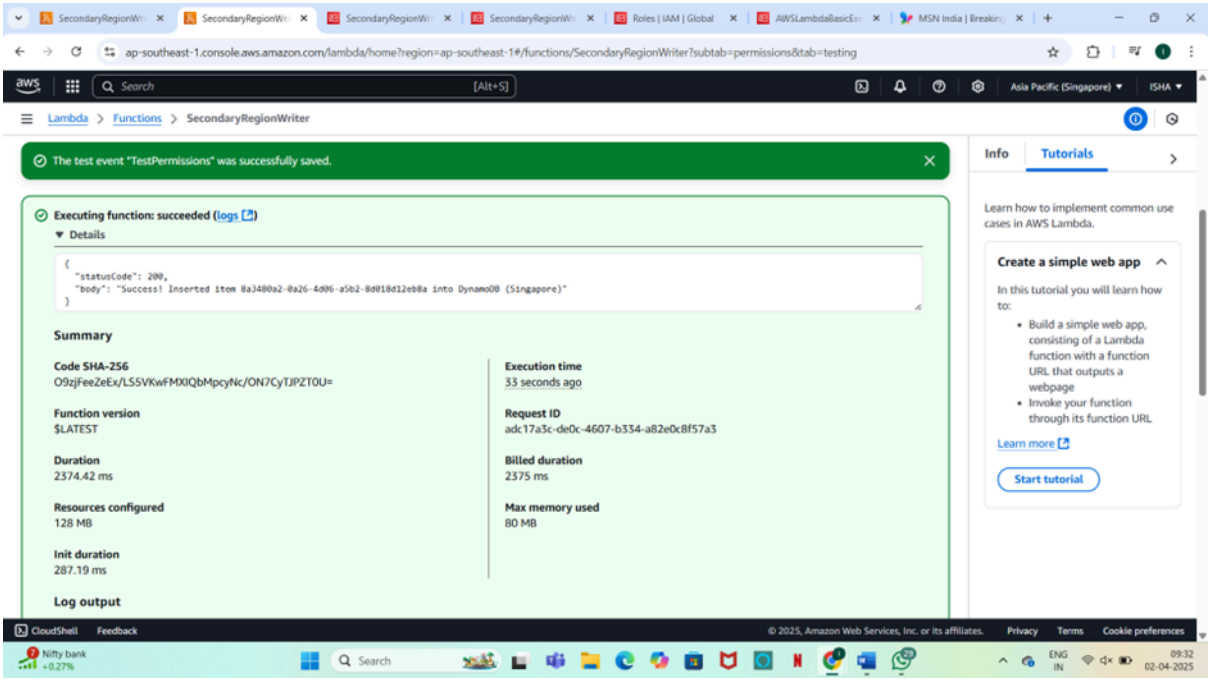

ap-southeast-1.console.aws.amazon.com/lambda/home?region=ap-southeast-1#/functions/SecondaryRegionWriter?tab=code
Lambda > Functions > SecondaryRegionWriter
SecondaryRegionWriter
Throttle
Copy ARN
Actions
Function overview Info
Export to Infrastructure Composer
Download
Diagram
Template
SecondaryRegionWriter
Layers (0)
+ Add trigger
+ Add destination
Description
-
Last modified
9 minutes ago
Function ARN
arn:aws:lambda:ap-southeast-1:977099003849:function:SecondaryRegionWriter
Function URL Info
-
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
Code
Test
Monitor
Configuration
Aliases
Versions
Code source Info
Upload from
SecondaryRegionWriter
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

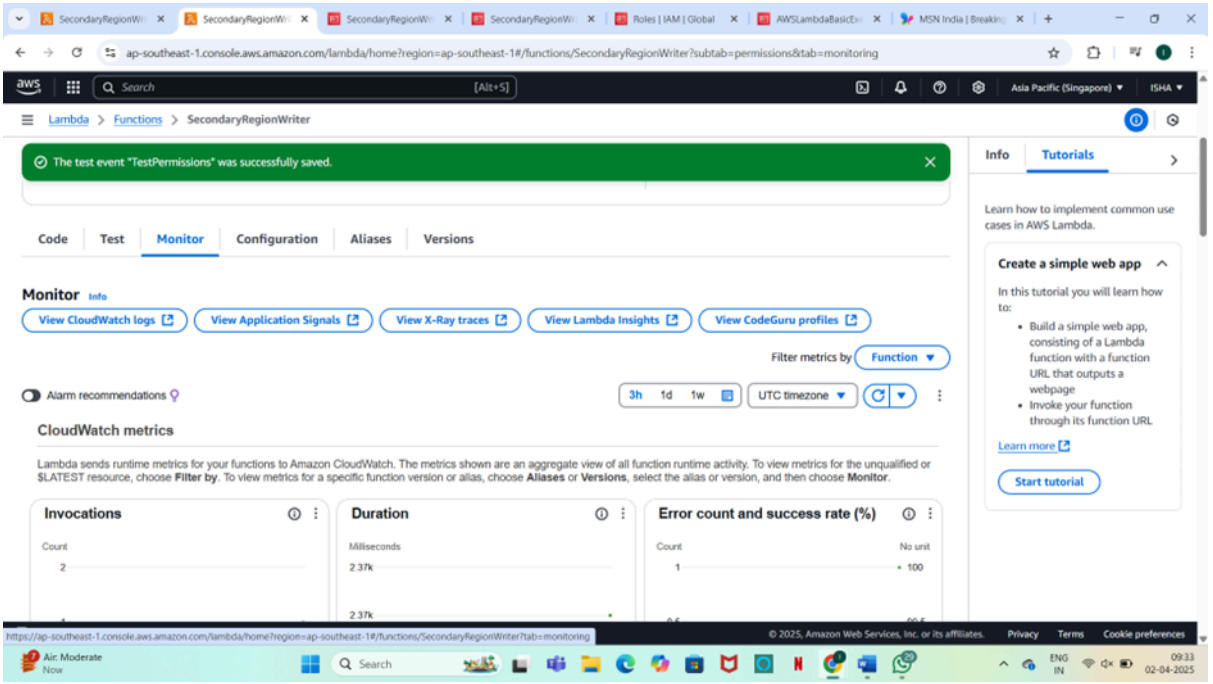

ap-southeast-1.console.aws.amazon.com/lambda/home?region=ap-southeast-1#/functions/SecondaryRegionWriter?subtab=permissions&tab=configure
Lambda > Functions > SecondaryRegionWriter
SecondaryRegionWriter
Throttle
Copy ARN
Actions
The trigger SecondaryRegionWriter-API was successfully added to function SecondaryRegionWriter. The function is now receiving events from the trigger.
Function overview
Export to Infrastructure Composer
Download
Diagram
Template
SecondaryRegionWriter
Layers
(0)
API Gateway
Add destination
Add trigger
Description
Last modified
39 minutes ago
Function ARN
arn:aws:lambda:ap-southeast-1:977099003849:function:SecondaryRegionWriter
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

