AWS Cost Monitoring & Budget Alerts

Services: AWS Budgets, Cost Explorer, Lambda, SES Goal: Automatically notify when budget thresholds are met.

Skills: Billing alerts, email notifications, tagging strategies.

Automatically monitor AWS cloud spending and send email alerts when the cost crosses defined thresholds (e.g., 50%, 80%, 100% of the budget).

Architecture Overview

1. **AWS Budgets** → Tracks spending vs. budget
2. **AWS CloudWatch Events** → Triggers Lambda when budget crosses threshold
3. **AWS Lambda** → Processes alert and sends email using SES
4. **AWS SES** → Sends email notifications to the user

STEP 1: Enable Billing & Cost Management Tools

Before creating budgets or alerts:

Enable Billing Alerts

1. Go to **AWS Billing Console**
2. Select **Billing Preferences**
3. Enable:
    - ☑ Receive Billing Alerts
    - ☑ Cost Explorer
    - ☑ Budgets Reports

STEP 2: Create a Budget in AWS Budgets

Create a Cost Budget

1. Open **AWS Budgets** from console
2. Click **Create budget**
3. Select **Cost Budget**
4. Choose type:
    - **Recurring monthly** (most common)
    - Enter **amount** (example: ₹1,000 or $10)
5. Set thresholds:
    - 50% (Alert)
    - 80% (Warning)
    - 100% (Critical)

Add Notification

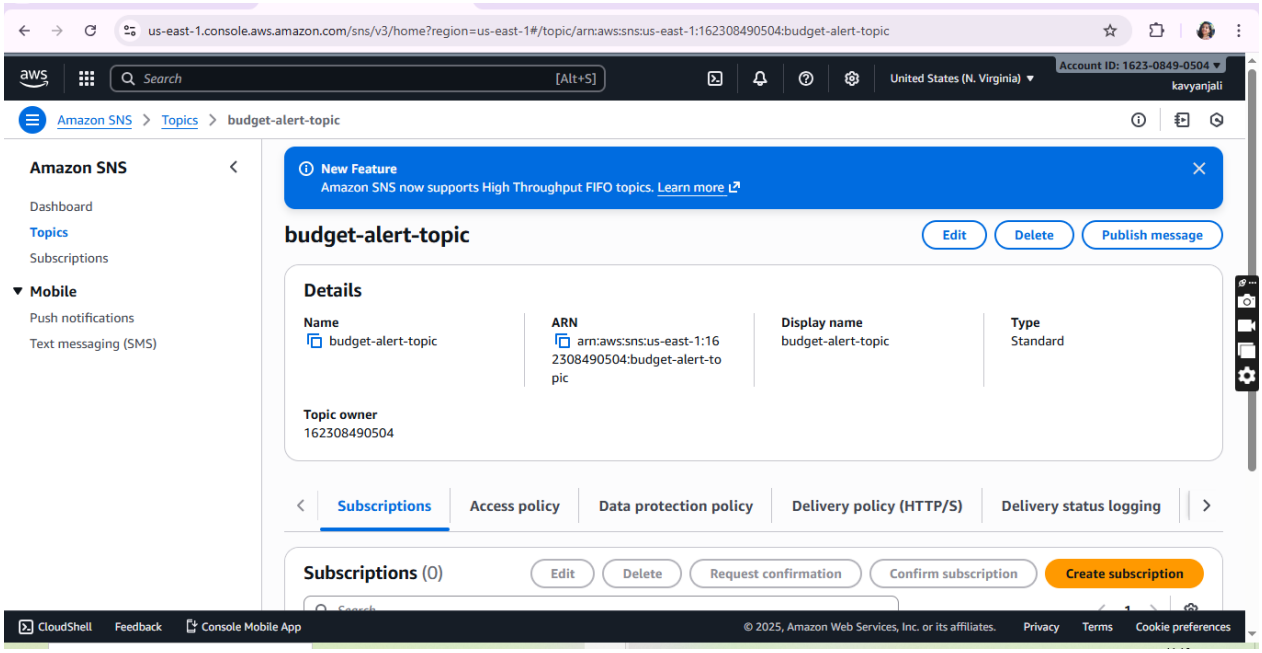
- **Email alerts**
- **SNS topic**
- **Lambda automation**
- **SNS Notification → Lambda trigger**
  (if SNS not used, CloudWatch event triggers Lambda directly)

