**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.

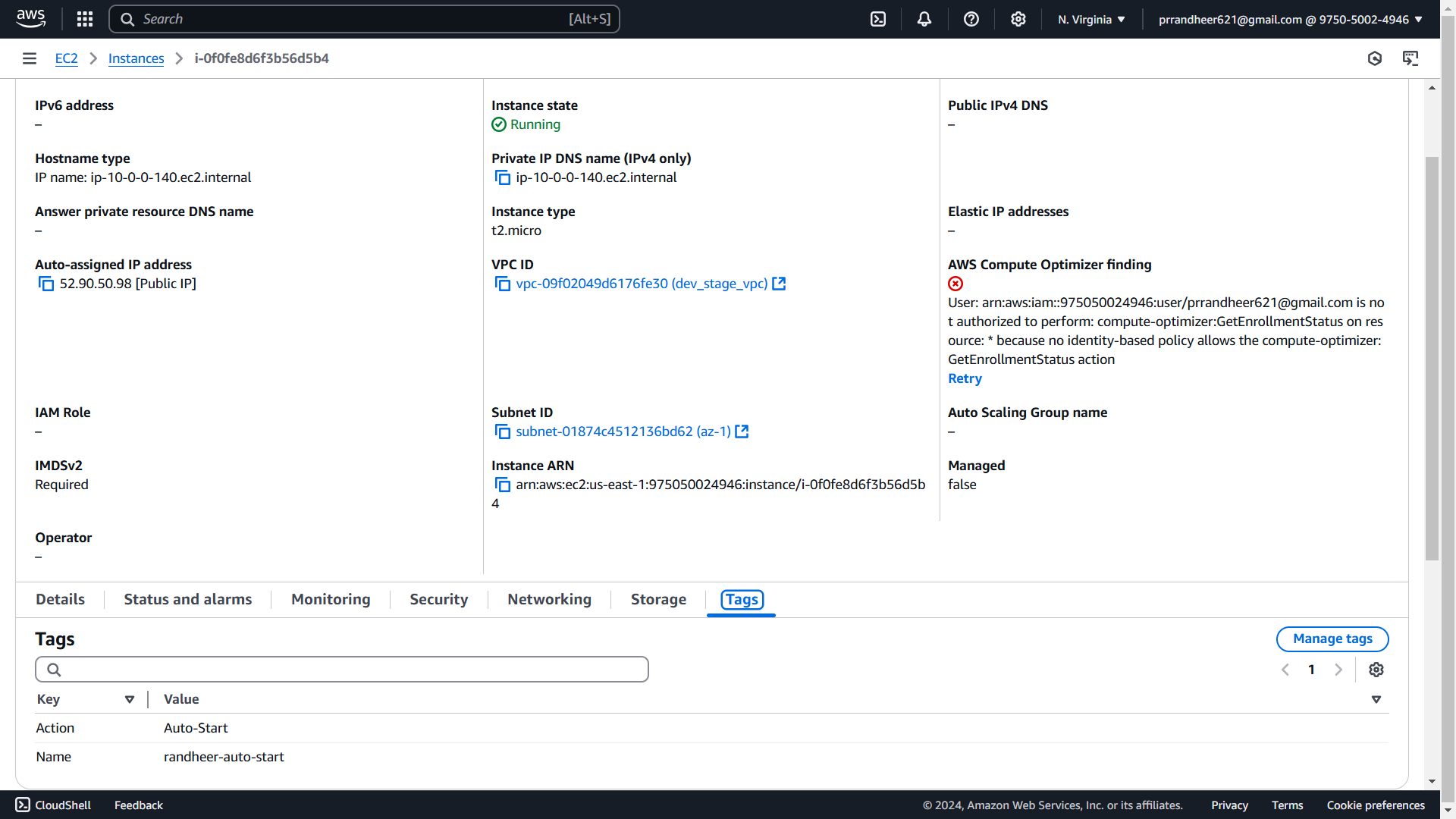
     4. Print instance IDs that were affected for logging purposes.

