Website Uptime Monitor using AWS

Project Overview

This project implements a Website Uptime Monitoring System using AWS managed services. The system continuously monitors the availability of a website and automatically sends email alerts when the website becomes unavailable.

The solution is serverless, cost-effective, and suitable for real-world production monitoring as well as interview demonstrations.

Objective

- Monitor website availability (HTTP/HTTPS)
- Detect downtime automatically
- Trigger alerts within 1 minute of failure
- Send email notifications when downtime occurs

Architecture Overview

Flow

Route 53 Health Check → CloudWatch Metrics → CloudWatch Alarm → Amazon SNS → Email Notification

Explanation:

- Route 53 checks website health from multiple AWS regions
- Health data is sent to CloudWatch as metrics
- CloudWatch Alarm evaluates the metric
- SNS sends email alerts when alarm triggers

AWS Services Used

Service	Purpose
Amazon Route 53	Website health monitoring
Amazon CloudWatch	Metrics & alarm evaluation
Amazon SNS	Email notifications

Step-by-Step Implementation

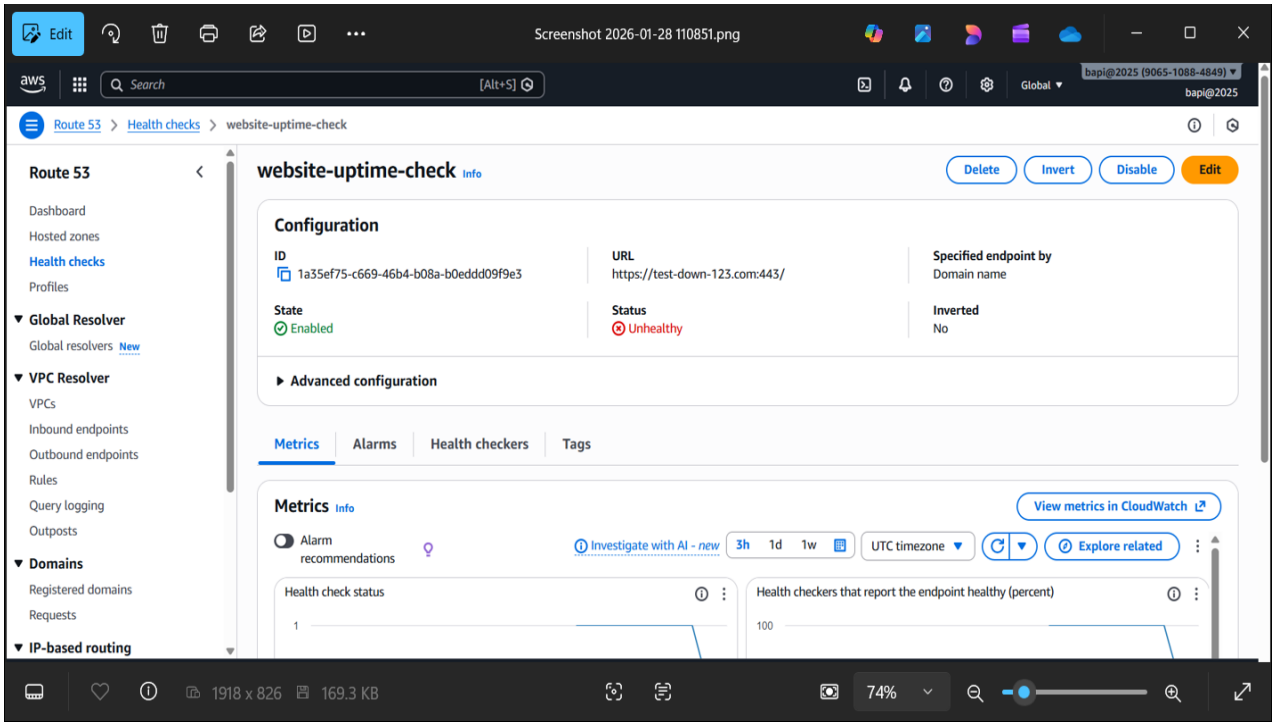
Step 1: Create Route 53 Health Check

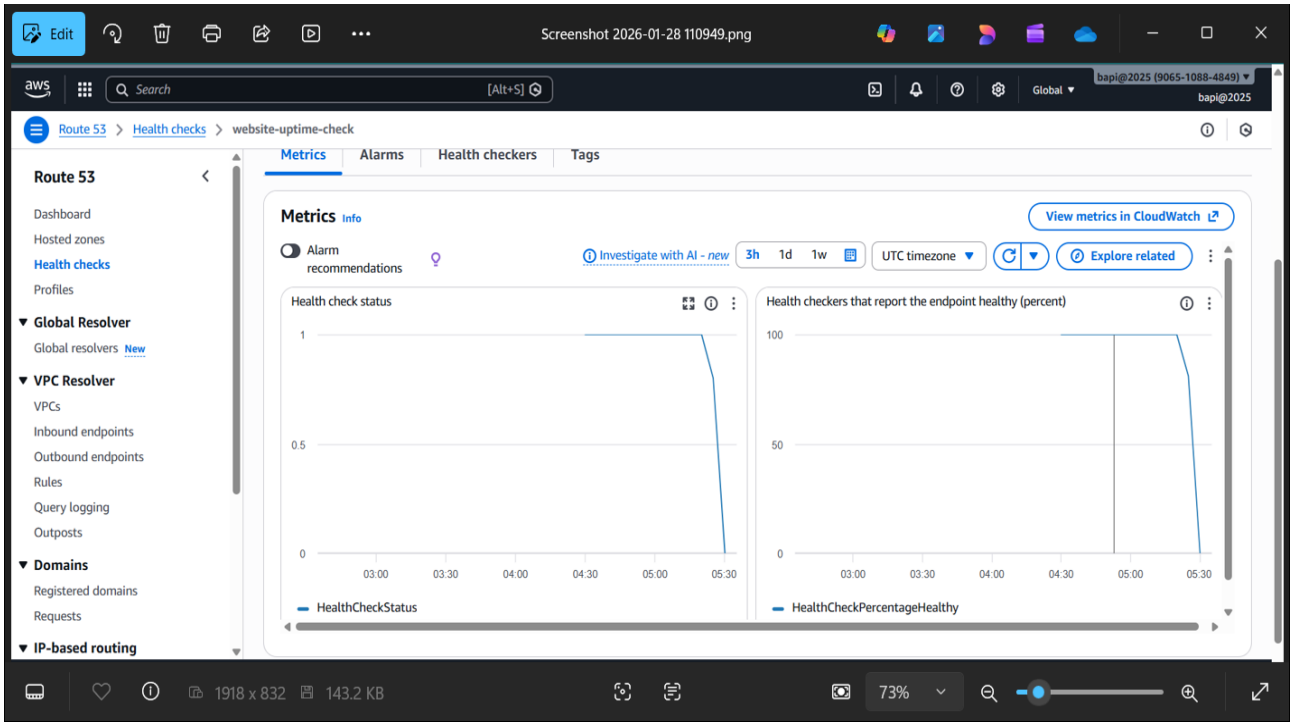
AWS Console Path: Route 53 → Health checks → Create health check

Configuration:

- Name: website-uptime-check
- Resource: Endpoint
- Endpoint type: Domain name
- Protocol: HTTPS
- Domain name: test-down-123.com
- Port: 443
- Path: /
- Request interval: 30 seconds
- Failure threshold: 3
- Health checker regions: Multiple regions selected

This health check continuously monitors the website availability.





Step 2: Verify Health Check Status

After creation, the health check status is visible in Route 53:

- Healthy (1) → Website is UP
- Unhealthy (0) → Website is DOWN

This confirms that AWS is actively monitoring the endpoint.

Step 3: Monitor Health Metrics

Metrics Available:

- HealthCheckStatus (1 = Up, 0 = Down)
- HealthCheckPercentageHealthy

These metrics are automatically published to CloudWatch.

