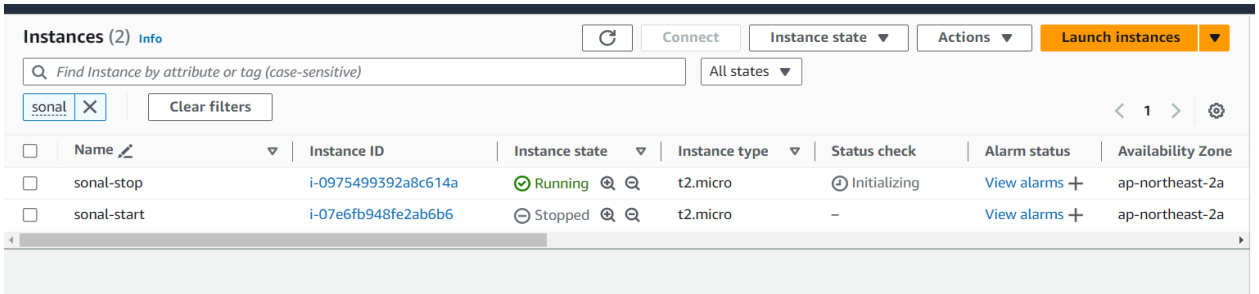
Assignment 1: Automated Instance Management Using AWS Lambda and Boto3

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.

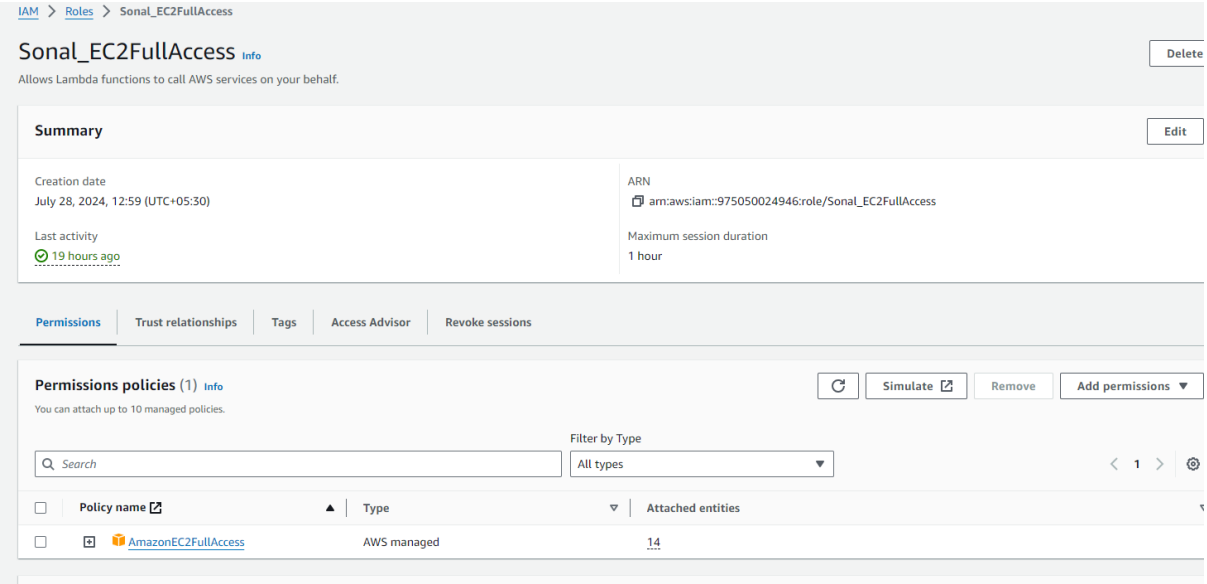
Step 1: EC2 Setup

- Navigate to the EC2 Dashboard.
- Create two instances with t2. micro instance type
- Add tags:
  For the first instance: Key = Action, Value = Auto-Stop
  For the second instance: Key = Action, Value = Auto-Start
- Review and Launch the instances.