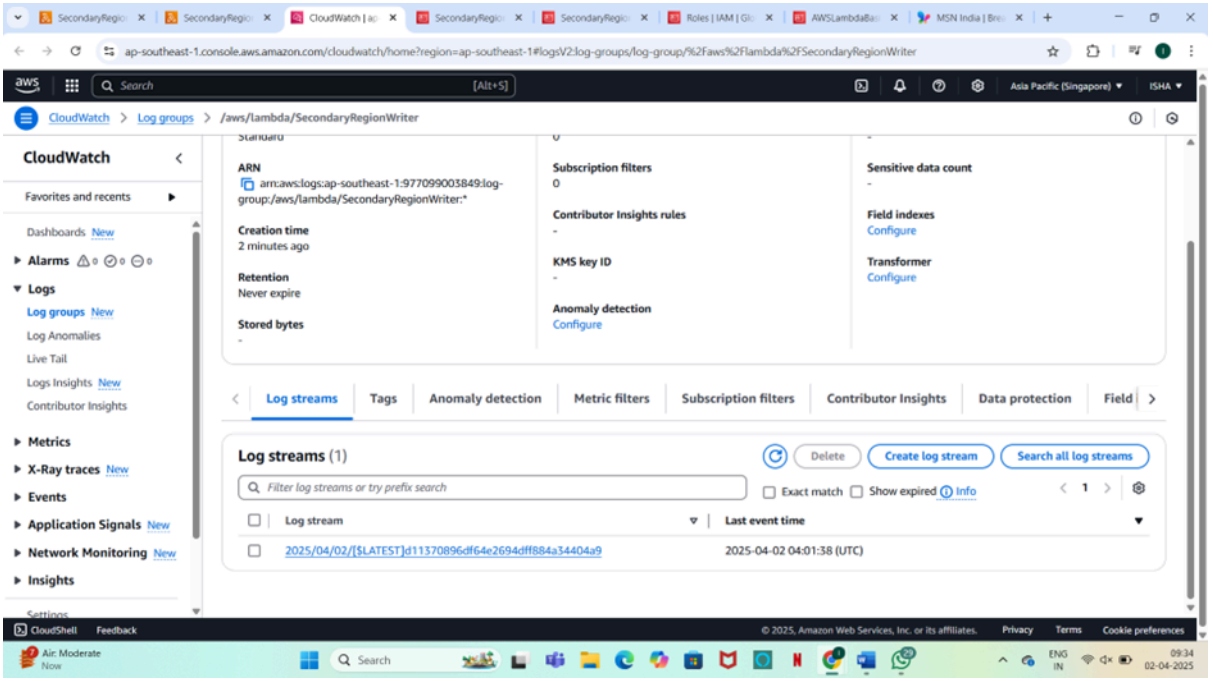

ap-southeast-1.console.aws.amazon.com/lambda/home?region=ap-southeast-1#/functions/SecondaryRegionWriter?subtab=permissions&tab=testing
Lambda > Functions > SecondaryRegionWriter
Code
Test
Monitor
Configuration
Aliases
Versions
Test event
Info
CloudWatch Logs Live Tail
Save
Test
To invoke your function without saving an event, configure the JSON event, then choose Test.
Test event action
Create new event
Edit saved event
Event name
TestPermissions
Maximum of 25 characters consisting of letters, numbers, dots, hyphens and underscores.
Event sharing settings
Private
This event is only available in the Lambda console and to the event creator. You can configure a total of 10.
Learn more
Shareable
This event is available to IAM users within the same account who have permissions to access and use shareable events.
Learn more
Template - optional
Hello World
Event JSON
Format JSON
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

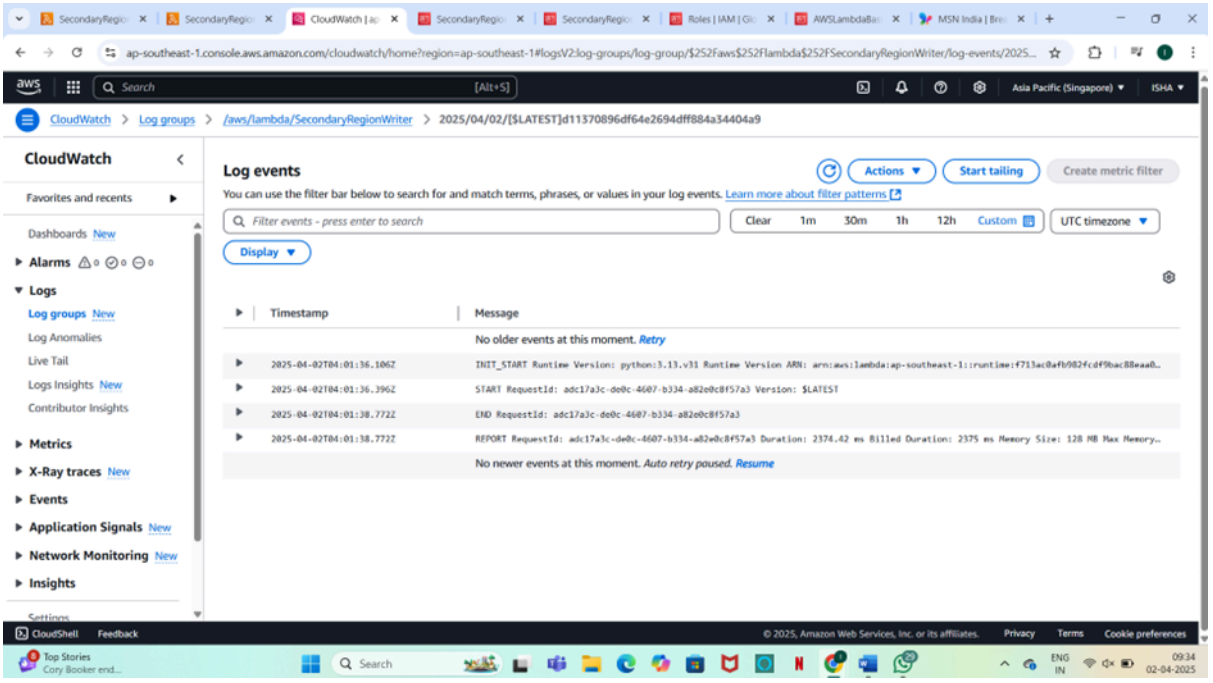

ap-southeast-1.console.aws.amazon.com/lambda/home?region=ap-southeast-1#/functions/SecondaryRegionWriter?subtab=permissions&tab=testing
Search
[Alt+S]
Asia Pacific (Singapore)
ISHA
Lambda > Functions > SecondaryRegionWriter
The test event "TestPermissions" was successfully saved.
Info
Tutorials
Learn how to implement common use cases in AWS Lambda.
Test event Info
Delete
CloudWatch Logs Live Tail
Save
Test
To invoke your function without saving an event, modify the event, then choose Test. Lambda uses the modified event to invoke your function, but does not overwrite the original event until you choose Save.
Test event action
Create new event
Edit saved event
Event name
TestPermissions
Event JSON
Format JSON
{
"key1": "value1",
"key2": "value2",
"key3": "value3"
}
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


ap-southeast-1.console.aws.amazon.com/lambda/home?region=ap-southeast-1#/functions/SecondaryRegionWriter?subtab=permissions&tab=testing
Search
[Alt+S]
Asia Pacific (Singapore)
ISHA
Lambda > Functions > SecondaryRegionWriter
The test event "TestPermissions" was successfully saved.
Info
Tutorials
Code
Test
Monitor
Configuration
Aliases
Versions
Executing function: succeeded (logs)
Details
Learn how to implement common use cases in AWS Lambda.
Test event Info
Delete
CloudWatch Logs Live Tail
Save
Test
To invoke your function without saving an event, modify the event, then choose Test. Lambda uses the modified event to invoke your function, but does not overwrite the original event until you choose Save.
Test event action
Create new event
Edit saved event
Event name
TestPermissions
Event JSON
Format JSON
{
"key1": "value1",
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

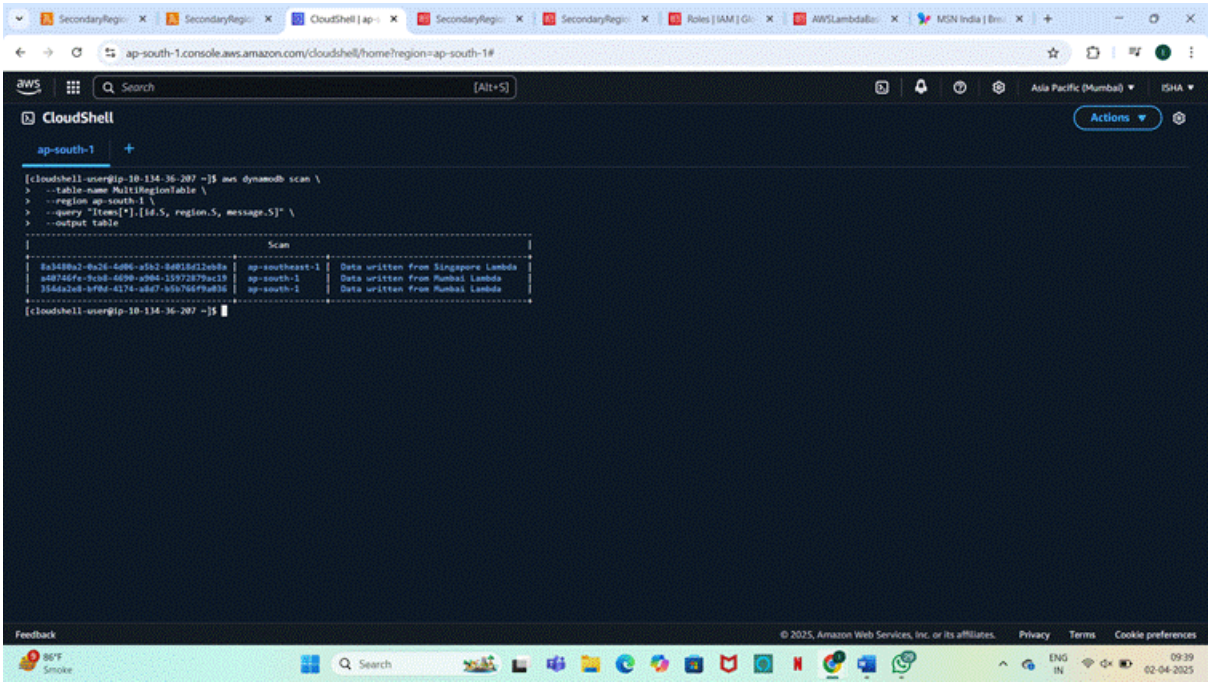

ap-southeast-1.console.aws.amazon.com/lambda/home?region=ap-southeast-1#/functions/SecondaryRegionWriter?subtab=permissions&tab=testing
Lambda > Functions > SecondaryRegionWriter
The test event "TestPermissions" was successfully saved.
Executing function: succeeded (logs)
Details
{
"statusCode": 200,
"body": "Success! Inserted item 8a3480a2-0a26-4d06-a5b2-8d018d12eb8a into DynamoDB (Singapore)"
}
Summary
Code SHA-256
O9zjFeeZeEx/LS5VKwFMXIQbMpcyNc/ON7CyTJPZT0U=
Execution time
33 seconds ago
Function version
$LATEST
Request ID
adc17a3c-de0c-4607-b334-a82e0c8f57a3
Duration
2374.42 ms
Billed duration
2375 ms
Resources configured
128 MB
Max memory used
80 MB
Init duration
287.19 ms
Log output
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

