Assignment 1 - EventBridge

Problem Statement:

Create a CloudWatch Event Rule which will monitor state changes for all EC2 instances. Add a filter to the CloudWatch Event Rule to monitor only these state changes: (pending, running, stopped,terminated). Create a Lambda function as a target for the CloudWatch Event Rule. This lambda function will do the following operations:

- Read instance details from CloudWatch Event.
- Creates a new record in the DynamoDB table if the record doesn't exist in DynamoDB for that instance. If a record already exists for the instance in DDB, then updates that record with the current instance state.

Create a SAM template for deploying the complete solution.

Solution:

SAM template for creating lambda function which will trigger when instance changes happen.

```
AWSTemplateFormatVersion: '2010-09-09'
Transform: AWS::Serverless-2016-10-31
Description: >
  ec2-state-monitor
  Sample SAM Template for ec2-state-monitor
Resources:
  EC2StateMonitorFunction:
    Type: AWS::Serverless::Function
    Properties:
      Handler: app.lambda_handler
      Runtime: python3.8
      CodeUri: ec2_state_monitor
      Timeout: 30
      Environment:
        Variables:
          TABLE_NAME: !Ref EC2StateTable
      Policies:
        - DynamoDBCrudPolicy:
            TableName: !Ref EC2StateTable
      Events:
        EC2StateEvent:
          Type: CloudWatchEvent
          Properties:
            Pattern:
              source:
                - aws.ec2
              detail-type:
                - EC2 Instance State-change Notification
              detail:
                state:
                  - pending
                  - running
                  - stopped
                  - terminated
  EC2StateTable:
    Type: AWS::DynamoDB::Table
    Properties:
      AttributeDefinitions:
        - AttributeName: instance_id
          AttributeType: S
      KeySchema:
        - AttributeName: instance_id
          KeyType: HASH
      ProvisionedThroughput:
        ReadCapacityUnits: 5
        WriteCapacityUnits: 5
      TableName: EC2StateTable
```

Lambda Code to read the instance details from the event and update the data in the Dynamodb table.

```
import json
import boto3
import os
from botocore.exceptions import ClientError

client = boto3.client('dynamodb')
table_name = os.environ['TABLE_NAME']

def dynamodb_update(instanceid,state,time):
    try:
        response = client.put_item(
            TableName= table_name,
            Item={
                "instance_id": {
                    "S": instanceid
                },
                "state": {
                    "S": state
                },
                "time": {
                    "S": time
                }
            }
        )
        StatusCode = 200
        Body = "Success"
    except ClientError as e:
        StatusCode = 400
        Body = "Failed"
    return {
        'statusCode': StatusCode,
        'body': json.dumps(Body)
    }

def lambda_handler(event, context):
    instanceid = event['detail']['instance-id']
    state = event['detail']['state']
    time = event['time']
    dynamodb_update(instanceid,state,time)
```

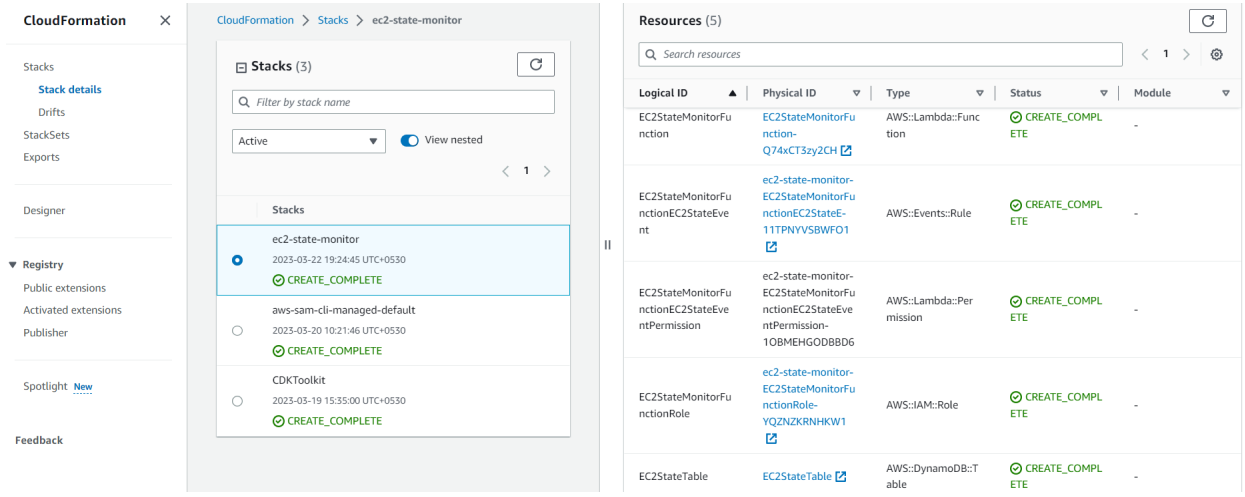
SAM deploy

```
[root@ip-172-31-0-37 ec2-state-monitor]# sam deploy

                Managed S3 bucket: aws-sam-cli-managed-default-samclisourcebucket-1nq2jloczupzn
                A different default S3 bucket can be set in samconfig.toml
                Or by specifying --s3-bucket explicitly.
        Uploading to 571143ab8f8cd318a2c64940be14169c  496 / 496  (100.00%)

        Deploying with following values
        ===============================
        Stack name                   : ec2-state-monitor
        Region                       : ap-south-1
        Confirm changeset            : True
        Disable rollback             : False
        Deployment s3 bucket         : aws-sam-cli-managed-default-samclisourcebucket-1nq2jloczupzn
        Capabilities                 : ["CAPABILITY_IAM"]
        Parameter overrides          : {}
        Signing Profiles             : {}
```

CloudFormation stack changeset

Operation	LogicalResourceId	ResourceType	Replacement
+ Add	EC2StateMonitorFunctionEC2StateEventPermission	AWS::Lambda::Permission	N/A
+ Add	EC2StateMonitorFunctionEC2StateEvent	AWS::Events::Rule	N/A
+ Add	EC2StateMonitorFunctionRole	AWS::IAM::Role	N/A
+ Add	EC2StateMonitorFunction	AWS::Lambda::Function	N/A
+ Add	EC2StateTable	AWS::DynamoDB::Table	N/A

Cloudformation created all resources specified in the SAM template.



States changes in DynamoDB table