4. Manual Invocation:

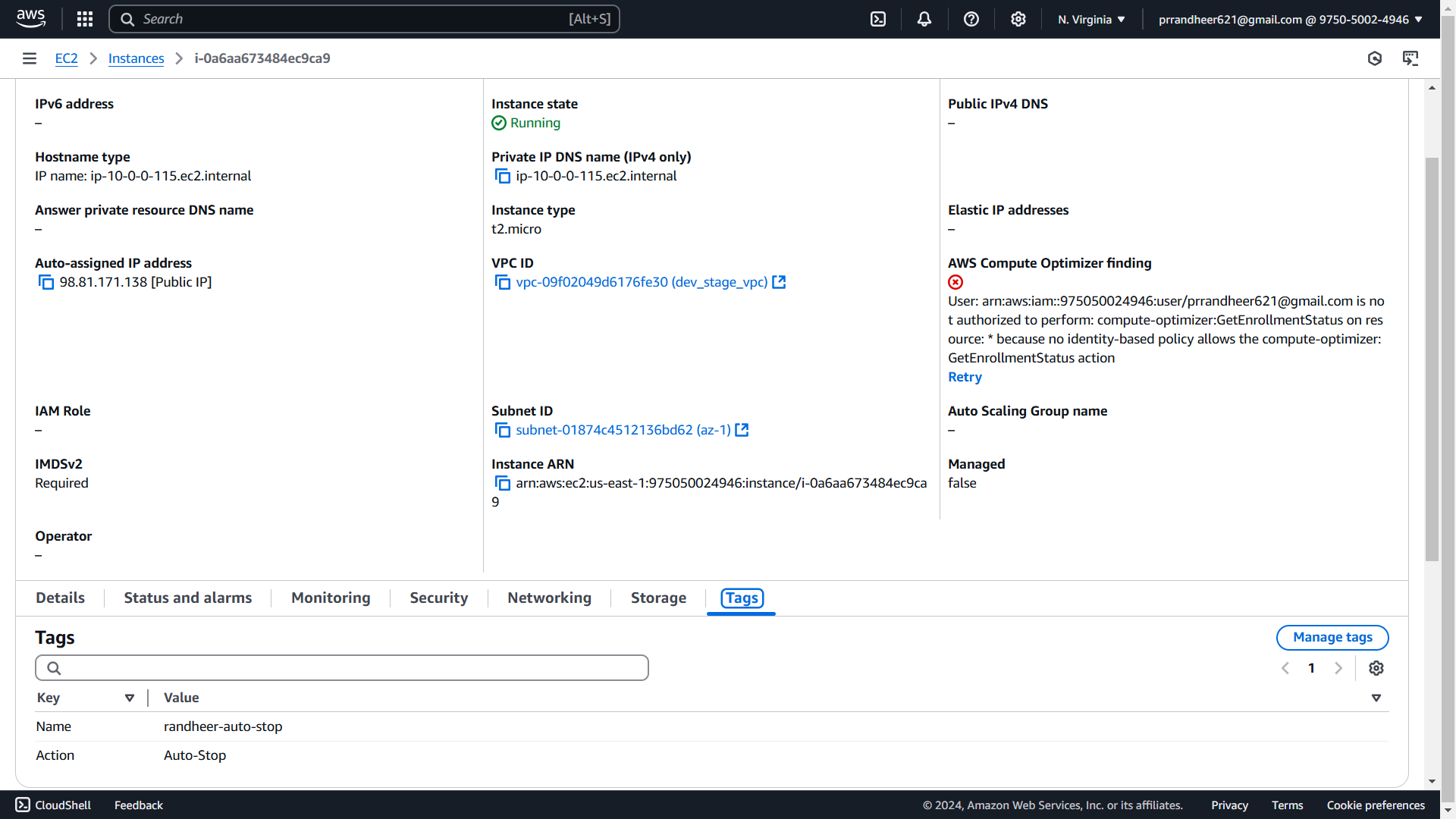
   - After saving your function, manually trigger it.