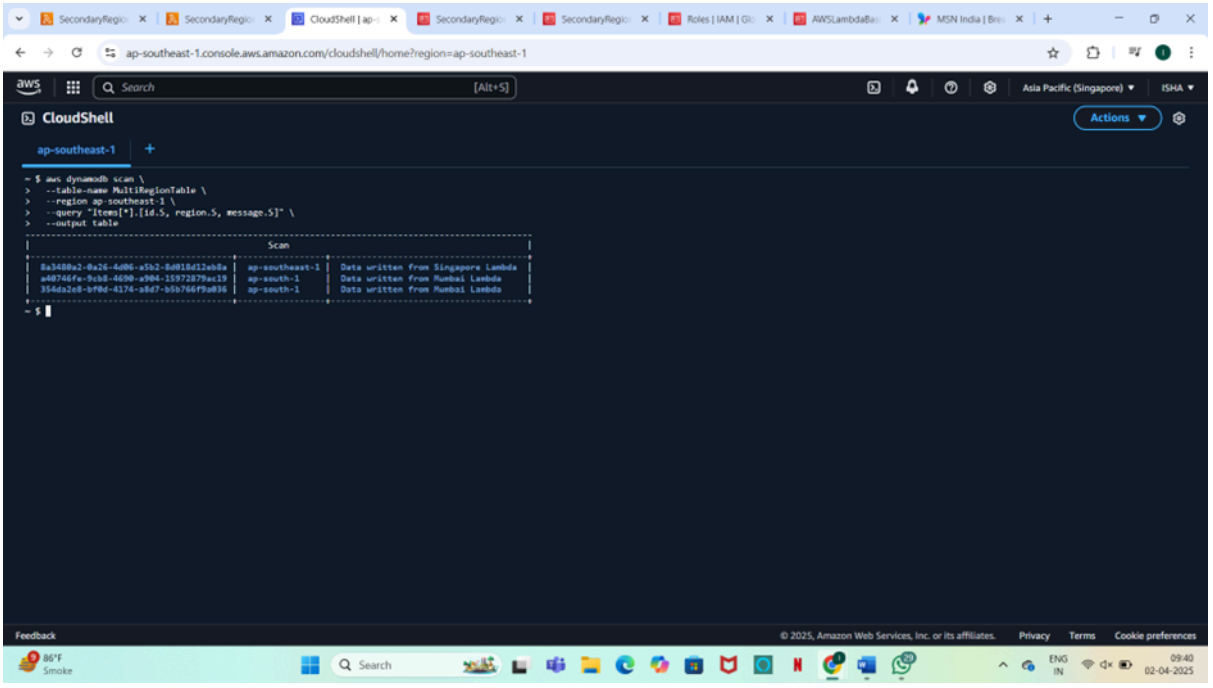

ap-southeast-1.console.aws.amazon.com/lambda/home?region=ap-southeast-1#/functions/SecondaryRegionWriter?subtab=permissions&tab=monitoring
Lambda > Functions > SecondaryRegionWriter
The test event "TestPermissions" was successfully saved.
Code
Test
Monitor
Configuration
Aliases
Versions
Monitor
Info
View CloudWatch logs
View Application Signals
View X-Ray traces
View Lambda Insights
View CodeGuru profiles
Filter metrics by
Function
Alarm recommendations
3h
1d
1w
UTC timezone
CloudWatch metrics
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.
Invocations
Duration
Error count and success rate (%)
Count
Milliseconds
No unit
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
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


CloudWatch > Log groups > /aws/lambda/SecondaryRegionWriter
CloudWatch
Favorites and recents
Dashboards New
Alarms
Logs
Log groups New
Log Anomalies
Live Tail
Logs Insights New
Contributor Insights
Metrics
X-Ray traces New
Events
Application Signals New
Network Monitoring New
Insights
ARN
arn:aws:logs:ap-southeast-1:977099003849:log-group:/aws/lambda/SecondaryRegionWriter:*
Creation time
2 minutes ago
Retention
Never expire
Stored bytes
Subscription filters
0
Contributor Insights rules
KMS key ID
Anomaly detection
Configure
Sensitive data count
Field indexes
Configure
Transformer
Configure
Log streams
Tags
Anomaly detection
Metric filters
Subscription filters
Contributor Insights
Data protection
Field
Log streams (1)
Delete
Create log stream
Search all log streams
Filter log streams or try prefix search
Exact match
Show expired
Info
Log stream
Last event time
2025/04/02/[$LATEST]d11370896df64e2694dff884a34404a9
2025-04-02 04:01:38 (UTC)
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


CloudWatch > Log groups > /aws/lambda/SecondaryRegionWriter > 2025/04/02/[$LATEST]d11370896df64e2694dff884a34404a9
CloudWatch
Log events
Actions
Start tailing
Create metric filter
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
Filter events - press enter to search
Clear
1m
30m
1h
12h
Custom
UTC timezone
Display
Timestamp
Message
No older events at this moment. Retry
INIT_START Runtime Version: python:3.13.v31 Runtime Version ARN: arn:aws:lambda:ap-southeast-1::runtime:f713ac0afb982fcdf9bac88eaa0...
START RequestId: adc17a3c-de0c-4607-b334-a82e0c8f57a3 Version: $LATEST
END RequestId: adc17a3c-de0c-4607-b334-a82e0c8f57a3
REPORT RequestId: adc17a3c-de0c-4607-b334-a82e0c8f57a3 Duration: 2374.42 ms Billed Duration: 2375 ms Memory Size: 128 MB Max Memory...
No newer events at this moment. Auto retry paused. Resume
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

