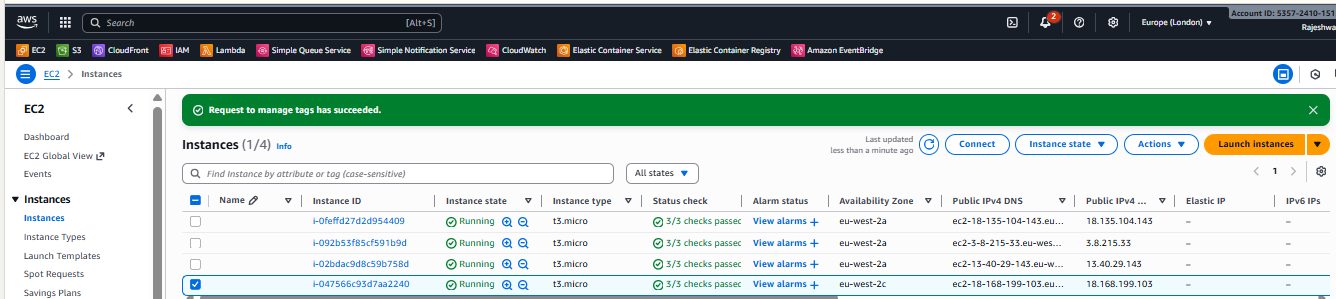
**Assignment 1: Automated Instance Management Using AWS Lambda and Boto3**

**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.

Create four EC2 instances:- Using Launch Instance

Create instance free:- use t3.micro



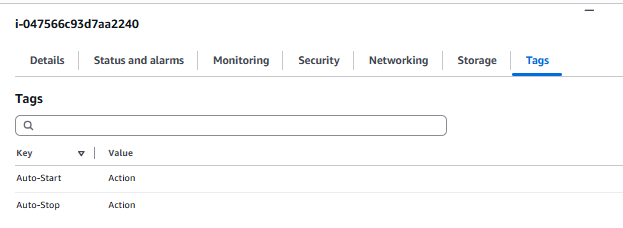
Tag one of them as `Auto-Stop` and the other as `Auto-Start`.

After running the instance select the instance and go on “Tags”

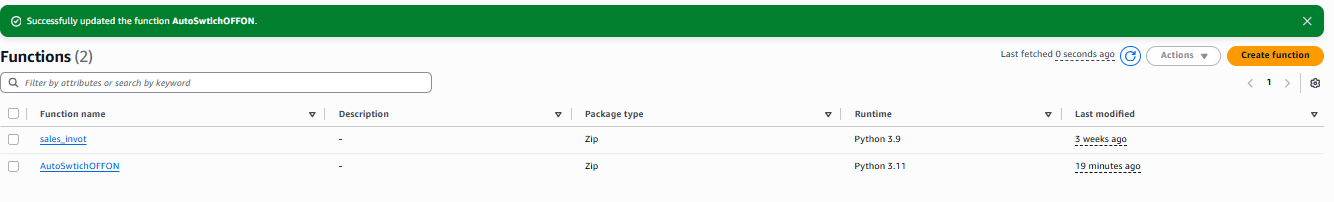
Create 2 Tags:- “Auto-Start” and the value should be “Action”

“Auto-Stop” and the value should be “Action”

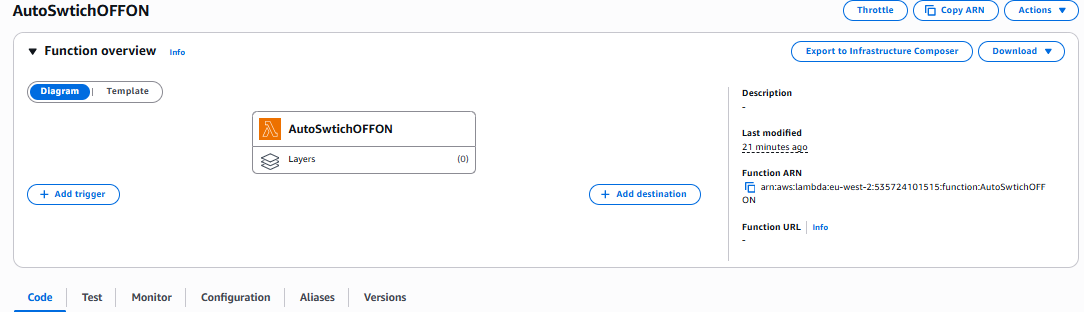
Like below example.