Step 4: Create CloudWatch Alarm

AWS Console Path: CloudWatch → Alarms → Create alarm

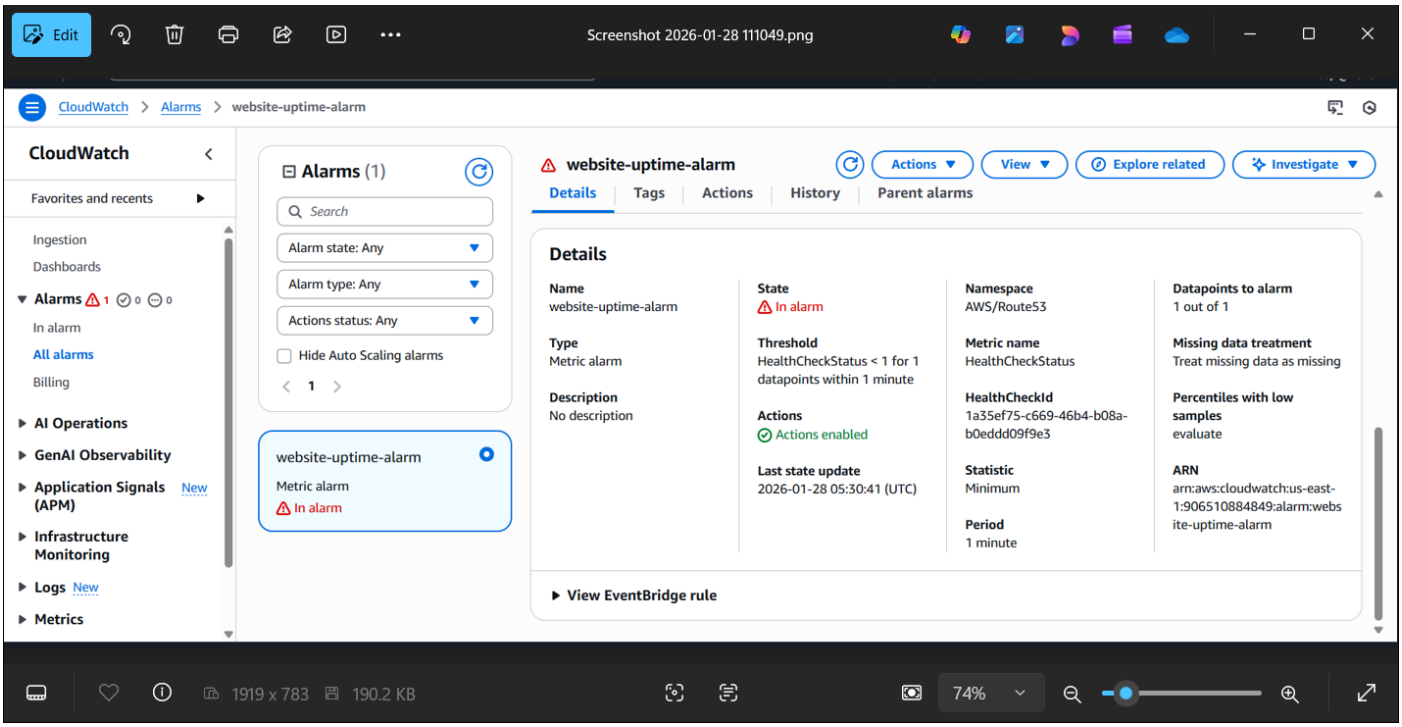
Metric Selection:

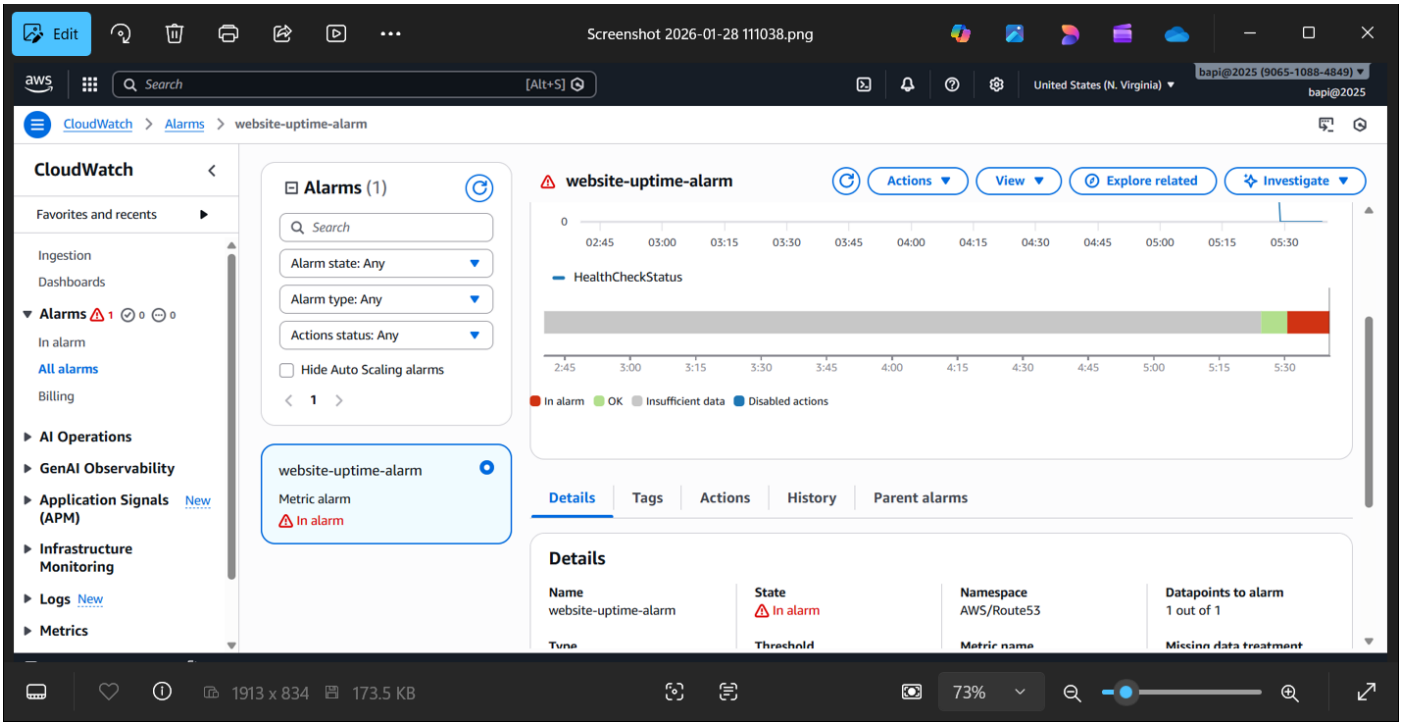
- Namespace: AWS/Route53
- Metric: HealthCheckStatus
- HealthCheckId: Select the created health check