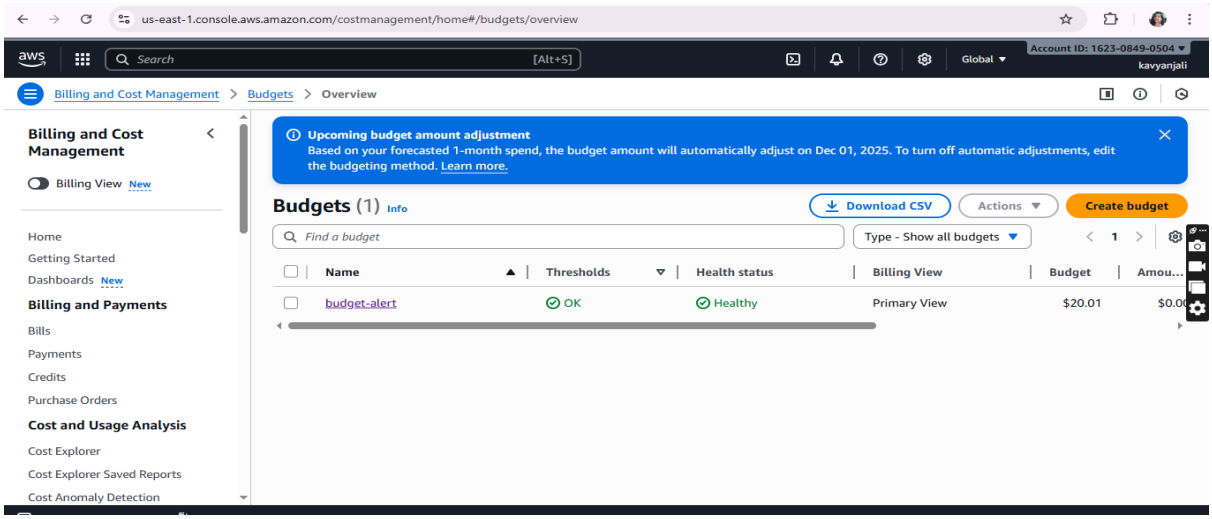
STEP 3: Create an SES Email Identity

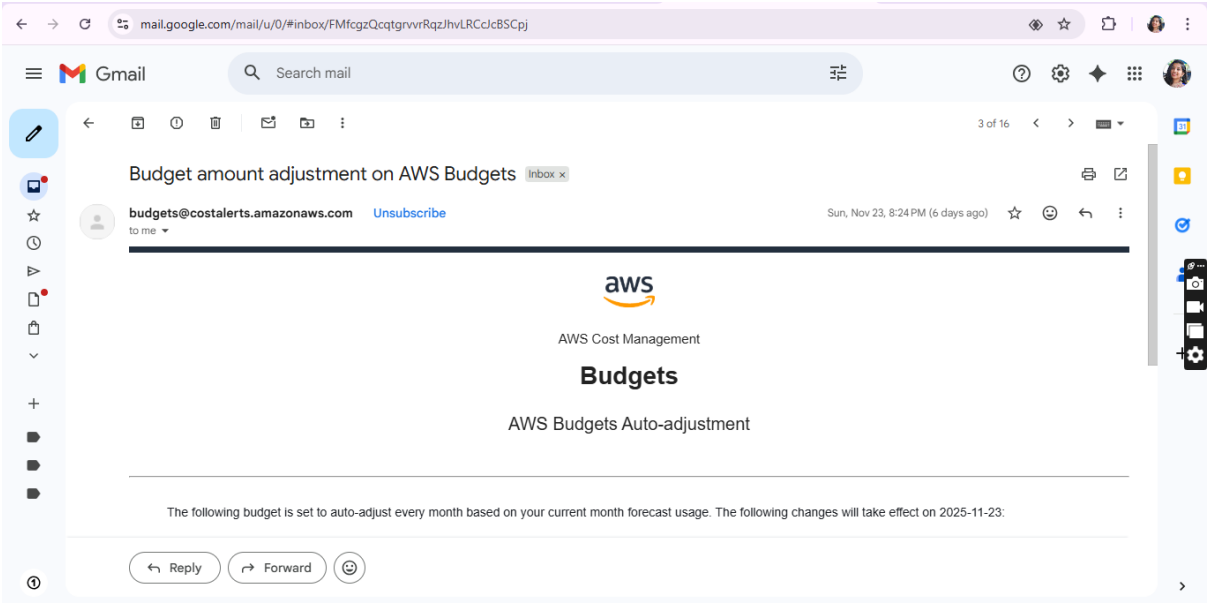
SES is used to send alert emails.