2. Lambda Function need to create for manage Instance status.

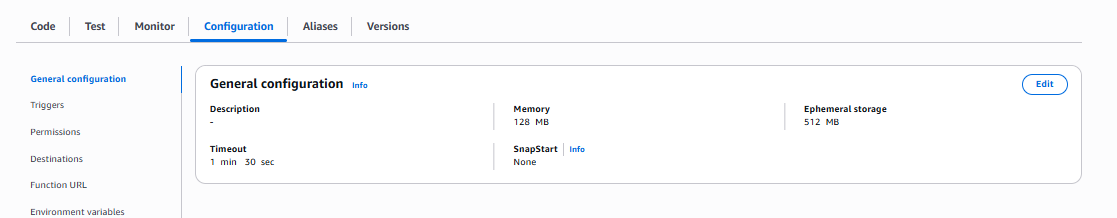


1. Click on “Create Function”

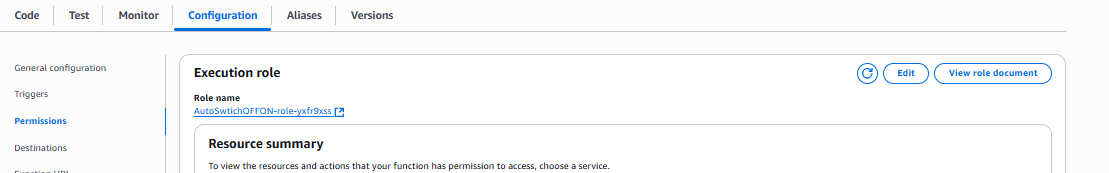


1. Go on “Configuration”.

Change the time from default to “1 min 30 sec”

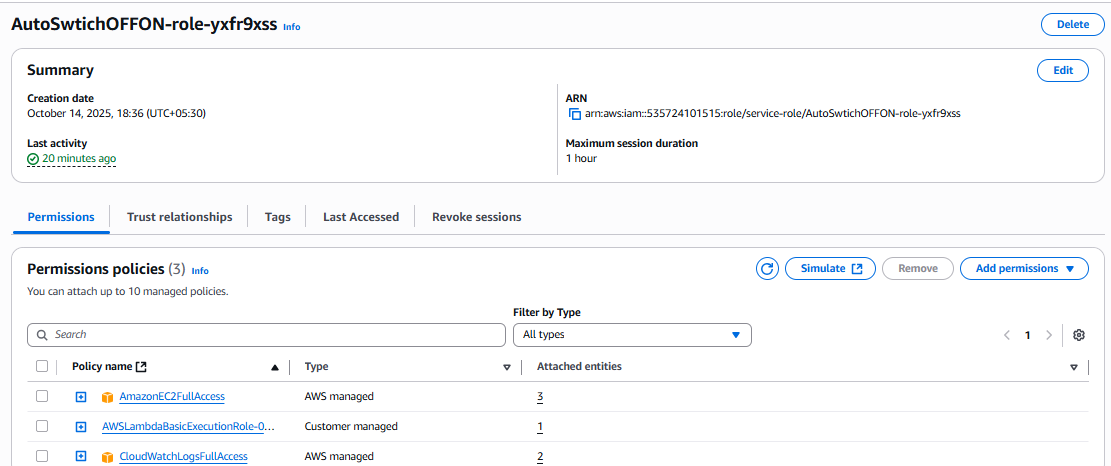


Change in “Permission”

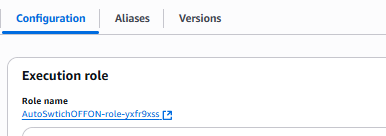


Create Permission in IAM role.

Give attached policies.



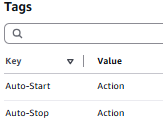
Select same role in “Permission”



3. Coding:

Create file name Lambda\_Funcation.py.

Add below code. In this code add Key “Auto-Start and Auto-Stop” with Value “Action”.



import boto3

import os

def lambda\_handler(event, context):

    ec2 = boto3.client('ec2')

    started\_instances = []

    stopped\_instances = []

    # --- Find instances with the "Auto-Start" tag ---

    auto\_start\_instances = ec2.describe\_instances(

        Filters=[

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

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

        ]

    )

    for reservation in auto\_start\_instances['Reservations']:

        for instance in reservation['Instances']:

            instance\_id = instance['InstanceId']

            started\_instances.append(instance\_id)

    if started\_instances:

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

        print(f"Started instances: {started\_instances}")

    else:

        print("No instances to start.")