Step 3: Configure Route 53 Health Checks



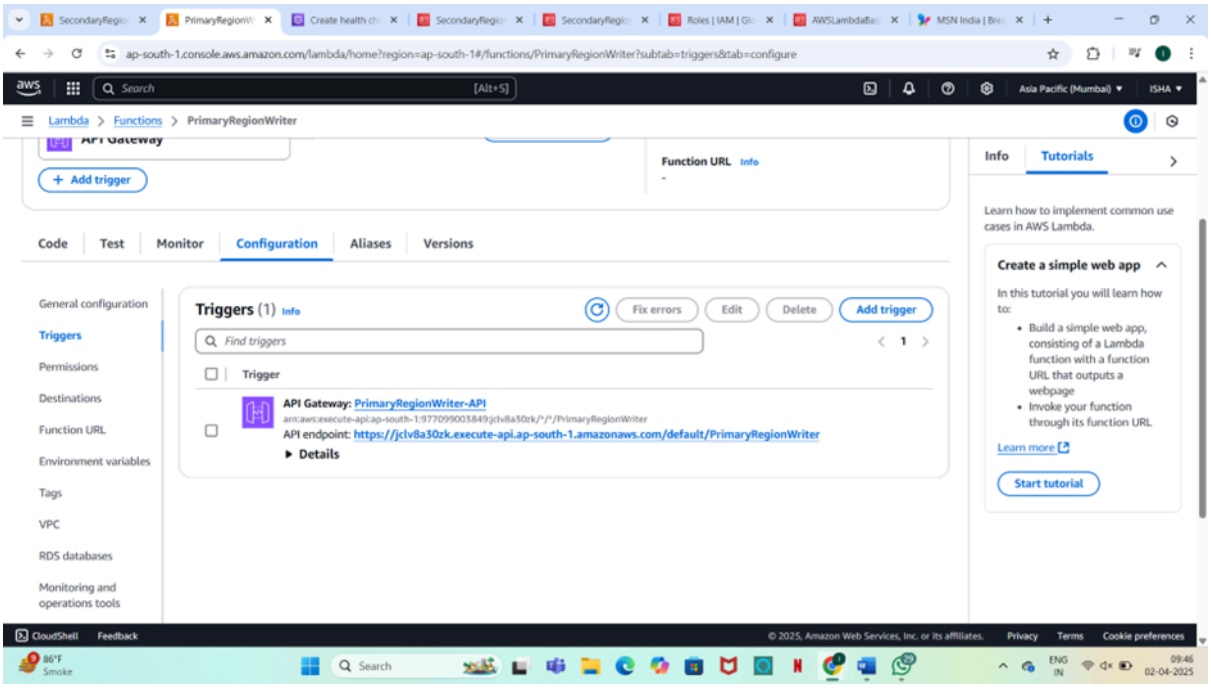


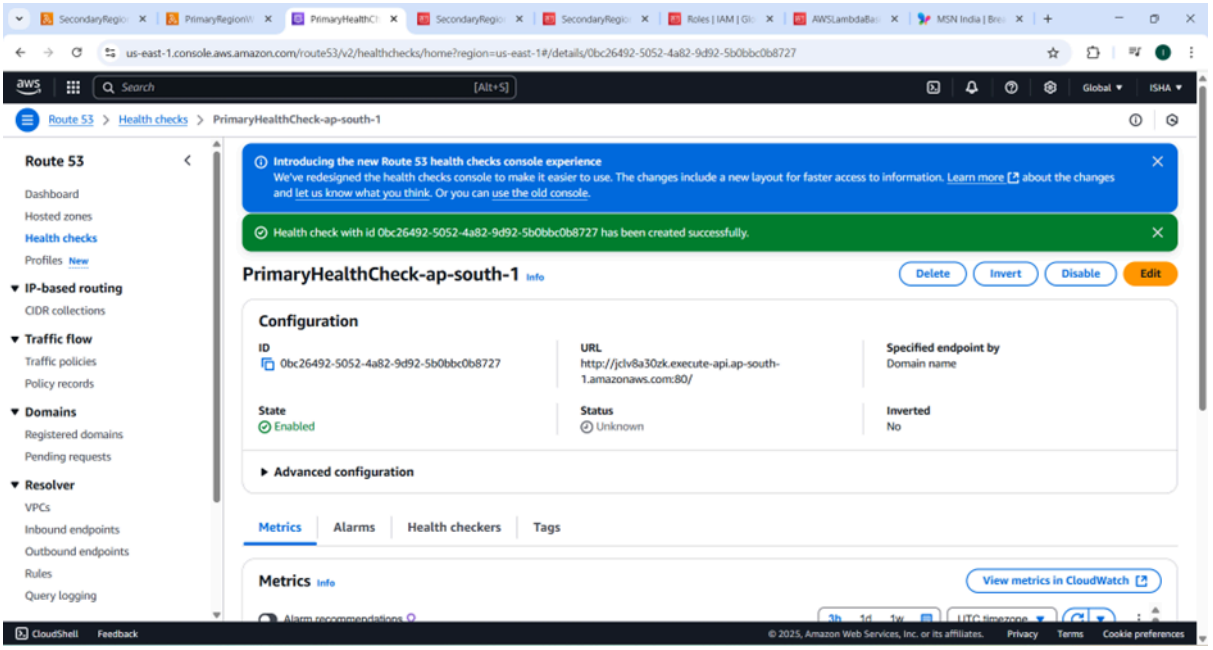
1. **Create Health Check for Mumbai:**

   Health checks monitor the status of our APIs in different AWS regions (Mumbai & Singapore).

They help Route 53 automatically switch traffic to a backup region if the primary region fails.

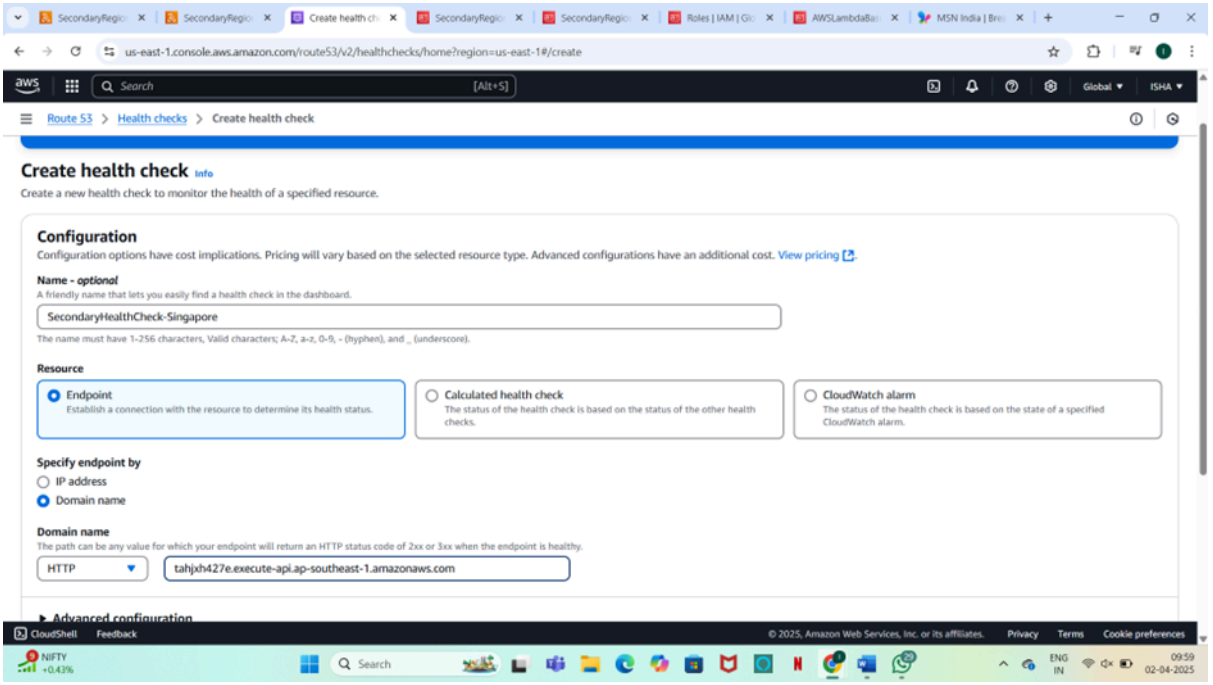
- **Name: PrimaryHealthCheck-ap-south-1**
- **Domain: [api-id].execute-api.ap-south-1.amazonaws.com**
- **Path: /default/PrimaryRegionWriter**
- **Protocol: HTTPS**

