

# 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

## 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.

Route 53 < website-uptime-check

**Configuration**

ID: 1a35ef75-c669-46b4-b08a-b0eddd09f9e3 URL: https://test-down-123.com:443/ Specified endpoint by Domain name

State: Enabled Status: Unhealthy Inverted: No

**Advanced configuration**

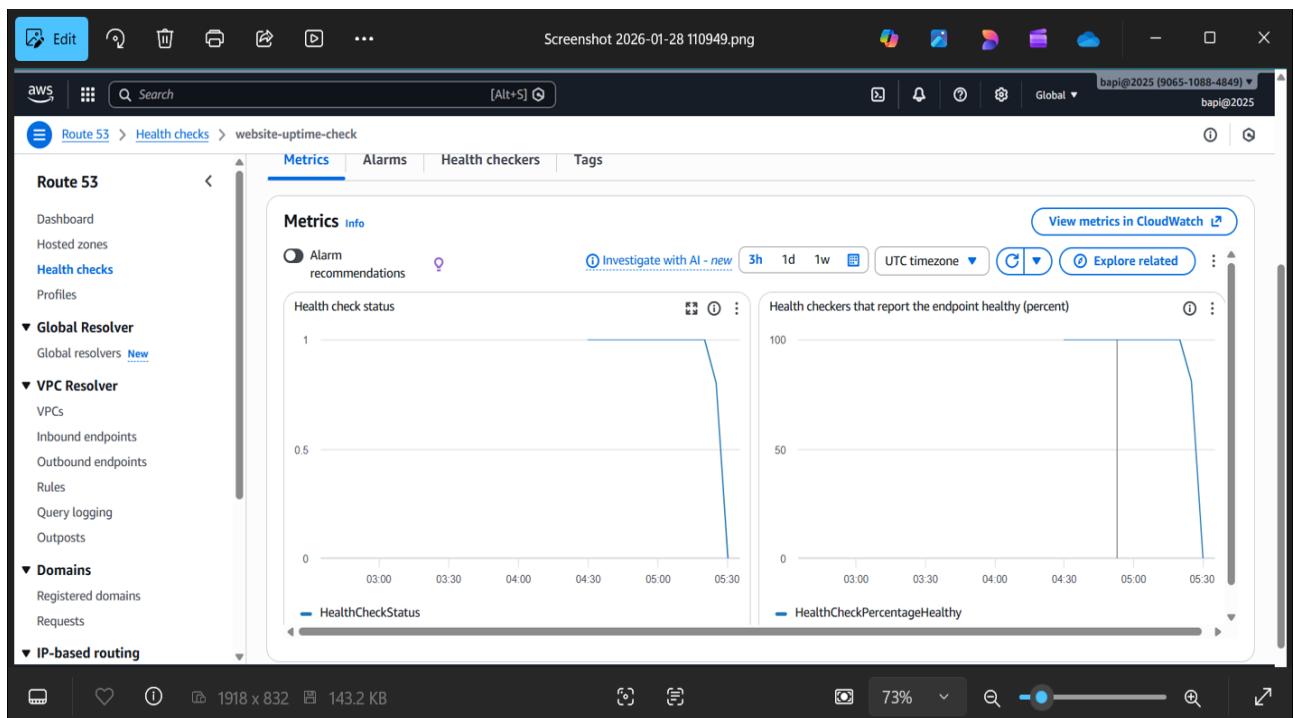
**Metrics** Alarms Health checkers Tags

**Metrics Info**

Alarm recommendations: 1 Investigate with AI - new 3h 1d 1w UTC timezone UTC Explore related

Health check status: 100% (blue line) and 0% (red line) from 03:30 to 05:30 UTC on Jan 28, 2026.

Health checkers that report the endpoint healthy (percent): 100% (blue line) and 0% (red line) from 03:30 to 05:30 UTC on Jan 28, 2026.