Steps:

1. Go to **Amazon SES**
2. Choose **Verified Identities**
3. Add your email (example: kavyakavyanajali@email@gmail.com)
4. Click **Verify**
5. Check your inbox → click confirm link







STEP 4: Create Alert Lambda Function

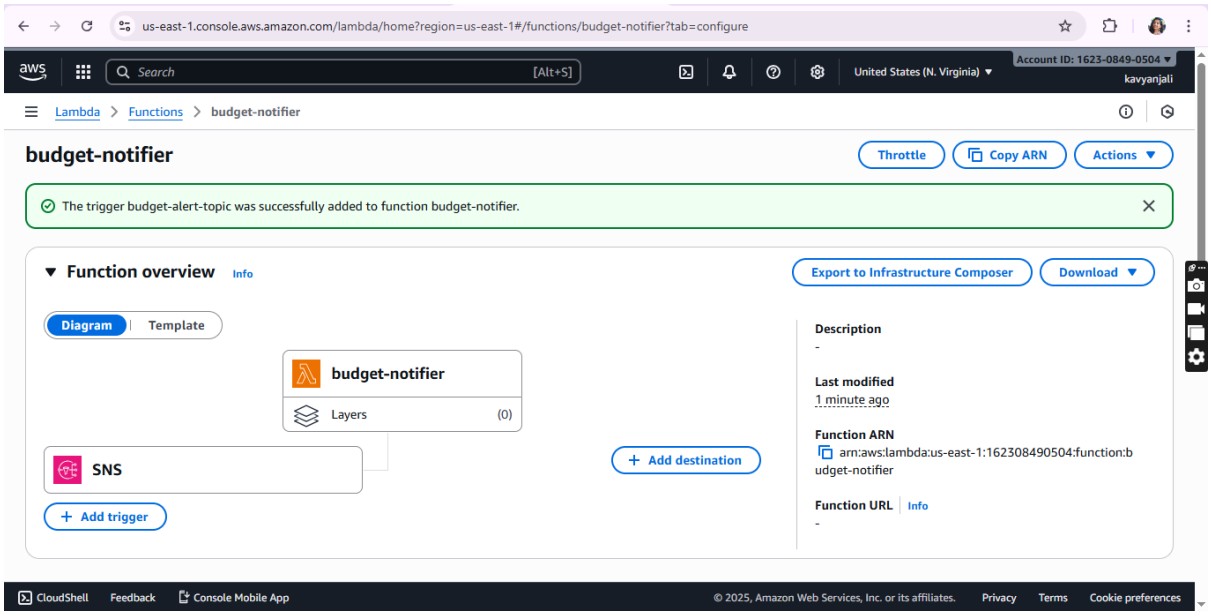
This Lambda sends email through SES when cost crosses threshold.

Python Lambda Code

```
import boto3

ses = boto3.client('ses')
def lambda_handler(event, context):
    budget_name = event['detail']['budgetName']
    cost = event['detail']['newBudgetAmount']
    status = event['detail']['budgetLimit']

    email_body = f"""
    AWS Budget Alert

    Budget Name: {budget_name}
    Current Cost: {cost}
    Threshold Reached: {status}
    """

    ses.send_email(
        Source='your_verified_email@example.com',
        Destination={'ToAddresses': ['recipient@example.com']},
        Message={
            'Subject': {'Data': 'AWS Budget Alert'},
            'Body': {'Text': {'Data': email_body}}
        }
    )

    return "Alert Sent"
```

IAM Permissions

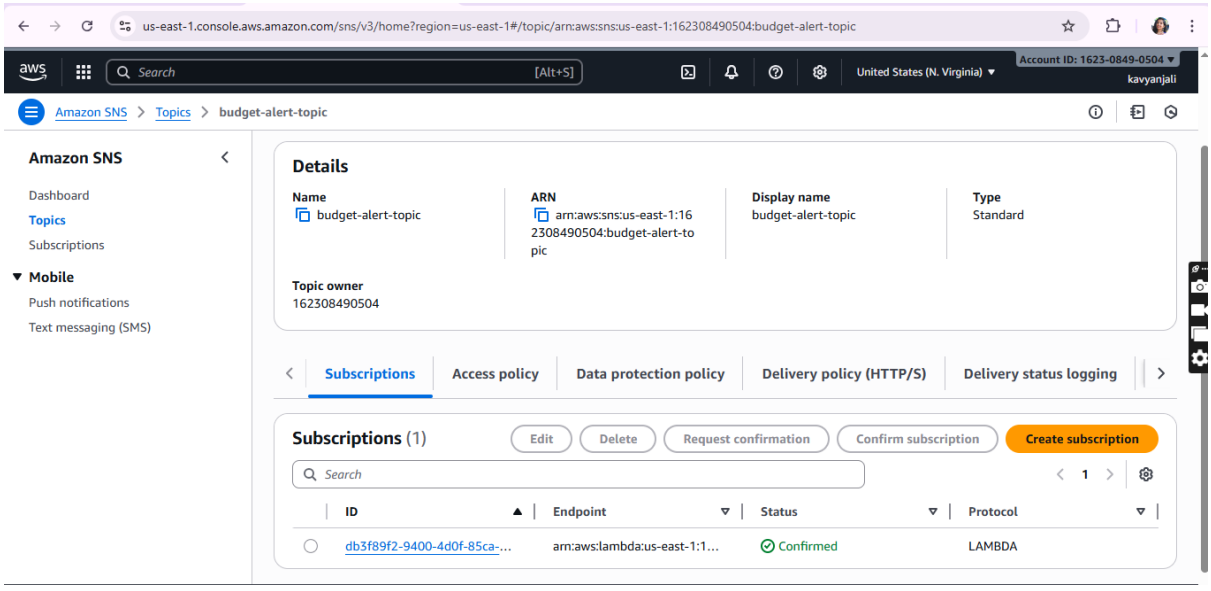
AmazonSESFullAccess,AWSBudgetsActionsRWD,CloudWatchEventsFullAccess