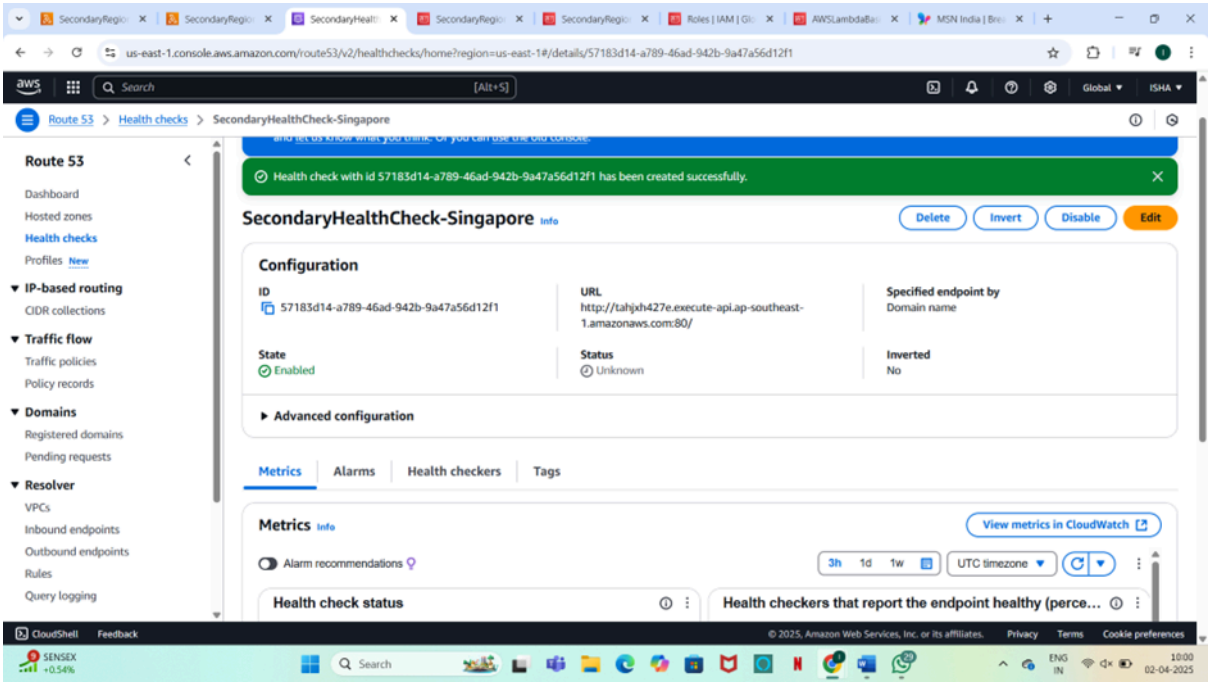




1. **Create Health Check for Singapore:**

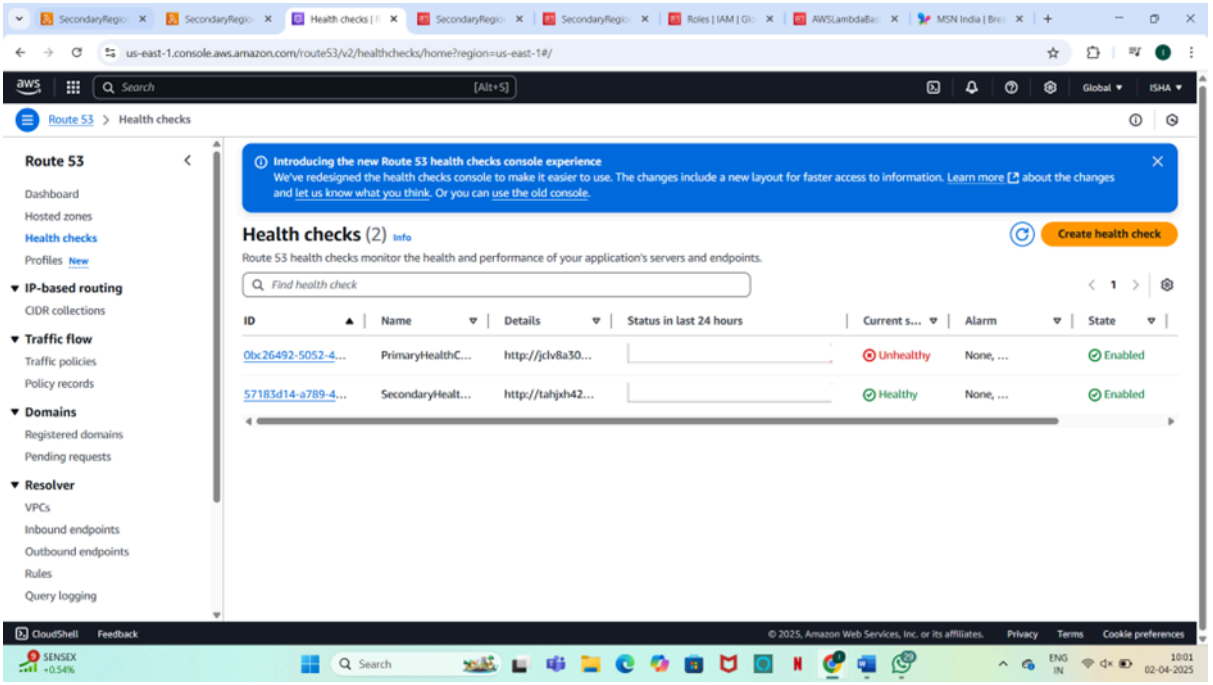
- **Name: SecondaryHealthCheck-ap-southeast-1**
- **Domain: [api-id].execute-api.ap-southeast-1.amazonaws.com**
- **Path: /default/SecondaryRegionWriter**





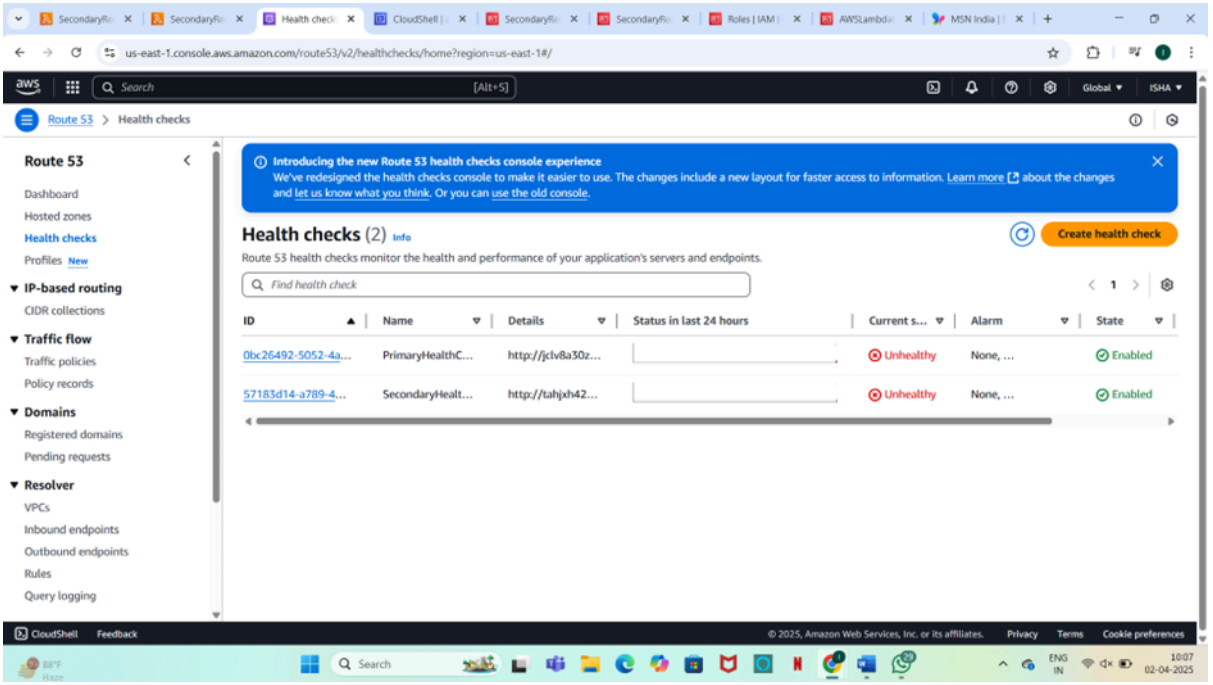
Verify Health Checks:

Wait ~5 mins → Check status in Route 53.



Step 4: Set Up Route 53 Failover

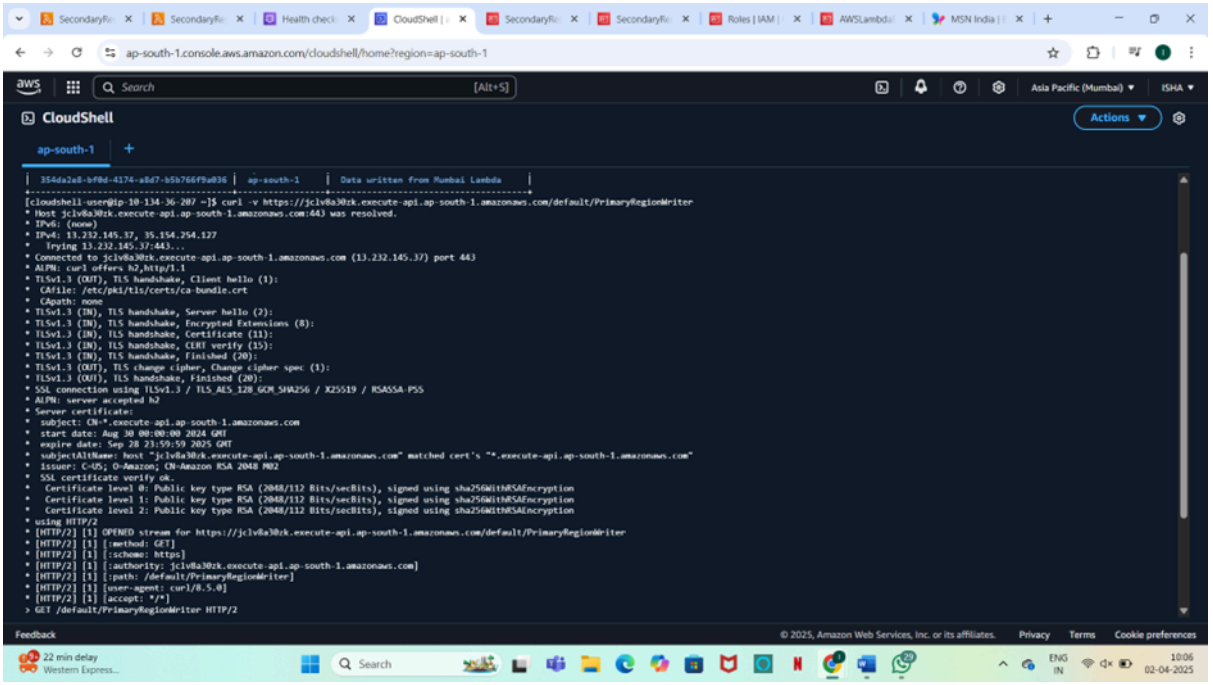
1. **Go to Route 53 → Hosted Zones → Select your domain.**
2. **Create A Record:**
    - **Name: api.yourdomain.com**
    - **Type: A → Alias to API Gateway (Mumbai).**
    - **Routing Policy: Failover**
    - **Primary: Linked to Mumbai health check.**
    - **Secondary: Linked to Singapore health check.**