Alarm Condition:

- Threshold type: Static
- Condition: Lower than 1
- Period: 1 minute
- Datapoints: 1 out of 1

Meaning: If the website goes down even once within a 1-minute period, the alarm triggers.



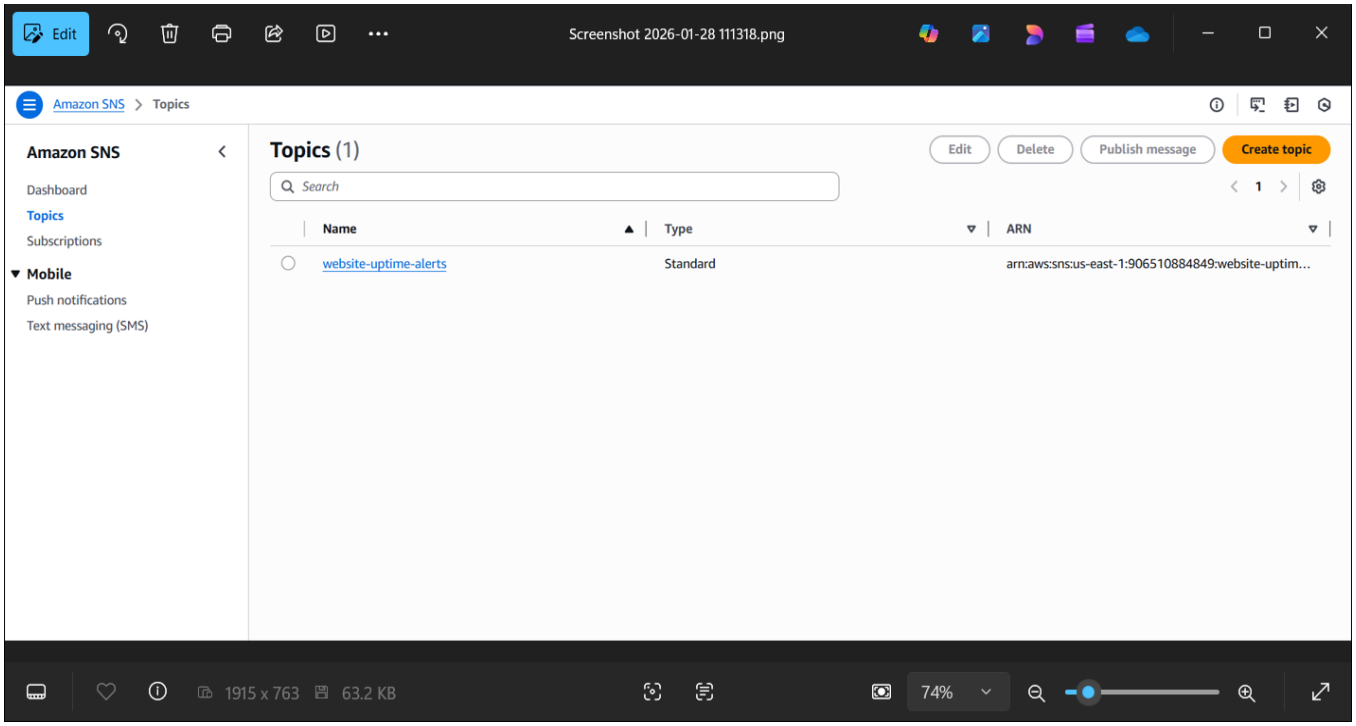

Edit
Screenshot 2026-01-28 111038.png
Search
[Alt+S]
United States (N. Virginia)
bapi@2025 (9065-1088-4849)
bapi@2025
CloudWatch
Alarms
website-uptime-alarm
CloudWatch
Favorites and recents
Ingestion
Dashboards
Alarms 1 0 0
In alarm
All alarms
Billing
AI Operations
GenAI Observability
Application Signals (APM) New
Infrastructure Monitoring
Logs New
Metrics
Alarms (1)
Search
Alarm state: Any
Alarm type: Any
Actions status: Any
Hide Auto Scaling alarms
1
website-uptime-alarm
Metric alarm
In alarm
website-uptime-alarm
Actions
View
Explore related
Investigate
0
02:45
03:00
03:15
03:30
03:45
04:00
04:15
04:30
04:45
05:00
05:15
05:30
HealthCheckStatus
In alarm
OK
Insufficient data
Disabled actions
Details
Tags
Actions
History
Parent alarms
Details
Name
website-uptime-alarm
State
In alarm
Namespace
AWS/Route53
Datapoints to alarm
1 out of 1
Type
Threshold
Metric name
Missing data treatment
1913 x 834
173.5 KB
73%

Step 5: Create SNS Topic

AWS Console Path: Amazon SNS → Topics → Create topic

Configuration:

- Topic name: website-uptime-alerts
- Type: Standard



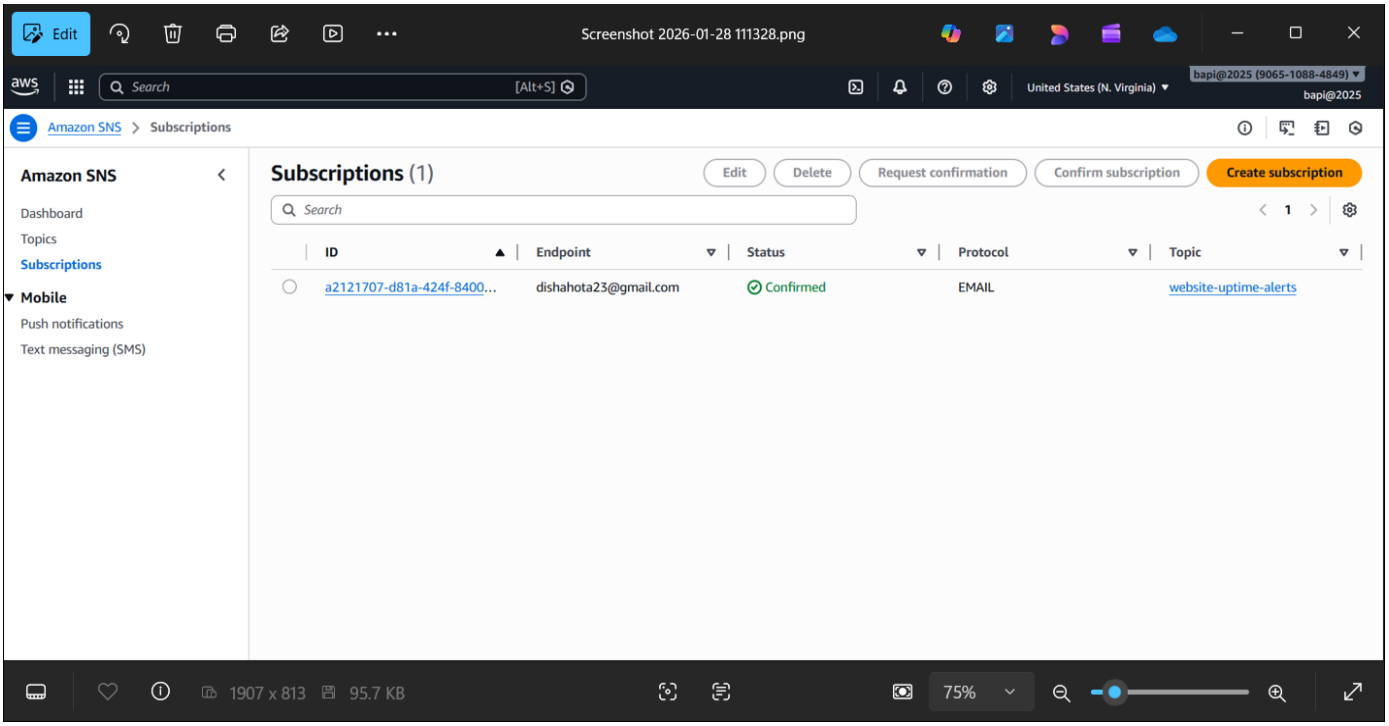
Step 6: Create SNS Email Subscription

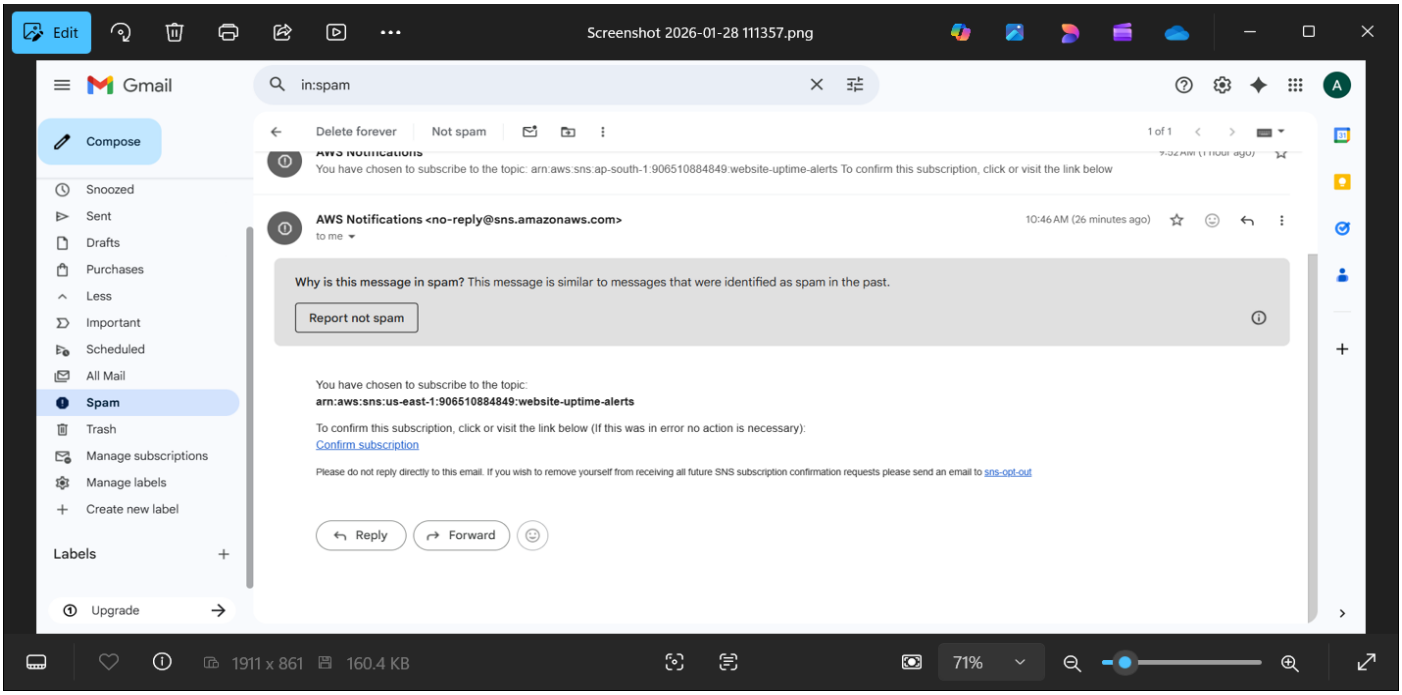
AWS Console Path: SNS → Subscriptions → Create subscription