## 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.

Screenshot 2026-01-28 111049.png

The screenshot shows the AWS CloudWatch Alarms console. On the left, a sidebar lists navigation options like CloudWatch, Favorites and recents, Ingestion, Dashboards, Alarms (1), AI Operations, GenAI Observability, Application Signals (APM), Infrastructure Monitoring, Logs, and Metrics. The main area displays a single alarm named "website-uptime-alarm". The alarm details are as follows:

- Name:** website-uptime-alarm
- Type:** Metric alarm
- Description:** No description
- State:** In alarm
- Threshold:** HealthCheckStatus < 1 for 1 datapoints within 1 minute
- Actions:** Actions enabled
- Last state update:** 2026-01-28 05:30:41 (UTC)
- Namespace:** AWS/Route53
- Metric name:** HealthCheckStatus
- HealthCheckId:** 1a35ef75-c669-46b4-b08a-b0eddd09f9e3
- Statistic:** Minimum
- Period:** 1 minute
- Datapoints to alarm:** 1 out of 1
- Missing data treatment:** Treat missing data as missing
- Percentiles with low samples:** evaluate
- ARN:** arn:aws:cloudwatch:us-east-1:906510884849:alarm:website-uptime-alarm

At the bottom, there is a link to "View EventBridge rule".

Screenshot 2026-01-28 111038.png

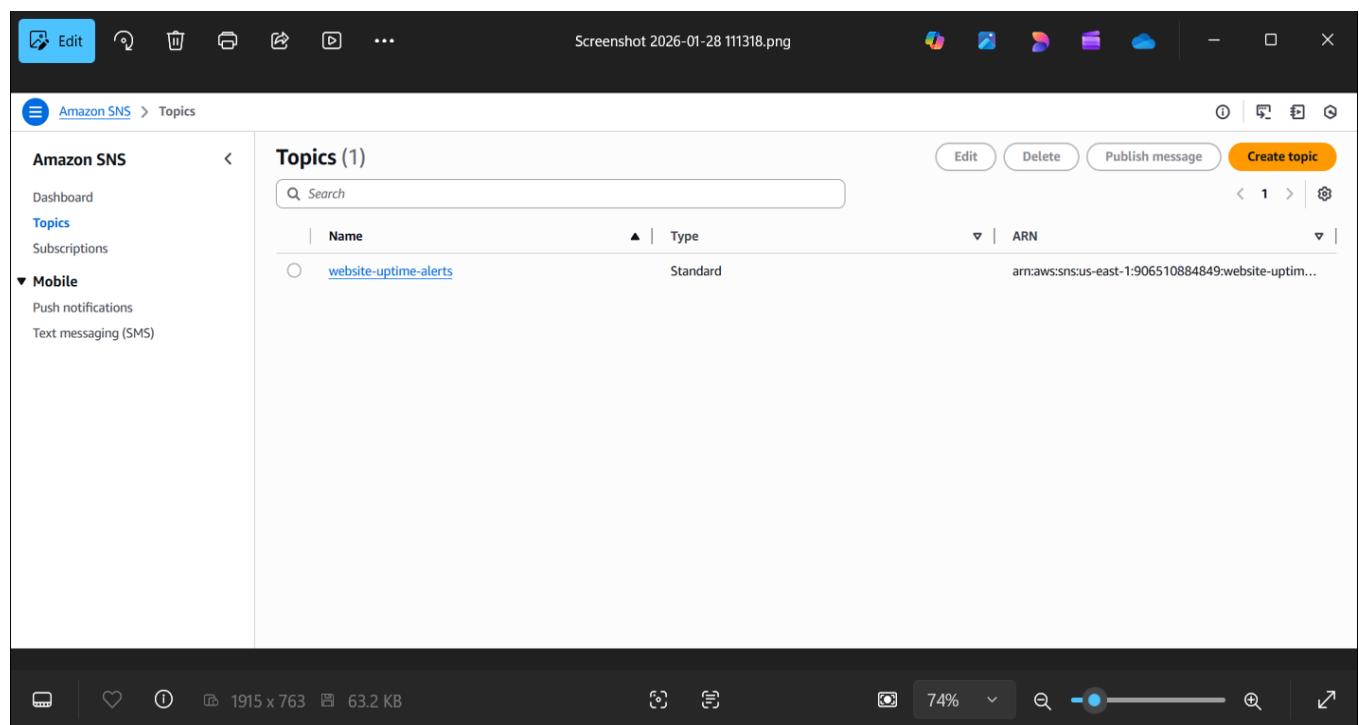
This screenshot is identical to the one above, showing the "website-uptime-alarm" in the CloudWatch Alarms console. It includes the same sidebar, alarm details, and a timeline chart for the metric "HealthCheckStatus" from 02:45 to 05:30. The chart shows a red bar indicating the alarm state from approximately 03:45 onwards.