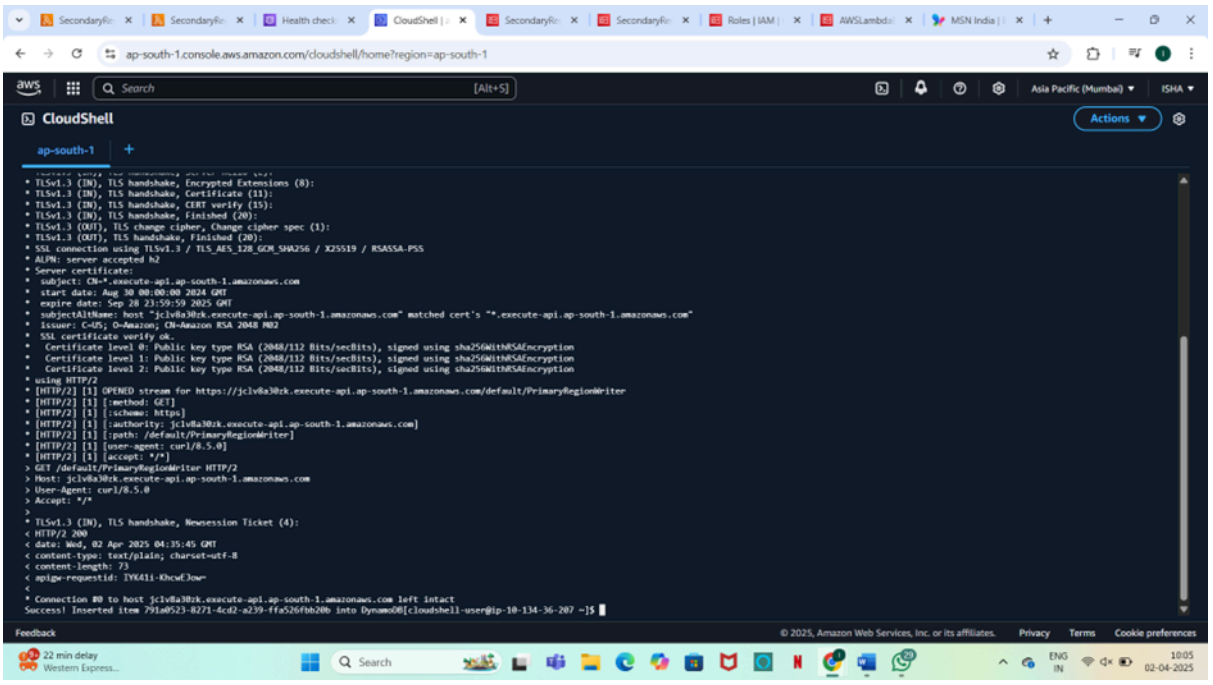


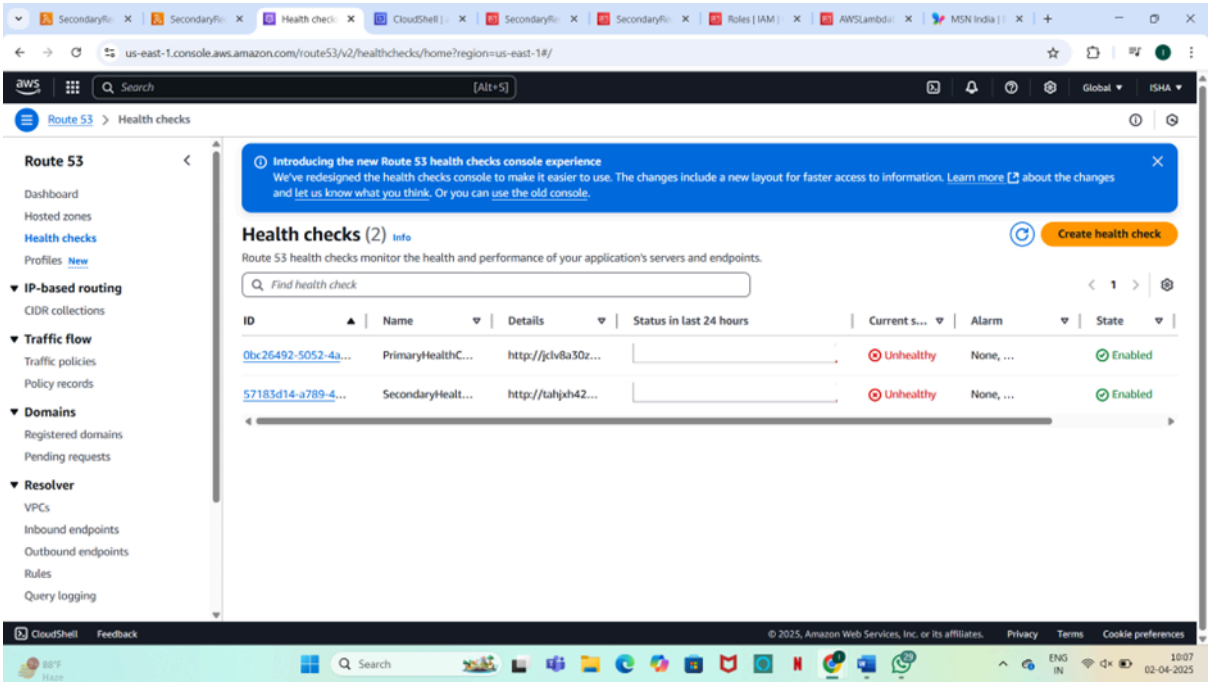
Step 5: Test Failover

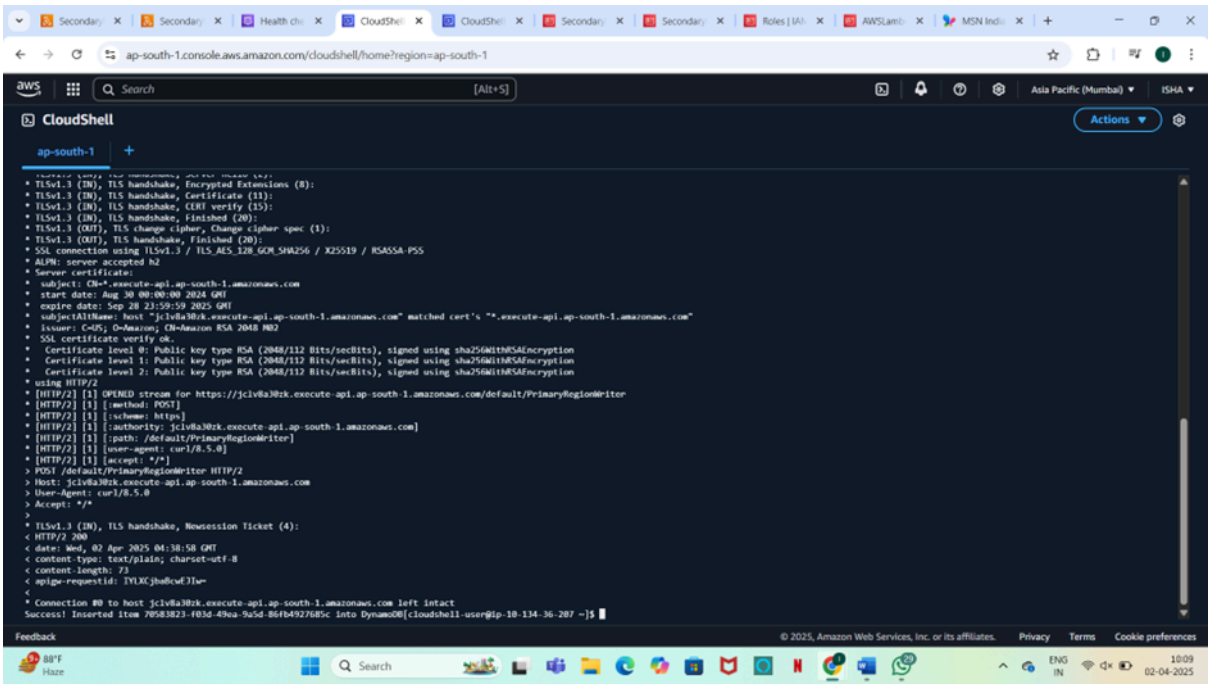
1. **Simulate Mumbai Failure:**
    - **Throttle Mumbai Lambda (Reserved Concurrency = 0).**
2. **Observe:**
    - **Route 53 shifts traffic to Singapore within ~2 mins.**
3. **Restore Mumbai:**
    - **Set concurrency back to Unreserved.**