Step 2: Lambda IAM Role

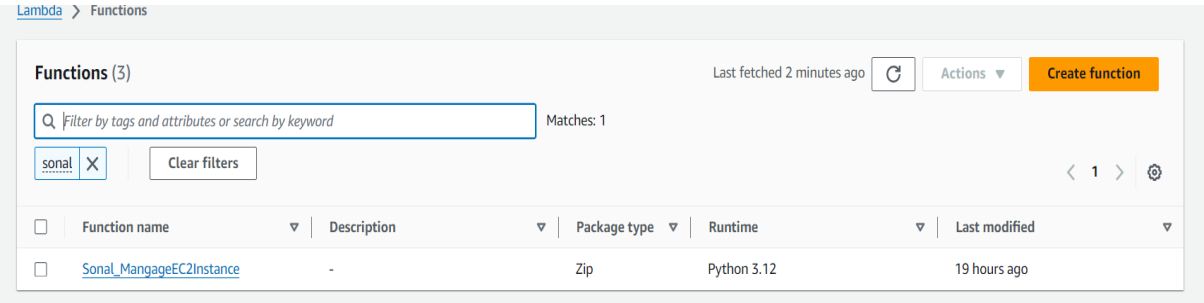
- Navigate to the IAM Dashboard and create new role for Lambda
- Attach the AmazonEC2FullAccess policy to this role.



Step 3: Lambda Function Creation:

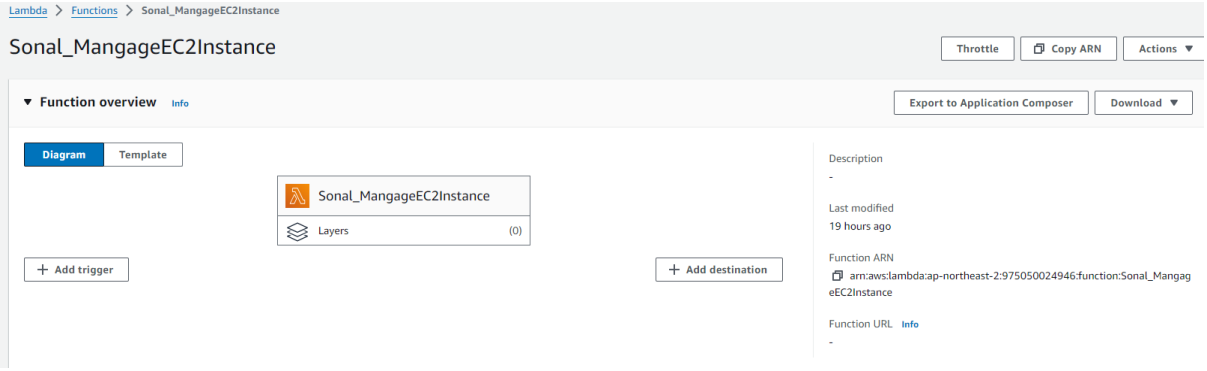
1. **Create Lambda Function:**
   - Navigate to the Lambda Dashboard.