## **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

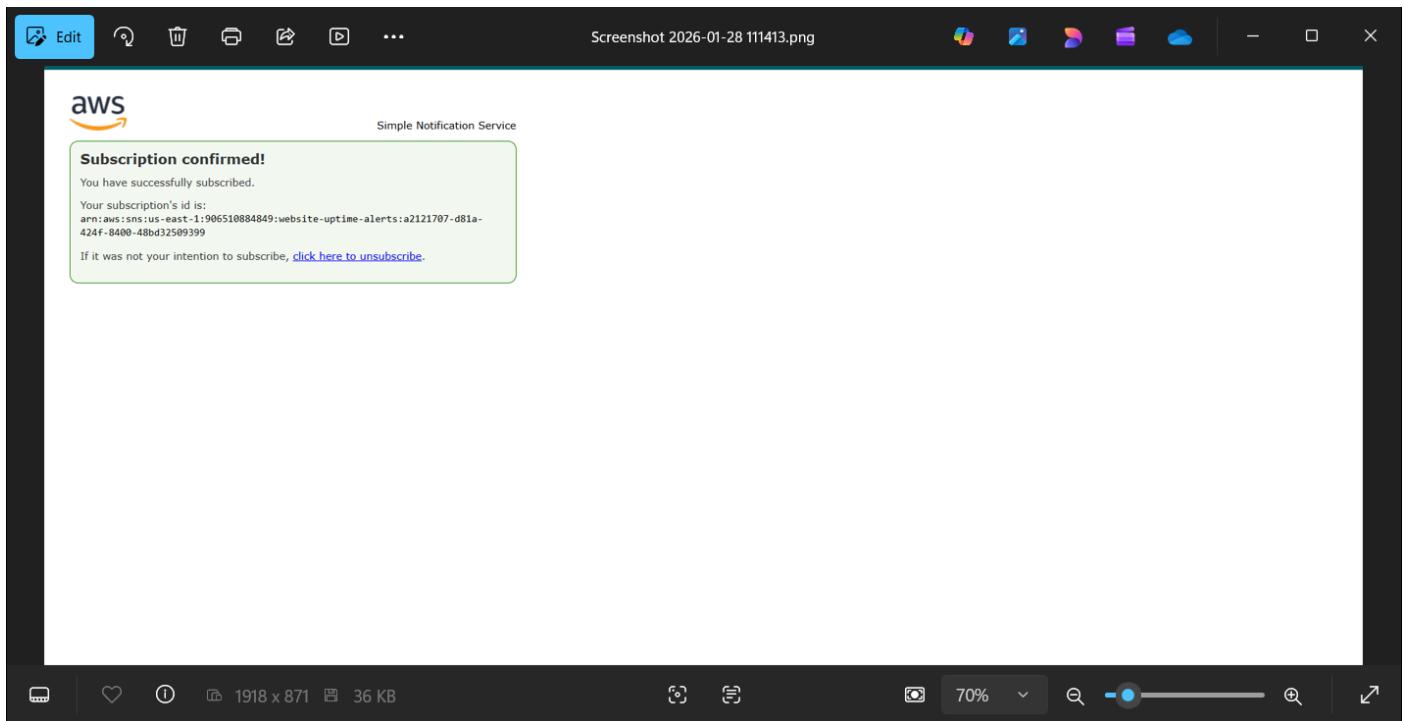
The subscription is confirmed via email.