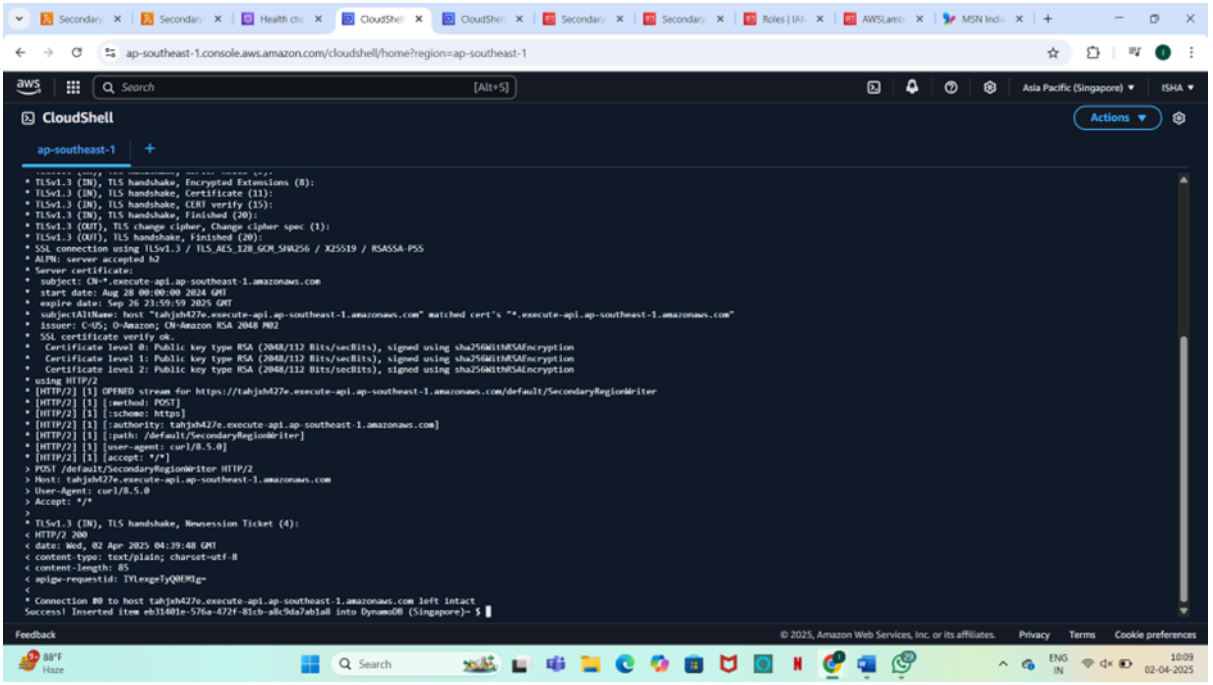
4. **Troubleshooting**
    - **Health Check "Unhealthy"**
    - **No Data Replication**
    - **API Gateway 403/5XX**


ap-south-1.console.aws.amazon.com/cloudshell/home?region=ap-south-1
CloudShell
Actions
ap-south-1
[cloudshell-user@ip-10-134-36-207 ~]$ curl -v https://jclv8a30zk.execute-api.ap-south-1.amazonaws.com/default/PrimaryRegionWriter
* Host jclv8a30zk.execute-api.ap-south-1.amazonaws.com:443 was resolved.
* IPv6: (none)
* IPv4: 13.232.145.37, 35.154.254.127
* Trying 13.232.145.37:443...
* Connected to jclv8a30zk.execute-api.ap-south-1.amazonaws.com (13.232.145.37) port 443
* ALPN: curl offers h2,http/1.1
* TLSv1.3 (OUT), TLS handshake, Client hello (1):
* CAfile: /etc/pki/tls/certs/ca-bundle.crt
* CApath: none
* TLSv1.3 (IN), TLS handshake, Server hello (2):
* TLSv1.3 (IN), TLS handshake, Encrypted Extensions (8):
* TLSv1.3 (IN), TLS handshake, Certificate (11):
* TLSv1.3 (IN), TLS handshake, CERT verify (15):
* TLSv1.3 (IN), TLS handshake, Finished (20):
* TLSv1.3 (OUT), TLS change cipher, Change cipher spec (1):
* TLSv1.3 (OUT), TLS handshake, Finished (20):
* SSL connection using TLSv1.3 / TLS_AES_128_GCM_SHA256 / X25519 / RSASSA-PSS
* ALPN: server accepted h2
* Server certificate:
* subject: CN=*.execute-api.ap-south-1.amazonaws.com
* start date: Aug 30 00:00:00 2024 GMT
* expire date: Sep 28 23:59:59 2025 GMT
* subjectAltName: host "jclv8a30zk.execute-api.ap-south-1.amazonaws.com" matched cert's "*.execute-api.ap-south-1.amazonaws.com"
* issuer: C=US; O=Amazon; CN=Amazon RSA 2048 M02
* SSL certificate verify ok.
* Certificate level 0: Public key type RSA (2048/112 Bits/secBits), signed using sha256WithRSAEncryption
* Certificate level 1: Public key type RSA (2048/112 Bits/secBits), signed using sha256WithRSAEncryption
* Certificate level 2: Public key type RSA (2048/112 Bits/secBits), signed using sha256WithRSAEncryption
* using HTTP/2
* [HTTP/2] [1] OPENED stream for https://jclv8a30zk.execute-api.ap-south-1.amazonaws.com/default/PrimaryRegionWriter
* [HTTP/2] [1] [:method: GET]
* [HTTP/2] [1] [:scheme: https]
* [HTTP/2] [1] [:authority: jclv8a30zk.execute-api.ap-south-1.amazonaws.com]
* [HTTP/2] [1] [:path: /default/PrimaryRegionWriter]
* [HTTP/2] [1] [user-agent: curl/8.5.0]
* [HTTP/2] [1] [accept: */*]
> GET /default/PrimaryRegionWriter HTTP/2
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


ap-south-1.console.aws.amazon.com/cloudshell/home?region=ap-south-1
CloudShell
Actions
ap-south-1
* TLSv1.3 (IN), TLS handshake, Encrypted Extensions (8):
* TLSv1.3 (IN), TLS handshake, Certificate (11):
* TLSv1.3 (IN), TLS handshake, CERT verify (15):
* TLSv1.3 (IN), TLS handshake, Finished (20):
* TLSv1.3 (OUT), TLS change cipher, Change cipher spec (1):
* TLSv1.3 (OUT), TLS handshake, Finished (20):
* SSL connection using TLSv1.3 / TLS_AES_128_GCM_SHA256 / X25519 / RSASSA-PSS
* ALPN: server accepted h2
* Server certificate:
* subject: CN=*.execute-api.ap-south-1.amazonaws.com
* start date: Aug 30 00:00:00 2024 GMT
* expire date: Sep 28 23:59:59 2025 GMT
* subjectAltName: host "jclv8a30zk.execute-api.ap-south-1.amazonaws.com" matched cert's "*.execute-api.ap-south-1.amazonaws.com"
* issuer: C=US; O=Amazon; CN=Amazon RSA 2048 M02
* SSL certificate verify ok.
* Certificate level 0: Public key type RSA (2048/112 Bits/secBits), signed using sha256WithRSAEncryption
* Certificate level 1: Public key type RSA (2048/112 Bits/secBits), signed using sha256WithRSAEncryption
* Certificate level 2: Public key type RSA (2048/112 Bits/secBits), signed using sha256WithRSAEncryption
* using HTTP/2
* [HTTP/2] [1] OPENED stream for https://jclv8a30zk.execute-api.ap-south-1.amazonaws.com/default/PrimaryRegionWriter
* [HTTP/2] [1] [:method: GET]
* [HTTP/2] [1] [:scheme: https]
* [HTTP/2] [1] [:authority: jclv8a30zk.execute-api.ap-south-1.amazonaws.com]
* [HTTP/2] [1] [:path: /default/PrimaryRegionWriter]
* [HTTP/2] [1] [user-agent: curl/8.5.0]
* [HTTP/2] [1] [accept: */*]
> GET /default/PrimaryRegionWriter HTTP/2
> Host: jclv8a30zk.execute-api.ap-south-1.amazonaws.com
> User-Agent: curl/8.5.0
> Accept: */*
>
* TLSv1.3 (IN), TLS handshake, Newsession Ticket (4):
< HTTP/2 200
< date: Wed, 02 Apr 2025 04:35:45 GMT
< content-type: text/plain; charset=utf-8
< content-length: 73
< apigw-requestid: IYK41i-KhcwEJow=
<
* Connection #0 to host jclv8a30zk.execute-api.ap-south-1.amazonaws.com left intact
Success! Inserted item 791a0523-8271-4cd2-a239-ffa526fbb20b into Dynamo0B[cloudshell-user@ip-10-134-36-207 ~]$
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