Create the Lambda function Console

steps: 1. Lambda → Create function → Author from scratch → Name: budget-notifier → Runtime: Python 3.11 (or 3.10/3.9) → Execution role: choose the role created earlier.

2. After creation, add an SNS trigger: Add trigger → SNS → choose aws-cost-budgets-topic.



Attach the Lambda function as an SNS subscription

If you added the SNS trigger during function creation it's attached; otherwise:

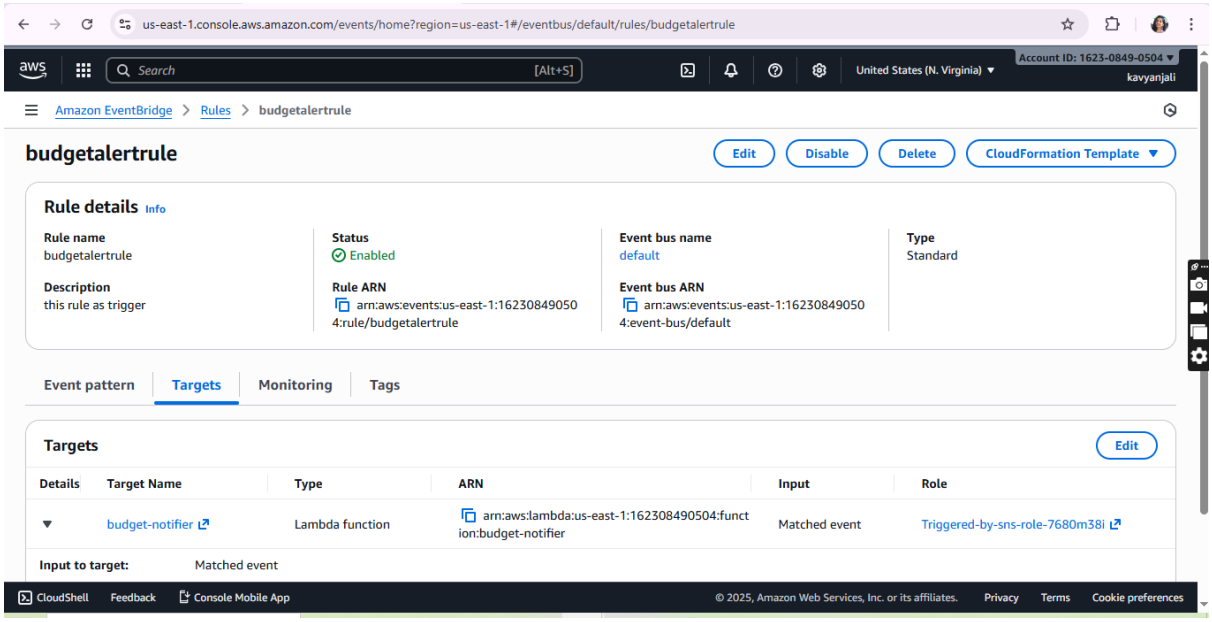
1. SNS → Topics → open aws-cost-budgets-topic → Subscriptions → Create subscription:
2. Protocol: AWS Lambda
3. Endpoint: Choose your Lambda function
4. Confirm the subscription.

STEP 5: Create an EventBridge Rule to Trigger Lambda

A rule is needed so AWS triggers Lambda when budget threshold is crossed.

Steps:

1. Go to **Amazon EventBridge**
2. Click **Create Rule**
3. Name: BudgetAlertRule
4. Event Pattern:
    - Prefill with **AWS Budgets → Threshold Breached**
5. Target:
    - **Lambda Function**



STEP 6: Testing

Test Case

- Lower the budget temporarily (e.g., set ₹10)
- Use AWS for 1 hour—your cost exceeds ₹10 → threshold triggers
- Lambda executes → SES sends an email

Conclusion

This project provides a **robust and automated solution** for AWS cost management using serverless architecture. It enhances visibility, prevents overspending, and supports cost optimization initiatives while requiring almost **zero maintenance**.