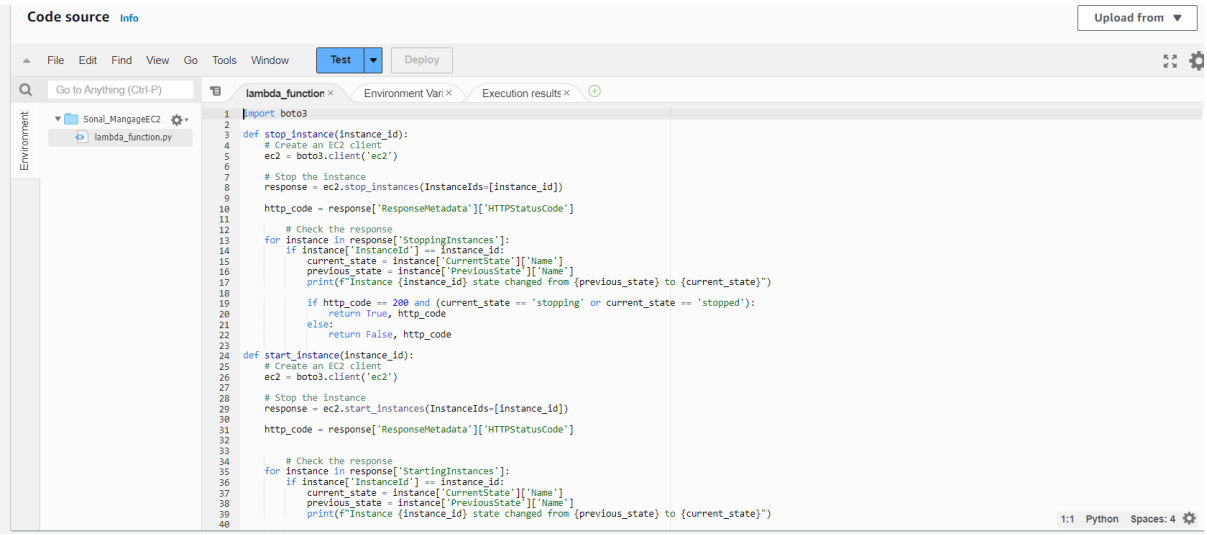
- Click "Create function."
- Select "Author from scratch."
- Name the function
- Choose Python 3.12 as the runtime.
- Under "Permissions," choose "Use an existing role."
- Select the role created in the previous step
- Click "Create function."



2. Write Boto3 Script:

- Initialize a boto3 EC2 client
- Describe instances with ‘Auto start’ and ‘Auto stop’ tags
- Detect all EC2 instances with the `Auto-Stop` tag and stop them
- Detect all EC2 instances with the `Auto-Start` tag and start them.





Python script

Assignment_1 > Automated_InstanceManagement.py > stop_instance

```python
import boto3


def stop_instance(instance_id):
    # Create an EC2 client
    ec2 = boto3.client('ec2')

    # Stop the instance
    response = ec2.stop_instances(InstanceIds=[instance_id])

    http_code = response['ResponseMetadata']['HTTPStatusCode']

        # Check the response
    for instance in response['StoppingInstances']:
        if instance['InstanceId'] == instance_id:
            current_state = instance['CurrentState']['Name']
            previous_state = instance['PreviousState']['Name']
            print(f"Instance {instance_id} state changed from {previous_state} to {current_state}")

            if http_code == 200 and (current_state == 'stopping' or current_state == 'stopped'):
                return True, http_code
            else:
                return False, http_code

def start_instance(instance_id):
    # Create an EC2 client
    ec2 = boto3.client('ec2')

    # Stop the instance
    response = ec2.start_instances(InstanceIds=[instance_id])

    http_code = response['ResponseMetadata']['HTTPStatusCode']

    print(response)

        # Check the response
    for instance in response['StartingInstances']:
        if instance['InstanceId'] == instance_id:
            current_state = instance['CurrentState']['Name']
```

Ln 8, Col 24 Spaces: 4 UTF-8

3. **Save the function:**
   - Click on "Deploy" to save the changes.

Step 4: Manual Execution

1. Execute the Lambda Function:

   - Navigate to the Lambda function dashboard.
   - Click on "Test."
   - Configure a test event (you can use the default test event template).
   - Click on "Create."
   - Click on "Test" again to manually execute the function.

2. Verify EC2 Instance State:

- Navigate to the EC2 Dashboard.
- Check the state of the instances.
- The instance tagged Auto-Stop should be in the "stopping" or "stopped" state.
- The instance tagged Auto-Start should be in the "pending" or "running" state.