Screenshot of the AWS SNS Subscriptions page. The left sidebar shows 'Amazon SNS' with 'Subscriptions' selected. The main area displays a table titled 'Subscriptions (1)'. The table has columns for ID, Endpoint, Status, Protocol, and Topic. One row is shown with the following details:

ID	Endpoint	Status	Protocol	Topic
a2121707-d81a-424f-8400...	dishahota23@gmail.com	Confirmed	EMAIL	website-uptime-alerts

Screenshot of a Gmail inbox. A single email from 'AWS Notifications <no-reply@sns.amazonaws.com>' is visible. The subject is '[REDACTED] To confirm this subscription, click or visit the link below'. The body of the email contains the following text:

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 (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](#).

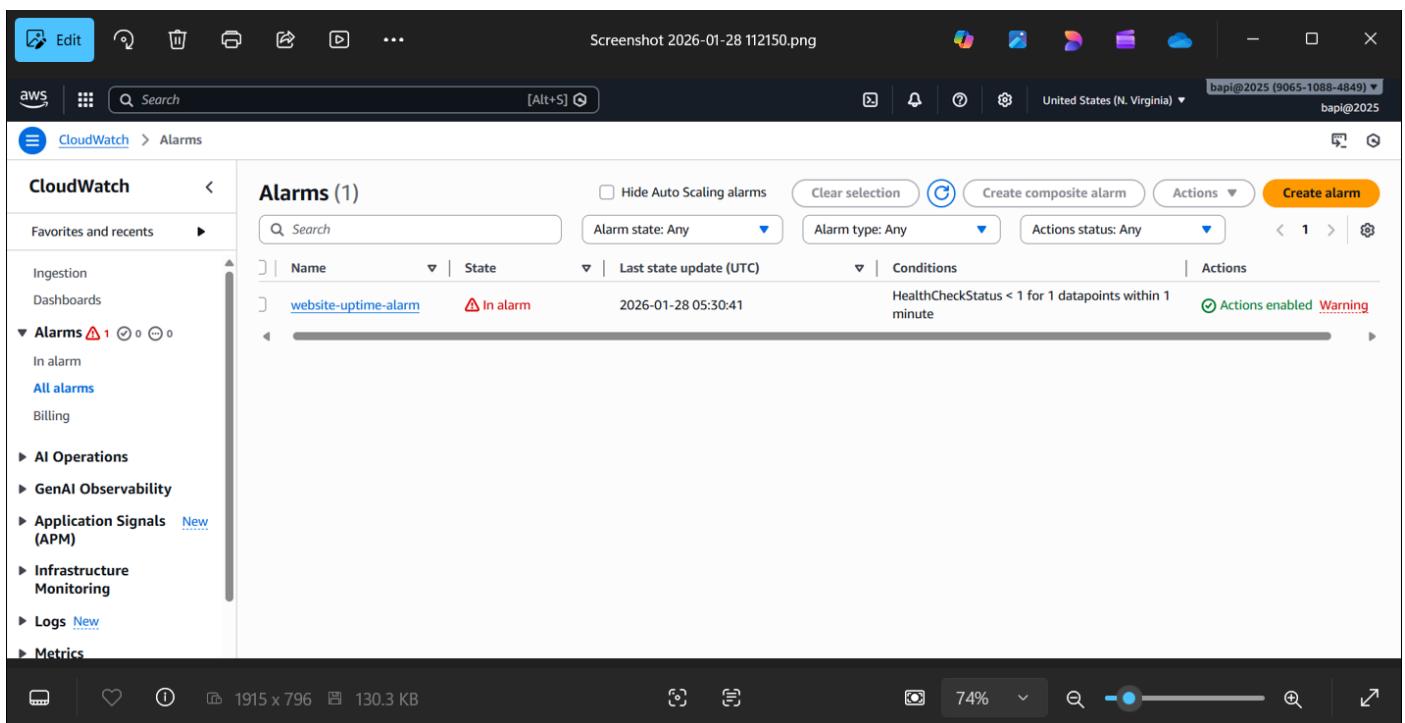


## Step 7: Attach SNS to CloudWatch Alarm

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

