

## Use case

### Cloud Resource Cleanup

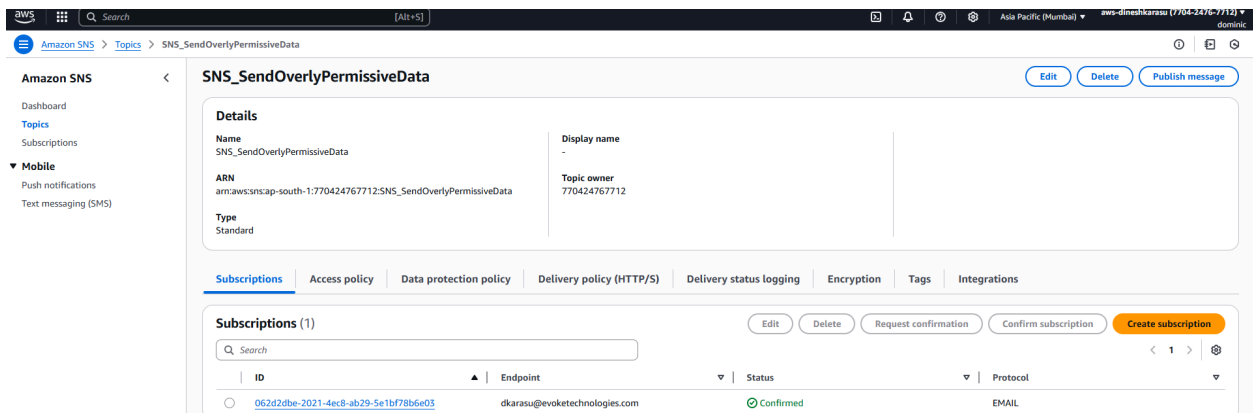
## Use case Description

Lambda triggered by EventBridge to identify stale resources

Delete or tag resources and send summary reports.

## Approach :

1. Create an SNS topic and subscribe to an email to send cleanup summary reports via email.
2. Configure Lambda
3. Set EventBridge Schedule
4. Create an EC2 instance and keep it in stopped state to test if it's terminated
5. Test



```
EXPLORER
└─ LAMBDA_CLEANUP_CLOUD_RESOURCES
   └─ lambda_function.py

DEPLOY
Deploy (Ctrl+Shift+U)
Test (Ctrl+Shift+I)

TEST EVENTS [NONE SELECTED]
+ Create new test event

ENVIRONMENT VARIABLES
>

lambda_function.py
1  import boto3
2  import datetime
3
4  # AWS clients
5  ec2 = boto3.client('ec2')
6  sns = boto3.client('sns')
7
8  # Replace with your actual SNS Topic ARN
9  SNS_TOPIC_ARN = 'arn:aws:sns:ap-south-1:7704247712:SNS_SendOverlyPermissiveData'
10
11 # Used to collect report lines
12 cleanup_report = []
13
14 # Calculate date 30 days ago for old snapshots
15 thirty_days_ago = datetime.datetime.now(datetime.timezone.utc) - datetime.timedelta(days=30)
16
17 def cleanup_stopped_instances():
18     response = ec2.describe_instances(
19         Filters=[('Name': 'instance-state-name', 'Values': ['stopped'])])
20
21     stopped_ids = []
22     for reservation in response['Reservations']:
23         for instance in reservation['Instances']:
24             stopped_ids.append(instance['InstanceId'])
25
26     if stopped_ids:
27         ec2.terminate_instances(InstanceIds=stopped_ids)
28         cleanup_report.append(f"Terminated stopped EC2 instances: {stopped_ids}")
29     else:
30         cleanup_report.append("No stopped EC2 instances found.")
31
32 def cleanup_unattached_volumes():
33     volumes = ec2.describe_volumes(
34         Filters=[('Name': 'status', 'Values': ['available'])])
35     volumes = volumes['Volumes']
36
37     for volume in volumes:
38         vol_id = volume['VolumeId']
39         # ...
```

```
lambda_function.py
33 def cleanup_unattached_volumes():
34     for volume in volumes:
35         vol_id = volume['VolumeId']
36         try:
37             ec2.delete_volume(VolumeId=vol_id)
38             cleanup_report.append(f"Deleted unattached EBS volume: {vol_id}")
39         except Exception as e:
40             cleanup_report.append(f"Could not delete volume {vol_id}: {str(e)}")
41
42     if not volumes:
43         cleanup_report.append("No unattached EBS volumes found.")
44
45 def cleanup_unused_eips():
46     addresses = ec2.describe_addresses()['Addresses']
47     released = False
48
49     for addr in addresses:
50         if 'InstanceId' not in addr:
51             try:
52                 ec2.release_address(AllocationId=addr['AllocationId'])
53                 cleanup_report.append(f"Released unused Elastic IP: {addr['PublicIp']}")
54                 released = True
55             except Exception as e:
56                 cleanup_report.append(f"Failed to release EIP {addr['PublicIp']}: {str(e)}")
57
58     if not released:
59         cleanup_report.append("No unused Elastic IPs found.")
60
61 def cleanup_old_snapshots():
62     snapshots = ec2.describe_snapshots(OwnerIds=['self'])['Snapshots']
63     old_snaps = [s for s in snapshots if s['StartTime'] < thirty_days_ago]
64
65     for snap in old_snaps:
66         snap_id = snap['SnapshotId']
67         try:
68             ec2.delete_snapshot(SnapshotId=snap_id)
69             cleanup_report.append(f"Deleted old snapshot: {snap_id}")
70         except Exception as e:
71             cleanup_report.append(f"Could not delete snapshot {snap_id}: {str(e)}")
```

```
lambda_function.py
49 def cleanup_unused_eips():
50     # ...
51
52 def cleanup_old_snapshots():
53     snapshots = ec2.describe_snapshots(OwnerIds=['self'])['Snapshots']
54     old_snaps = [s for s in snapshots if s['StartTime'] < thirty_days_ago]
55
56     for snap in old_snaps:
57         snap_id = snap['SnapshotId']
58         try:
59             ec2.delete_snapshot(SnapshotId=snap_id)
60             cleanup_report.append(f"Deleted old snapshot: {snap_id}")
61         except Exception as e:
62             cleanup_report.append(f"Could not delete snapshot {snap_id}: {str(e)}")
63
64     if not old_snaps:
65         cleanup_report.append("No old EBS snapshots found.")
66
67 def send_sns_summary():
68     message = "\n".join(cleanup_report)
69     sns.publish(
70         TopicArn=SNS_TOPIC_ARN,
71         Subject="AWS Cloud Resource Cleanup Report",
72         Message=message)
73
74 def lambda_handler(event, context):
75     cleanup_report.clear()
76     cleanup_report.append("=== Cloud Resource Cleanup Summary ===\n")
77
78     cleanup_stopped_instances()
79     cleanup_unattached_volumes()
80     cleanup_unused_eips()
81     cleanup_old_snapshots()
82
83     send_sns_summary()
84
85     return {
86         'statusCode': 200,
87         'body': 'Cleanup complete. Summary sent via SNS.'
88     }
```

