

Use case

Auto-Remediation Using CloudWatch + Lambda

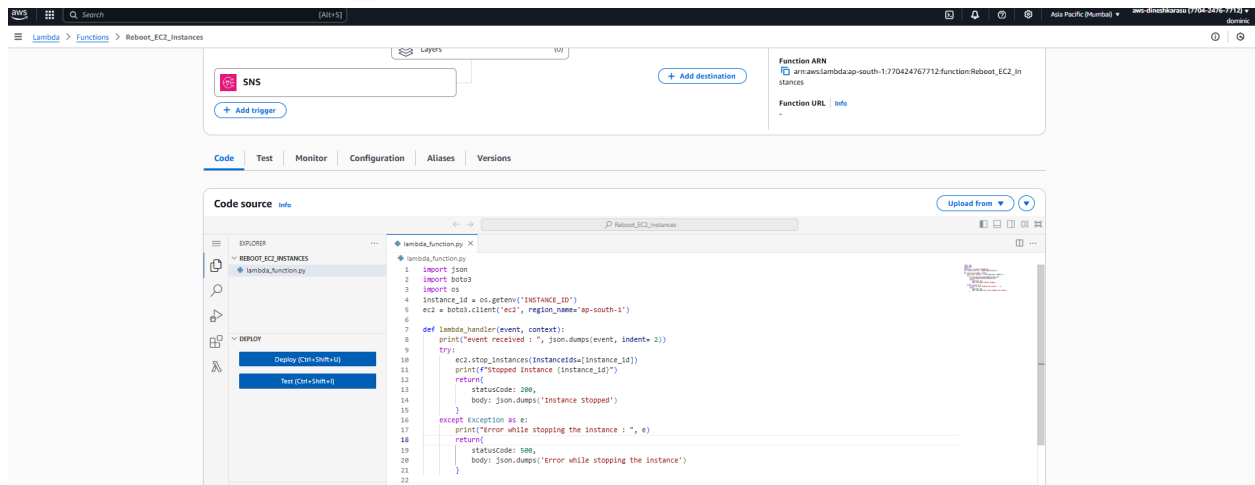
Use case Description

Create a CloudWatch alarm for EC2 status checks.

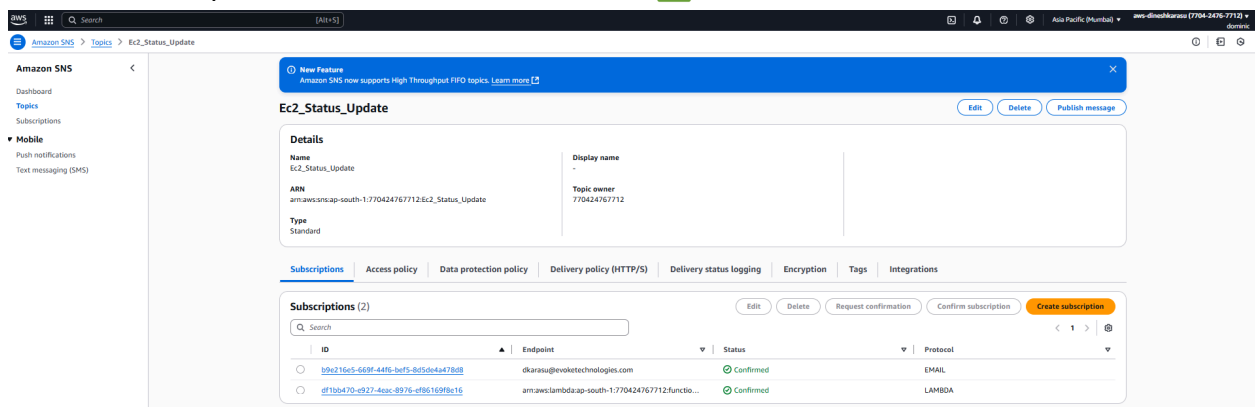
Trigger a Lambda function to restart the instance or notify via SNS

Approach :