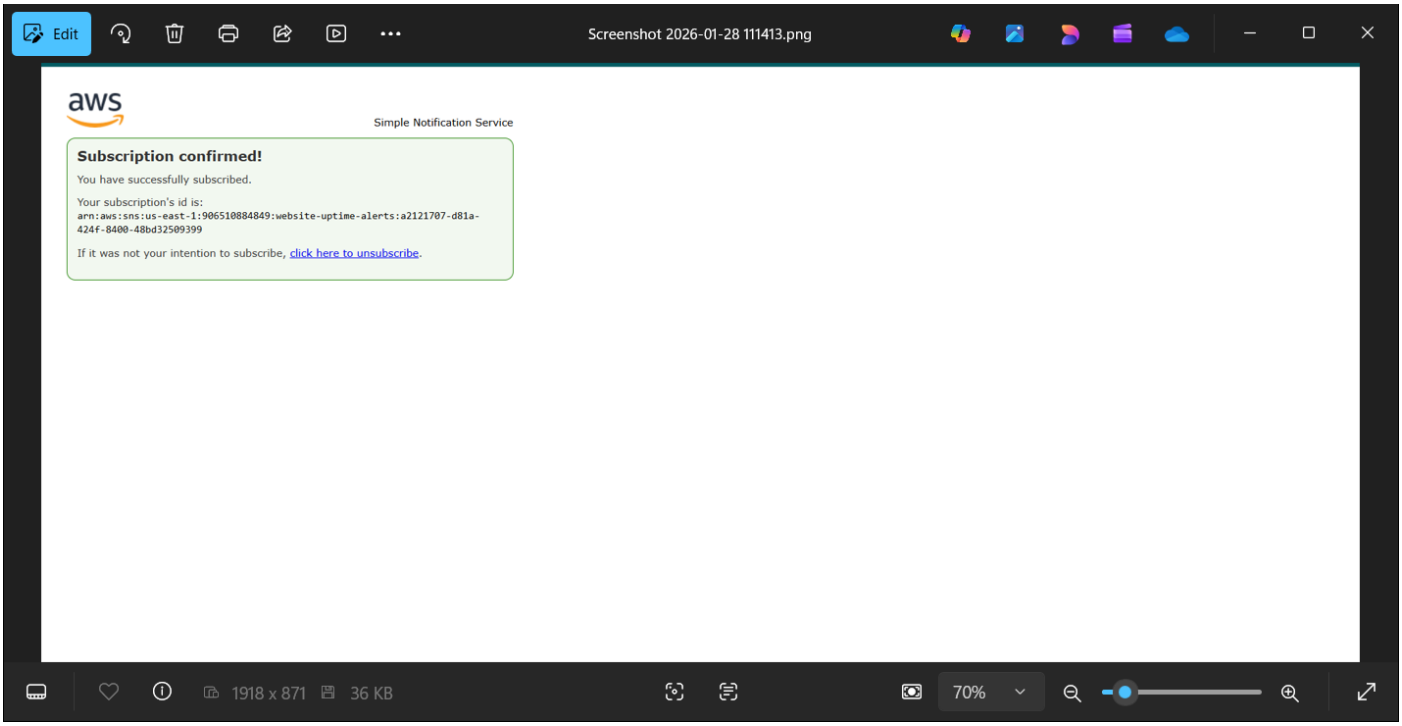
Configuration:

- Topic: website-uptime-alerts
- Protocol: Email
- Endpoint: your-dishahota23@gmail.com

The subscription is confirmed via email.


Edit
Screenshot 2026-01-28 111328.png
Search
[Alt+S]
United States (N. Virginia)
bapi@2025 (9065-1088-4849)
bapi@2025
Amazon SNS > Subscriptions
Amazon SNS
Dashboard
Topics
Subscriptions
Mobile
Push notifications
Text messaging (SMS)
Subscriptions (1)
Edit
Delete
Request confirmation
Confirm subscription
Create subscription
Search
ID
Endpoint
Status
Protocol
Topic
a2121707-d81a-424f-8400...
dishahota23@gmail.com
Confirmed
EMAIL
website-uptime-alerts
1907 x 813
95.7 KB
75%


