**https://github.com/suvaatnbu/serverlessaaws**

**Description**

**General Notes:**

**Submission:**

- Provide the Python code in GitHub and share the GitHub link as a submission.

- Provide brief documentation on the steps followed.

- Take a screenshot of the steps and put them into the documentation.

**Evaluation Criteria:**

- Correctness of the code.

- The ability to complete the task.

- Clarity and organization of documentation.

**You are only required to do any 4 of the below for the grading. However, it is recommended that you practice as much as possible, which will help you build your expertise with automation and various AWS services.**

**Tasks:**

**Assignment 1: Automated Instance Management Using AWS Lambda and Boto3**

**Objective:** In this assignment, you will gain hands-on experience with AWS Lambda and Boto3, Amazon's SDK for Python. You will create a Lambda function that will automatically manage EC2 instances based on their tags.

**Task:** You're tasked to automate the stopping and starting of EC2 instances based on tags. Specifically:

1. Setup:

   - Create two EC2 instances.

   - Tag one of them as `Auto-Stop` and the other as `Auto-Start`.

2. Lambda Function Creation:

   - Set up an AWS Lambda function.

   - Ensure that the Lambda function has the necessary IAM permissions to describe, stop, and start EC2 instances.

3. Coding:

   - Using Boto3 in the Lambda function:

     - Detect all EC2 instances with the `Auto-Stop` tag and stop them.

     - Detect all EC2 instances with the `Auto-Start` tag and start them.

4. Testing:

   - Manually invoke the Lambda function.

   - Confirm that the instance tagged `Auto-Stop` stops and the one tagged `Auto-Start` starts.

**Instructions:**

1. EC2 Setup:

   - Navigate to the EC2 dashboard and create two new t2.micro instances (or any other available free-tier type).

   - Tag the first instance with a key `Action` and value `Auto-Stop`.

   - Tag the second instance with a key `Action` and value `Auto-Start`.

2. Lambda IAM Role:

   - In the IAM dashboard, create a new role for Lambda.

   - Attach the `AmazonEC2FullAccess` policy to this role. (Note: In a real-world scenario, you would want to limit permissions for better security.)

3. Lambda Function:

   - Navigate to the Lambda dashboard and create a new function.

   - Choose Python 3.x as the runtime.

   - Assign the IAM role created in the previous step.

   - Write the Boto3 Python script to:

     1. Initialize a boto3 EC2 client.

     2. Describe instances with `Auto-Stop` and `Auto-Start` tags.

     3. Stop the `Auto-Stop` instances and start the `Auto-Start` instances.

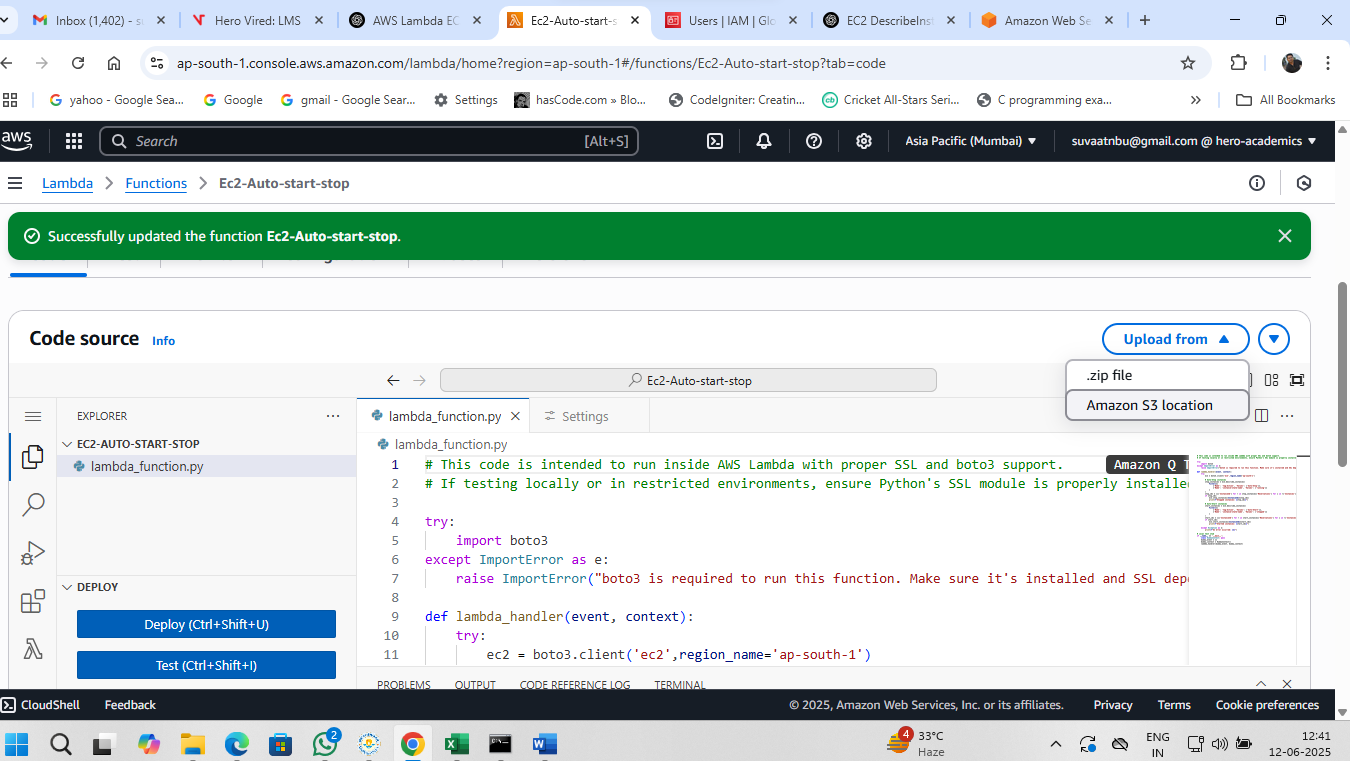
     4. Print instance IDs that were affected for logging purposes.

4. Manual Invocation:

   - After saving your function, manually trigger it.

   - Go to the EC2 dashboard and confirm that the instances' states have changed according to their tags.

ANWER:



# This code is intended to run inside AWS Lambda with proper SSL and boto3 support.

# If testing locally or in restricted environments, ensure Python's SSL module is properly installed.

try:

    import boto3

except ImportError as e:

    raise ImportError("boto3 is required to run this function. Make sure it's installed and SSL dependencies are available.") from e

def lambda\_handler(event, context):

    try:

        ec2 = boto3.client('ec2',region\_name='ap-south-1')

        # Auto-Stop instances

        stop\_instances = ec2.describe\_instances(

            Filters=[

                {'Name': 'tag:Action', 'Values': ['Auto-Stop']},

                {'Name': 'instance-state-name', 'Values': ['running']}

            ]

        )

        stop\_ids = [i['InstanceId'] for r in stop\_instances['Reservations'] for i in r['Instances']]

        if stop\_ids:

            ec2.stop\_instances(InstanceIds=stop\_ids)

            print(f"Stopped instances: {stop\_ids}")

        # Auto-Start instances

        start\_instances = ec2.describe\_instances(

            Filters=[

                {'Name': 'tag:Action', 'Values': ['Auto-Start']},

                {'Name': 'instance-state-name', 'Values': ['stopped']}

            ]

        )

        start\_ids = [i['InstanceId'] for r in start\_instances['Reservations'] for i in r['Instances']]

        if start\_ids:

            ec2.start\_instances(InstanceIds=start\_ids)

            print(f"Started instances: {start\_ids}")

    except Exception as e:

        print(f"An error occurred: {e}")

# Local test stub

if \_\_name\_\_ == '\_\_main\_\_':

    class DummyContext: pass

    dummy\_event = {}

    dummy\_context = DummyContext()

    lambda\_handler(dummy\_event, dummy\_context)