    # --- Find instances with the "Auto-Stop" tag ---

    auto\_stop\_instances = ec2.describe\_instances(

        Filters=[

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

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

        ]

    )

    for reservation in auto\_stop\_instances['Reservations']:

        for instance in reservation['Instances']:

            instance\_id = instance['InstanceId']

            stopped\_instances.append(instance\_id)

    if stopped\_instances:

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

        print(f"Stopped instances: {stopped\_instances}")

    else:

        print("No instances to stop.")

    return {

        'statusCode': 200,

        'body': {

            'Started': started\_instances,

            'Stopped': stopped\_instances

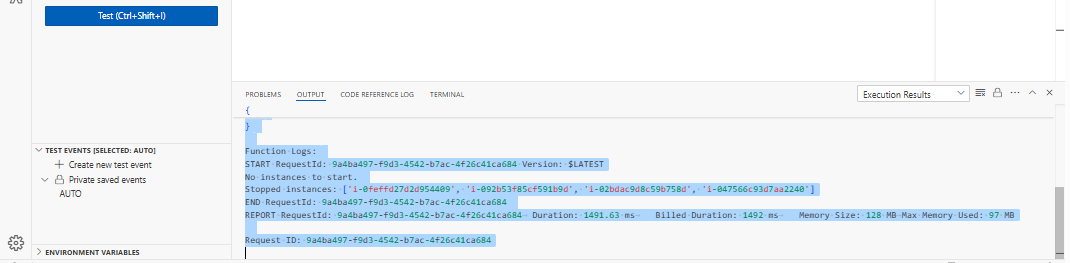
        }

    }

Click on “Deploy”

4. Testing:

Then click on “Test” to check this code is working properly.



Output like below.

Status: Succeeded

Test Event Name: AUTO

Response:

{

  "statusCode": 200,

  "body": {

    "Started": [],

    "Stopped": [

      "i-0feffd27d2d954409",

      "i-092b53f85cf591b9d",

      "i-02bdac9d8c59b758d",

      "i-047566c93d7aa2240"

    ]

  }

}

Function Logs:

START RequestId: 9a4ba497-f9d3-4542-b7ac-4f26c41ca684 Version: $LATEST

No instances to start.

Stopped instances: ['i-0feffd27d2d954409', 'i-092b53f85cf591b9d', 'i-02bdac9d8c59b758d', 'i-047566c93d7aa2240']

END RequestId: 9a4ba497-f9d3-4542-b7ac-4f26c41ca684

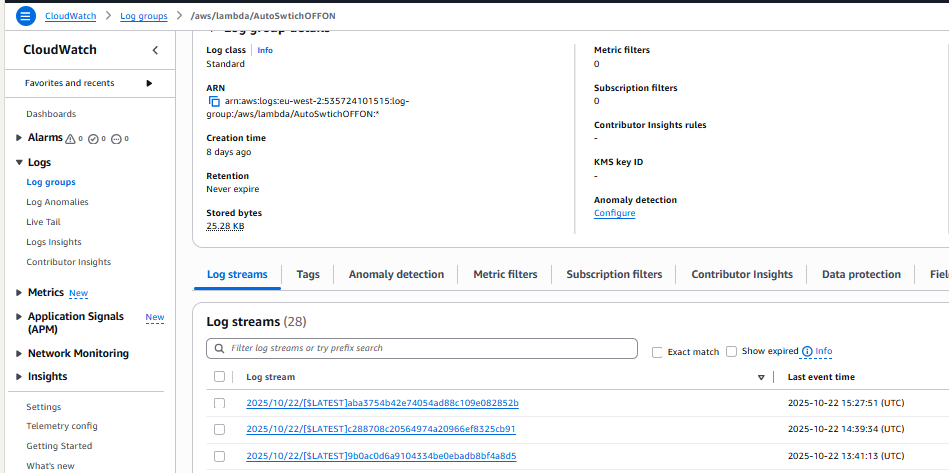
REPORT RequestId: 9a4ba497-f9d3-4542-b7ac-4f26c41ca684  Duration: 1491.63 ms    Billed Duration: 1492 ms    Memory Size: 128 MB Max Memory Used: 97 MB

Request ID: 9a4ba497-f9d3-4542-b7ac-4f26c41ca684

Here in this output it showing that “No instances is to start. But some instances is running. So stop instances rule worked”.

Verify:

Go on CloudWatcher.



Check the latest Event for output confirmation.