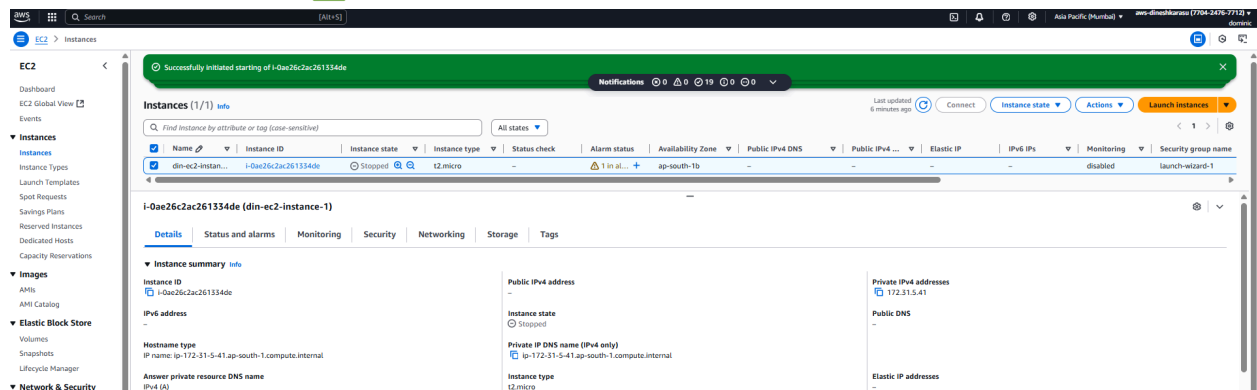
create a Lambda function ✓



create a SNS topic and subscribe the lambda to it ✓



create an EC2 instance ✓



The screenshot shows the AWS Management Console for an EC2 instance named 'din-ec2-instance-1' with ID 'i-0ae26c2ac261334de'. The instance is in a 'Stopped' state. The console displays various details including the instance type (t2.micro), availability zone (ap-south-1b), and public IP address (172.31.5.41). The instance is associated with the 'launch-wizard-1' security group. The console also shows the instance's hostname, IP name, and answer private resource DNS name.

Create cloudwatch alarm to monitor running state of EC2 instance ⚠

CloudWatch does not natively provide direct metrics for instance state transitions like "Running"

So, We'll do this:

Use EventBridge to detect when an EC2 instance enters the running state.

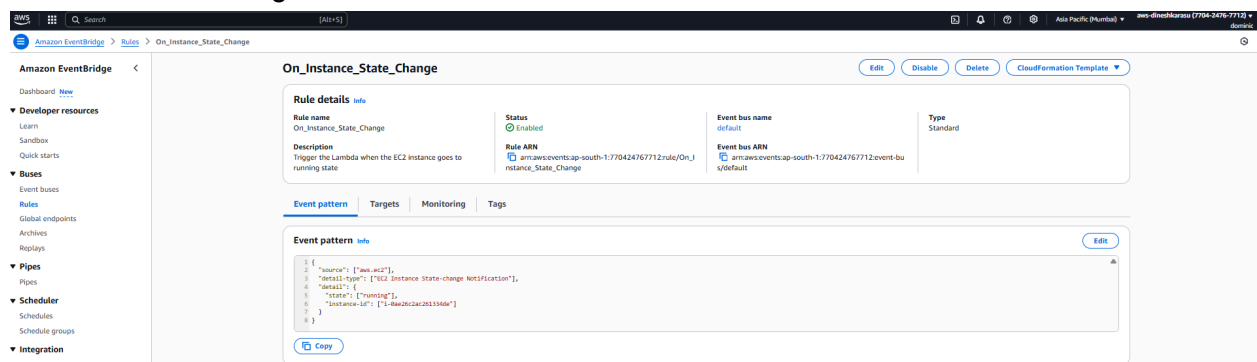
Trigger a Lambda function from EventBridge.

The Lambda will publish a custom CloudWatch metric.

Create a CloudWatch Alarm on that custom metric.

This way, CloudWatch Alarm triggers when the instance goes to running state, even from stopped.