Amazon EventBridge

Dashboard

Developer resources

Buses

Pipes

Scheduler

DeleteUnusedInstances

Schedule detail

Schedule name

DeleteUnusedInstances

Description

Schedule group name

default

Status

Enabled

Schedule ARN

arn:aws:scheduler:ap-south-1:770424767712:schedule/default/DeleteUnusedInstances

Action after completion

DELETE

Schedule start time

-

Schedule end time

-

Execution time zone

Asia/Calcutta

Flexible time window

-

Created date

Jul 07, 2025, 17:14:08 (UTC+05:30)

Last modified date

Jul 07, 2025, 17:14:08 (UTC+05:30)

Schedule

One time schedule

Your schedule is set for Mon, Jul 07, 2025, 17:20:00 (UTC+05:30).

EC2

Dashboard

EC2 Global View

Events

Instances

Instances (1/3)

Find Instance by attribute or tag (case-sensitive)

All states

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP
din-ec2-instan...	i-0ae26c2ac261334de	Terminated	t2.micro	-	View alarms +	ap-south-1b	-	-	-
My_Instance_2	i-0a3e1ddde5ac7a6cf	Terminated	t2.micro	-	View alarms +	ap-south-1b	-	-	-
ec-1-3	i-057ba66c5bdfd1766	Running	t2.micro	Initializing	View alarms +	ap-south-1b	ec2-13-203-79-46.ap-s...	13.203.79.46	-

## AWS Cloud Resource Cleanup Report

AN

AWS Notifications<no-reply@sns.amazonaws.com>

To: Dinesh Karasu

Mon 7/7/2025 5:20 PM

CAUTION: This email originated from outside of the organization. Do not click links or open attachments unless you recognize the sender and know the content is safe.

=== Cloud Resource Cleanup Summary ===

- Terminated stopped EC2 instances: [i-057ba66c5bdfd1766]
- No unattached EBS volumes found.
- No unused Elastic IPs found.
- No old EBS snapshots found.

EC2

Dashboard

EC2 Global View

Events

Instances

Instances (1/3)

Find Instance by attribute or tag (case-sensitive)

All states

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP
din-ec2-instan...	i-0ae26c2ac261334de	Terminated	t2.micro	-	View alarms +	ap-south-1b	-	-	-
My_Instance_2	i-0a3e1ddde5ac7a6cf	Terminated	t2.micro	-	View alarms +	ap-south-1b	-	-	-
ec-1-3	i-057ba66c5bdfd1766	Terminated	t2.micro	-	View alarms +	ap-south-1b	-	-	-

i-057ba66c5bdfd1766 (ec-1-3)

Details

Status and alarms

Monitoring

Security

Networking

Storage

Tags

Instance summary

Instance ID

i-057ba66c5bdfd1766

Public IPv4 address

-

Private IPv4 addresses

-