Edit
Screenshot 2026-01-28 111357.png
Gmail
in:spam
Compose
Snoozed
Sent
Drafts
Purchases
Less
Important
Scheduled
All Mail
Spam
Trash
Manage subscriptions
Manage labels
Create new label
Labels
Upgrade
Delete forever
Not spam
1 of 1
You have chosen to subscribe to the topic: arn:aws:sns:ap-south-1:906510884849:website-uptime-alerts To confirm this subscription, click or visit the link below
AWS Notifications <no-reply@sns.amazonaws.com>
to me
10:46 AM (26 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
You have chosen to subscribe to the topic:
arn:aws:sns:us-east-1:906510884849:website-uptime-alerts
To confirm this subscription, click or visit the link below (If this was in error no action is necessary):
Confirm subscription
Please do not reply directly to this email. If you wish to remove yourself from receiving all future SNS subscription confirmation requests please send an email to sns-opt-out
Reply
Forward
1911 x 861
160.4 KB
71%

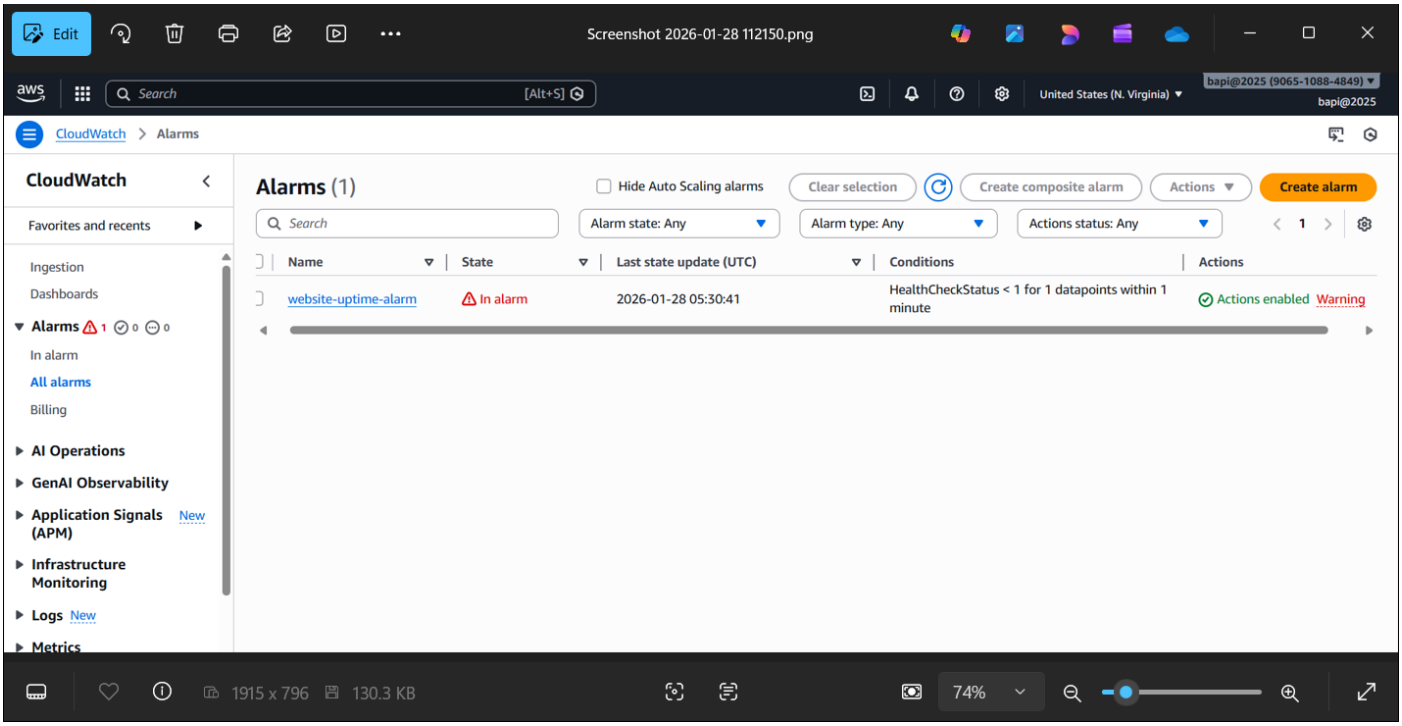


Step 7: Attach SNS to CloudWatch Alarm

In the CloudWatch Alarm configuration:

- Alarm state trigger: In alarm
- Notification target: website-uptime-alerts

This connects downtime detection to email alerts.



Step 8: Simulate Website Failure

A test domain (test-down-123.com) is used to simulate downtime. When the endpoint becomes unreachable:

- Route 53 marks it as Unhealthy
- CloudWatch Alarm enters ALARM state
- SNS sends an email notification