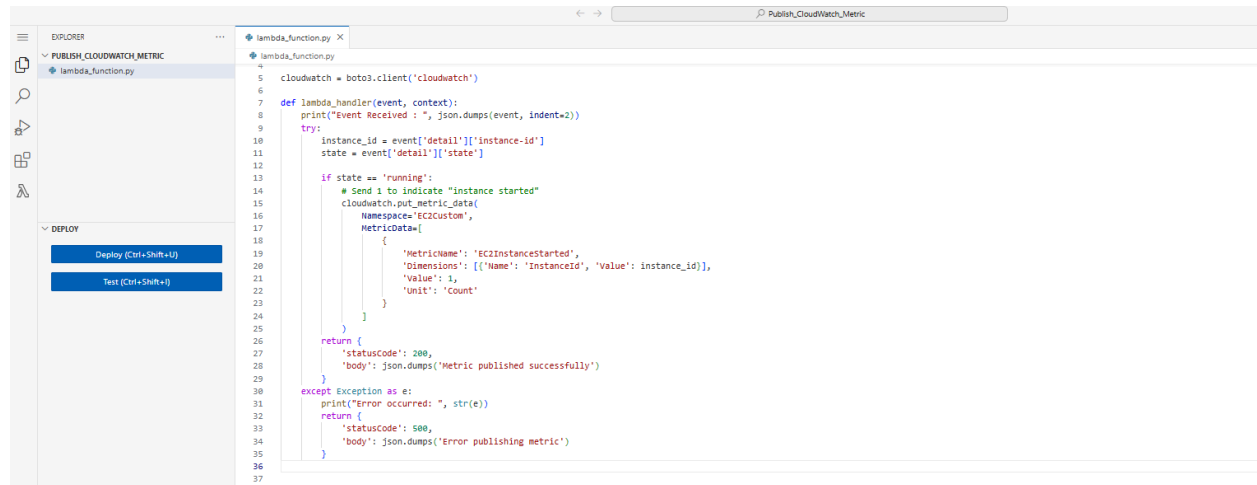
Create an event bridge rule



The screenshot shows the Amazon EventBridge console for an event rule named 'On_Instance_State_Change'. The rule is enabled and triggers the Lambda function 'aws-ec2-instance-state-change-notification' when an EC2 instance enters the 'running' state. The event pattern is defined as follows:

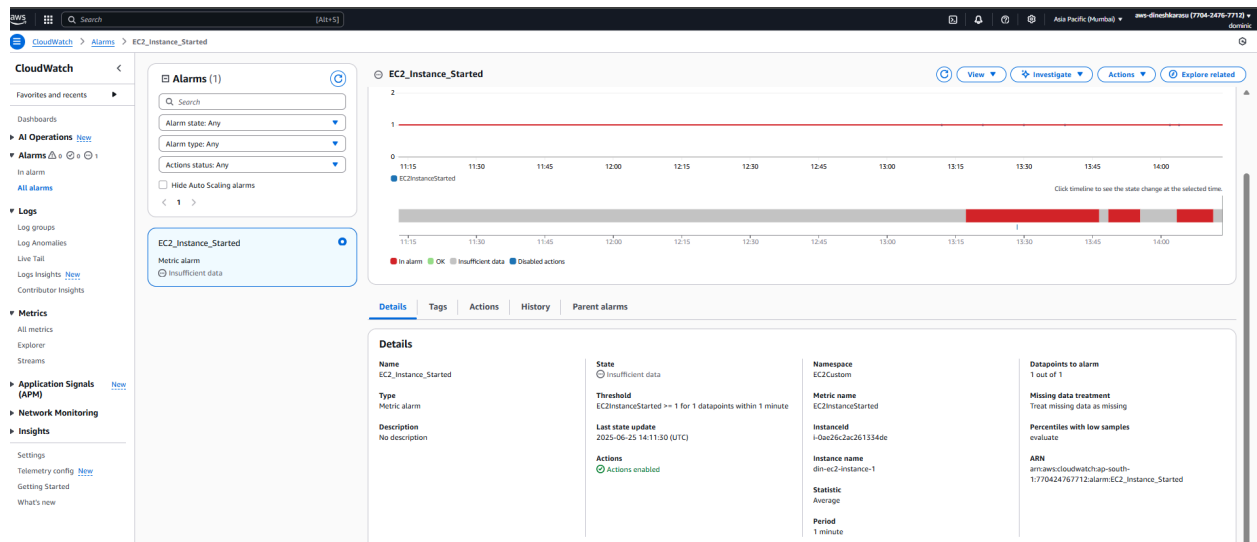
```
1 {
2   "source": ["aws.ec2"],
3   "detail-type": ["EC2 Instance State-change Notification"],
4   "detail": {
5     "state": ["running"],
6     "instance-id": ["i-0ae26c2ac261334de"]
7   }
8 }
```