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

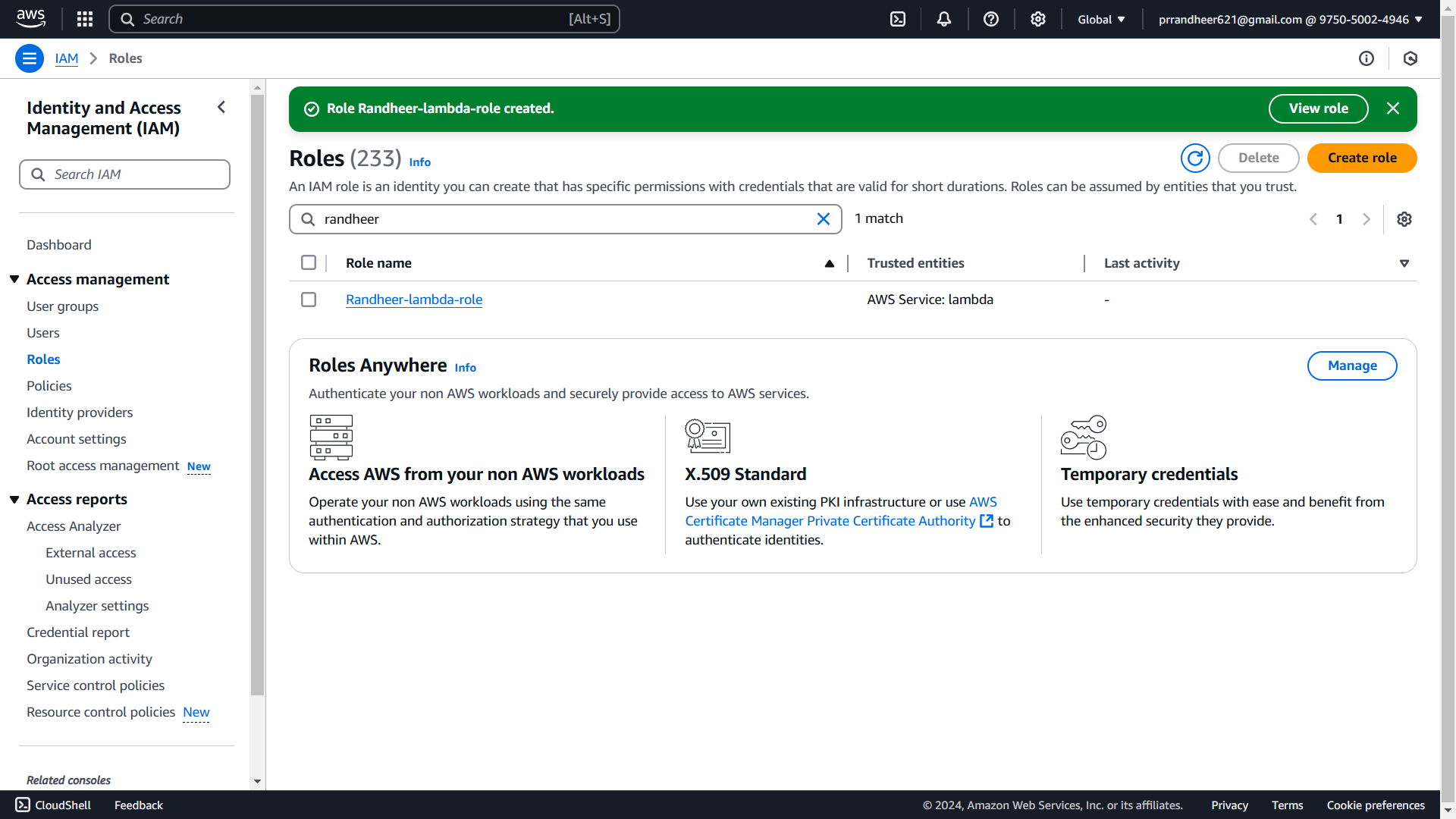
**Instance with Auto-Start tag**



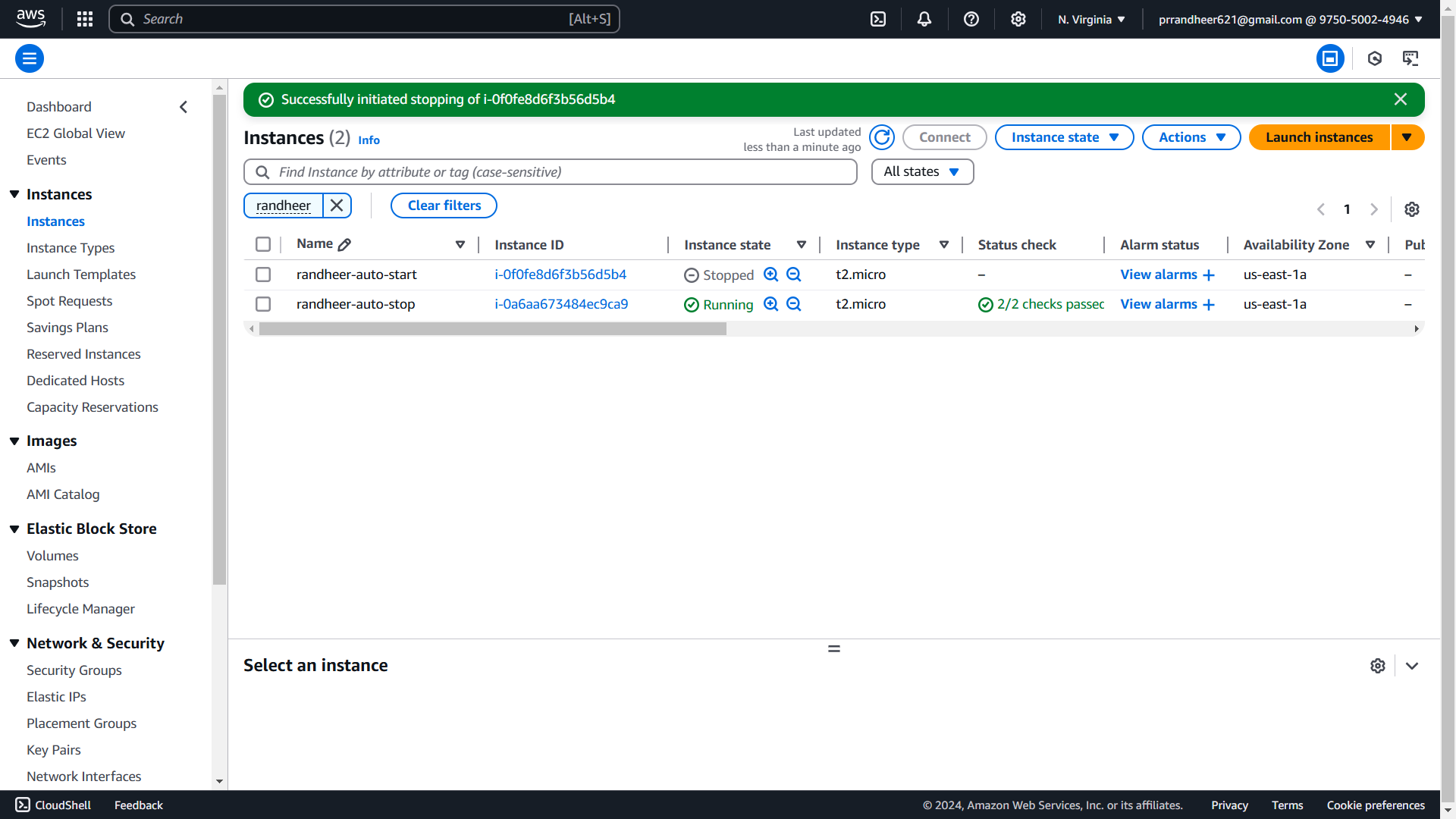
**Instance with Auto-Stop tag**



Created IAM Lambda role with **AmazonEC2FullAccess**



Initial State



**#Lambda Function to trigger instances Auto Start and Auto Stop**

import boto3

def lambda\_handler(event, context):

ec2\_client = boto3.client('ec2')

# Describe instances with specific tags

response = ec2\_client.describe\_instances(

Filters=[

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

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

]

)

# Stop instances with Auto-Stop tag

auto\_stop\_instances = [

instance['InstanceId']

for reservation in response['Reservations']

for instance in reservation['Instances']

]

if auto\_stop\_instances:

ec2\_client.stop\_instances(InstanceIds=auto\_stop\_instances)

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

# Describe instances with Auto-Start tag

response = ec2\_client.describe\_instances(

Filters=[

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

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

]

)

# Start instances with Auto-Start tag

auto\_start\_instances = [

instance['InstanceId']

for reservation in response['Reservations']

for instance in reservation['Instances']

]

if auto\_start\_instances:

ec2\_client.start\_instances(InstanceIds=auto\_start\_instances)

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

return {

'statusCode': 200,

'body': 'EC2 instances managed successfully!'

}

**After running Lambda Function**