us-east-1.console.aws.amazon.com/route53/v2/healthchecks/home?region=us-east-1#/
Route 53 > Health checks
Route 53
Dashboard
Hosted zones
Health checks
Profiles New
IP-based routing
CIDR collections
Traffic flow
Traffic policies
Policy records
Domains
Registered domains
Pending requests
Resolver
VPCs
Inbound endpoints
Outbound endpoints
Rules
Query logging
Introducing the new Route 53 health checks console experience
We've redesigned the health checks console to make it easier to use. The changes include a new layout for faster access to information. Learn more about the changes and let us know what you think. Or you can use the old console.
Health checks (2) Info
Route 53 health checks monitor the health and performance of your application's servers and endpoints.
Create health check
Find health check
ID
Name
Details
Status in last 24 hours
Current s...
Alarm
State
0bc26492-5052-4a...
PrimaryHealthC...
http://jclv8a30z...
Unhealthy
None, ...
Enabled
57183d14-a789-4...
SecondaryHealt...
http://tahjxh42...
Unhealthy
None, ...
Enabled
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


ap-south-1.console.aws.amazon.com/cloudshell/home?region=ap-south-1
CloudShell
Actions
ap-south-1
* TLSv1.3 (IN), TLS handshake, Encrypted Extensions (8):
* TLSv1.3 (IN), TLS handshake, Certificate (11):
* TLSv1.3 (IN), TLS handshake, CERT verify (15):
* TLSv1.3 (IN), TLS handshake, Finished (20):
* TLSv1.3 (OUT), TLS change cipher, Change cipher spec (1):
* TLSv1.3 (OUT), TLS handshake, Finished (20):
* SSL connection using TLSv1.3 / TLS_AES_128_GCM_SHA256 / X25519 / RSASSA-PSS
* ALPN: server accepted h2
* Server certificate:
* subject: CN=*.execute-api.ap-south-1.amazonaws.com
* start date: Aug 30 00:00:00 2024 GMT
* expire date: Sep 28 23:59:59 2025 GMT
* subjectAltName: host "jclv8a30zk.execute-api.ap-south-1.amazonaws.com" matched cert's "*.execute-api.ap-south-1.amazonaws.com"
* issuer: C=US; O=Amazon; CN=Amazon RSA 2048 M02
* SSL certificate verify ok.
* Certificate level 0: Public key type RSA (2048/112 Bits/secBits), signed using sha256WithRSAEncryption
* Certificate level 1: Public key type RSA (2048/112 Bits/secBits), signed using sha256WithRSAEncryption
* Certificate level 2: Public key type RSA (2048/112 Bits/secBits), signed using sha256WithRSAEncryption
* using HTTP/2
* [HTTP/2] [1] OPENED stream for https://jclv8a30zk.execute-api.ap-south-1.amazonaws.com/default/PrimaryRegionWriter
* [HTTP/2] [1] [:method: POST]
* [HTTP/2] [1] [:scheme: https]
* [HTTP/2] [1] [:authority: jclv8a30zk.execute-api.ap-south-1.amazonaws.com]
* [HTTP/2] [1] [:path: /default/PrimaryRegionWriter]
* [HTTP/2] [1] [user-agent: curl/8.5.0]
* [HTTP/2] [1] [accept: */*]
> POST /default/PrimaryRegionWriter HTTP/2
> Host: jclv8a30zk.execute-api.ap-south-1.amazonaws.com
> User-Agent: curl/8.5.0
> Accept: */*
>
* TLSv1.3 (IN), TLS handshake, Newsession Ticket (4):
< HTTP/2 200
< date: Wed, 02 Apr 2025 04:38:58 GMT
< content-type: text/plain; charset=utf-8
< content-length: 73
< apigw-requestid: IYLXCjba8cwEJIw=
<
* Connection #0 to host jclv8a30zk.execute-api.ap-south-1.amazonaws.com left intact
Success! Inserted item 70583823-f03d-49ea-9a5d-86fb4927685c into Dynamo0B[cloudshell-user@ip-10-134-36-207 ~]$
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


ap-southeast-1.console.aws.amazon.com/cloudshell/home?region=ap-southeast-1
Search
[Alt+S]
Asia Pacific (Singapore)
ISHA
CloudShell
Actions
ap-southeast-1
* TLSv1.3 (IN), TLS handshake, Encrypted Extensions (8):
* TLSv1.3 (IN), TLS handshake, Certificate (11):
* TLSv1.3 (IN), TLS handshake, CERT verify (15):
* TLSv1.3 (IN), TLS handshake, Finished (20):
* TLSv1.3 (OUT), TLS change cipher, Change cipher spec (1):
* TLSv1.3 (OUT), TLS handshake, Finished (20):
* SSL connection using TLSv1.3 / TLS_AES_128_GCM_SHA256 / X25519 / RSASSA-PSS
* ALPN: server accepted h2
* Server certificate:
* subject: CN=*.execute-api.ap-southeast-1.amazonaws.com
* start date: Aug 28 00:00:00 2024 GMT
* expire date: Sep 26 23:59:59 2025 GMT
* subjectAltName: host "tahjxh427e.execute-api.ap-southeast-1.amazonaws.com" matched cert's "*.execute-api.ap-southeast-1.amazonaws.com"
* issuer: C=US; O=Amazon; CN=Amazon RSA 2048 M02
* SSL certificate verify ok.
* Certificate level 0: Public key type RSA (2048/112 Bits/secBits), signed using sha256WithRSAEncryption
* Certificate level 1: Public key type RSA (2048/112 Bits/secBits), signed using sha256WithRSAEncryption
* Certificate level 2: Public key type RSA (2048/112 Bits/secBits), signed using sha256WithRSAEncryption
* using HTTP/2
* [HTTP/2] [1] OPENED stream for https://tahjxh427e.execute-api.ap-southeast-1.amazonaws.com/default/SecondaryRegionWriter
* [HTTP/2] [1] [:method: POST]
* [HTTP/2] [1] [:scheme: https]
* [HTTP/2] [1] [:authority: tahjxh427e.execute-api.ap-southeast-1.amazonaws.com]
* [HTTP/2] [1] [:path: /default/SecondaryRegionWriter]
* [HTTP/2] [1] [user-agent: curl/8.5.0]
* [HTTP/2] [1] [accept: */*]
> POST /default/SecondaryRegionWriter HTTP/2
> Host: tahjxh427e.execute-api.ap-southeast-1.amazonaws.com
> User-Agent: curl/8.5.0
> Accept: */*
>
* TLSv1.3 (IN), TLS handshake, Newsession Ticket (4):
< HTTP/2 200
< date: Wed, 02 Apr 2025 04:39:48 GMT
< content-type: text/plain; charset=utf-8
< content-length: 85
< apigw-requestid: IYLexgeIyQ0EMIg=
<
* Connection #0 to host tahjxh427e.execute-api.ap-southeast-1.amazonaws.com left intact
Success! Inserted item eb31401e-576a-472f-81cb-a8c9da7ab1a8 into DynamoDB (Singapore)~ $
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
88°F
Haze
Search
ENG
IN
10:09
02-04-2025

Security & Compliance Considerations

- **Data Encryption** – Implement AWS KMS for encryption at rest and in transit to protect sensitive information.
- **IAM Role-Based Access Control** – Define strict IAM roles to ensure only authorized users and services access critical data.
- **DDoS Protection** – Use AWS Shield Advanced and AWS WAF to mitigate cyber threats and maintain uptime.
- **Regulatory Compliance** – Ensure adherence to industry standards like GDPR, TRAI, and telecom data privacy laws.

Expected Outcomes

- **99.99% Service Availability** – Thanks to multi-region redundancy and automated failover.
- **Faster Fraud Detection** – AI-driven models reduce financial loss and prevent security breaches.
- **Accurate & Automated Billing** – Eliminates revenue leakage and enhances transparency in customer transactions.
- **Enhanced Customer Satisfaction** – The self-service portal improves user engagement and reduces customer support load.
- **Operational Cost Reduction** – Serverless computing minimizes infrastructure costs while ensuring scalability.

Conclusion

This AWS-powered telecom solution provides a high-performing, cost-efficient, and resilient system that ensures real-time network monitoring, automated billing, and fraud prevention. With a secure and scalable architecture, customers benefit from an interactive self-service portal while ensuring continuous service availability and compliance.