Create Lambda function to publish cloudwatch metric

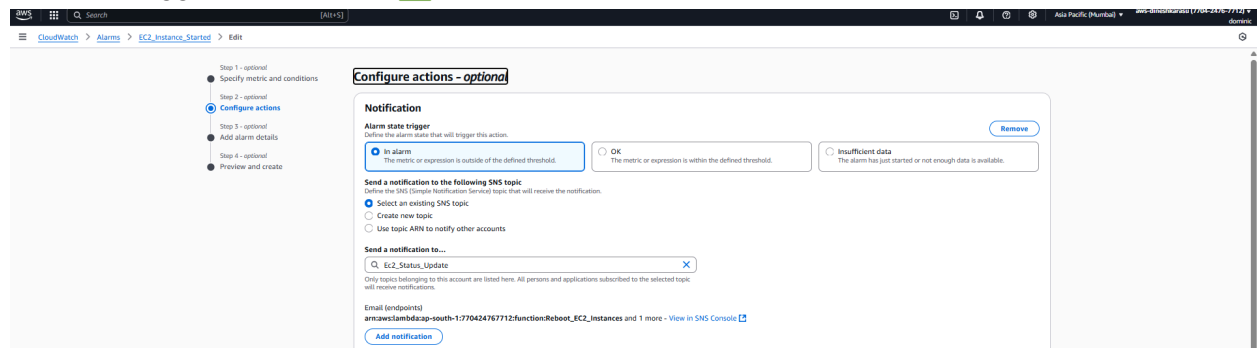


```
1  lambda_function.py
2
3  cloudwatch = boto3.client('cloudwatch')
4
5  def lambda_handler(event, context):
6      print("Event Received : ", json.dumps(event, indent=2))
7
8      try:
9          instance_id = event['detail']['instance-id']
10         state = event['detail']['state']
11
12         if state == 'running':
13             # Send 1 to indicate "instance started"
14             cloudwatch.put_metric_data(
15                 Namespace='EC2Custom',
16                 MetricData=[
17                     {
18                         'MetricName': 'EC2InstanceStarted',
19                         'Dimensions': [{'Name': 'InstanceId', 'Value': instance_id}],
20                         'Value': 1,
21                         'Unit': 'Count'
22                     }
23                 ]
24             )
25
26             return {
27                 'statusCode': 200,
28                 'body': json.dumps('Metric published successfully')}
29
30         except Exception as e:
31             print("Error occurred: ", str(e))
32             return {
33                 'statusCode': 500,
34                 'body': json.dumps('Error publishing metric')}
35
36
37
```

Create CloudWatch Alarm on the created metric



set alarm trigger to SNS topic



The screenshot shows the 'Configure actions - optional' step in the AWS CloudWatch console. The alarm state trigger is set to 'In alarm'. The notification is configured to send to the following SNS topic:

Send a notification to the following SNS topic:

- Select an existing SNS topic
- Create new topic
- Use topic ARN to notify other accounts

Send a notification to...

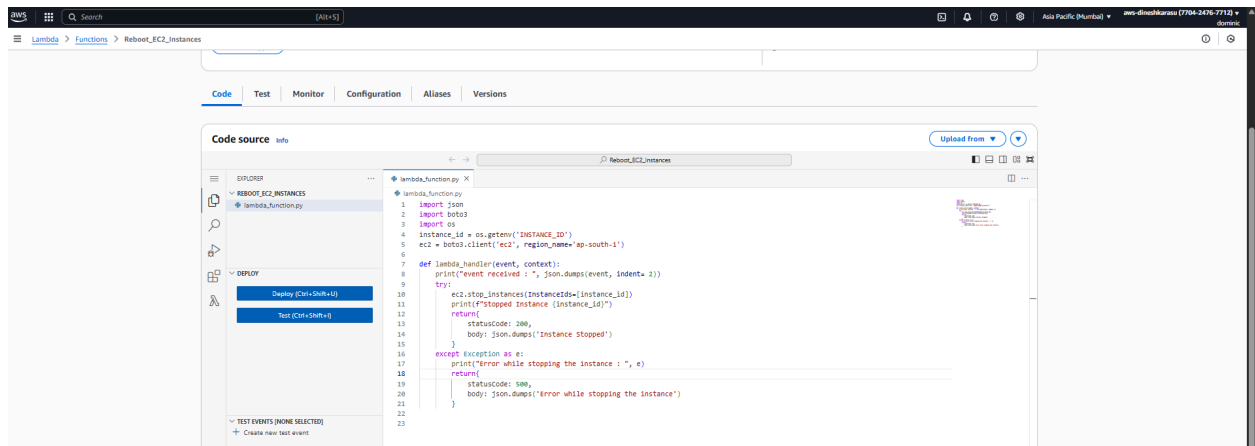
EC2_Instance_Started

Only topics belonging to this account are listed here. All persons and applications subscribed to the selected topic will receive notifications.

Email (endpoints): arnaws:lambda:ap-south-1:770424767712:function:Reboot_EC2_Instance and 1 more - View in SNS Console

Add notification

Write py code in Lambda to Stop the instance ✓



Test:

1. Change the instance state from stopped to Running
2. The alarm goes to In-alarm state
3. The instance gets stopped automatically