**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.

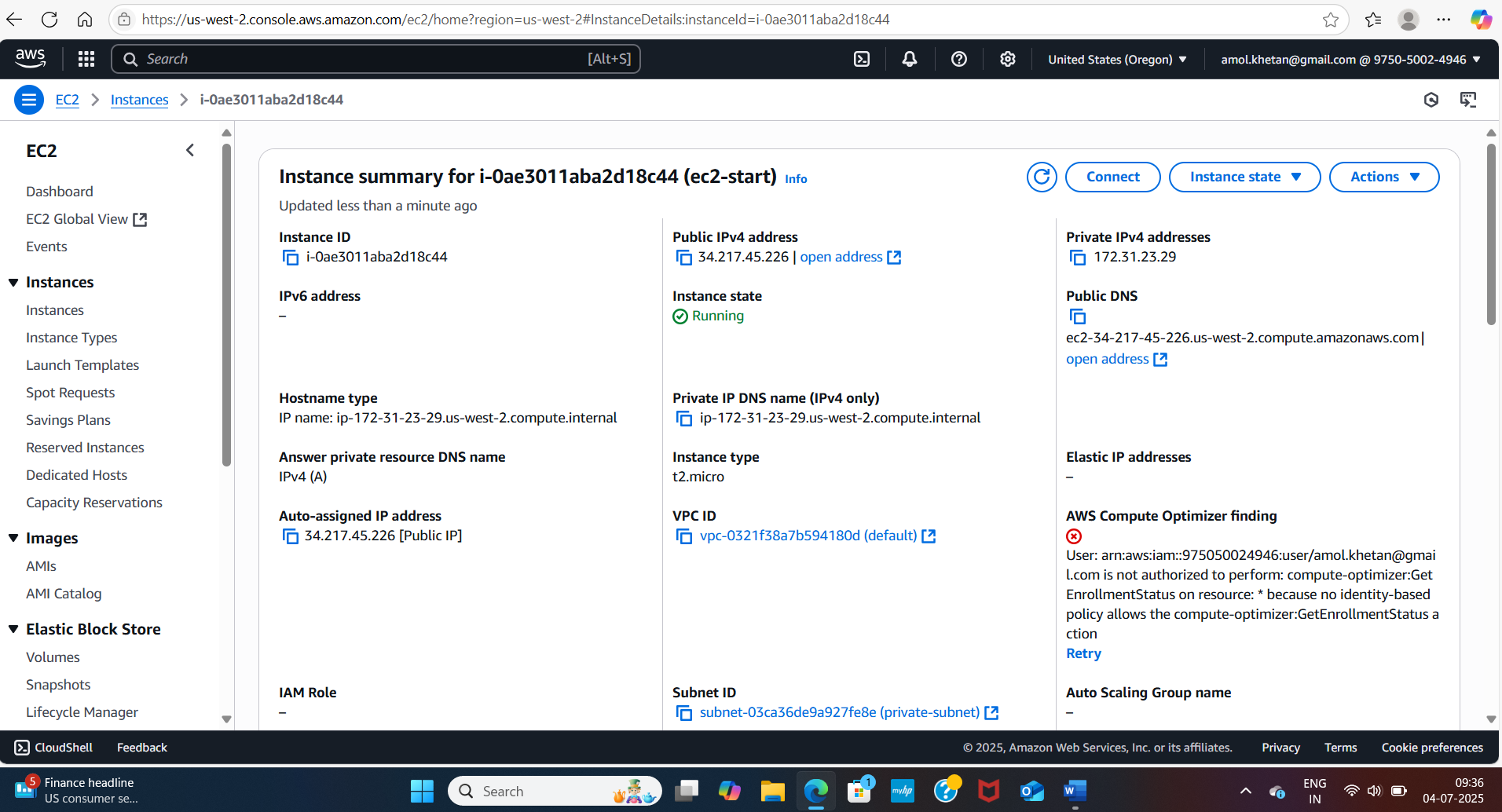
**Task:**You're tasked to automate the stopping and starting of EC2 instances based on tags. Specifically:

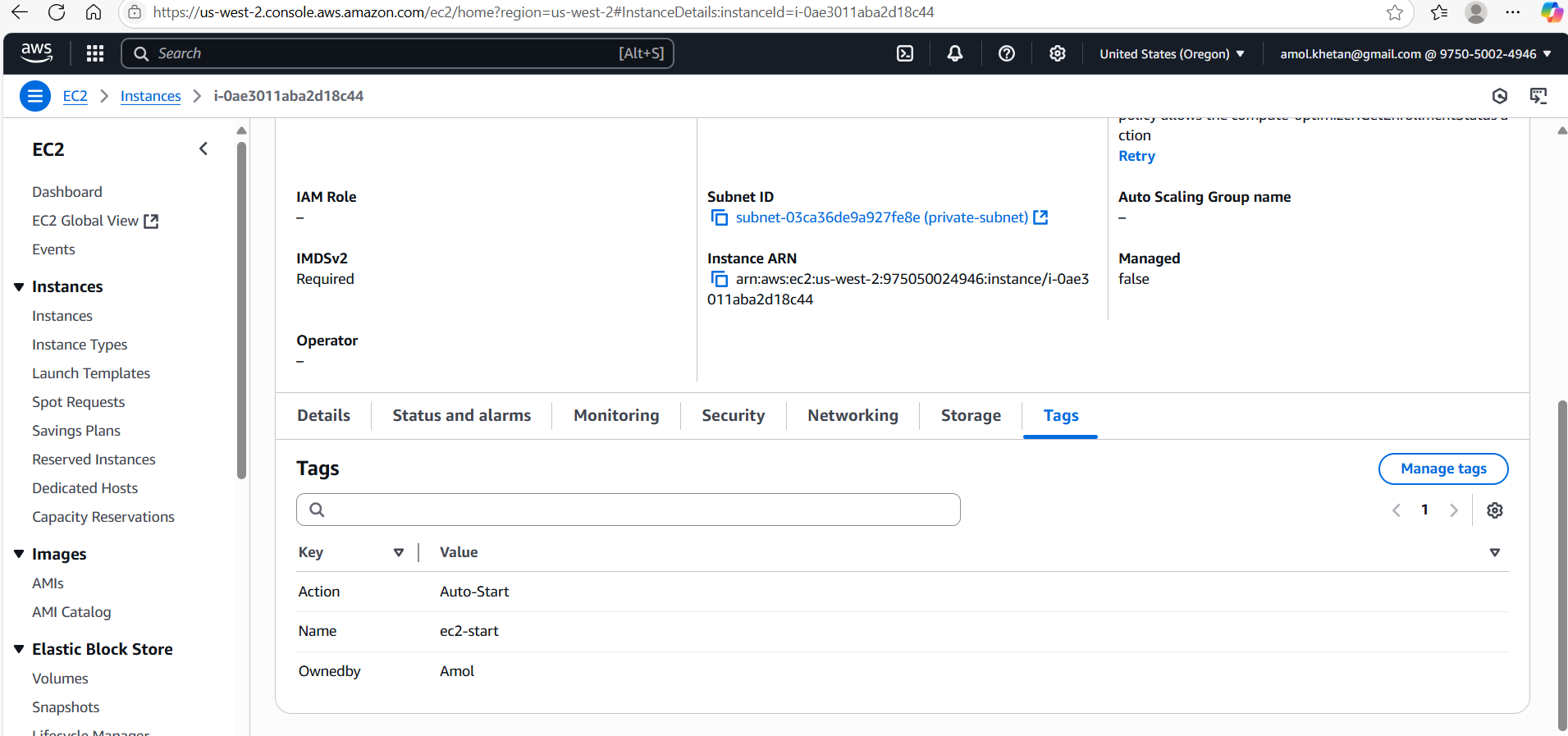
**1. Setup:**

   - Create two EC2 instances.

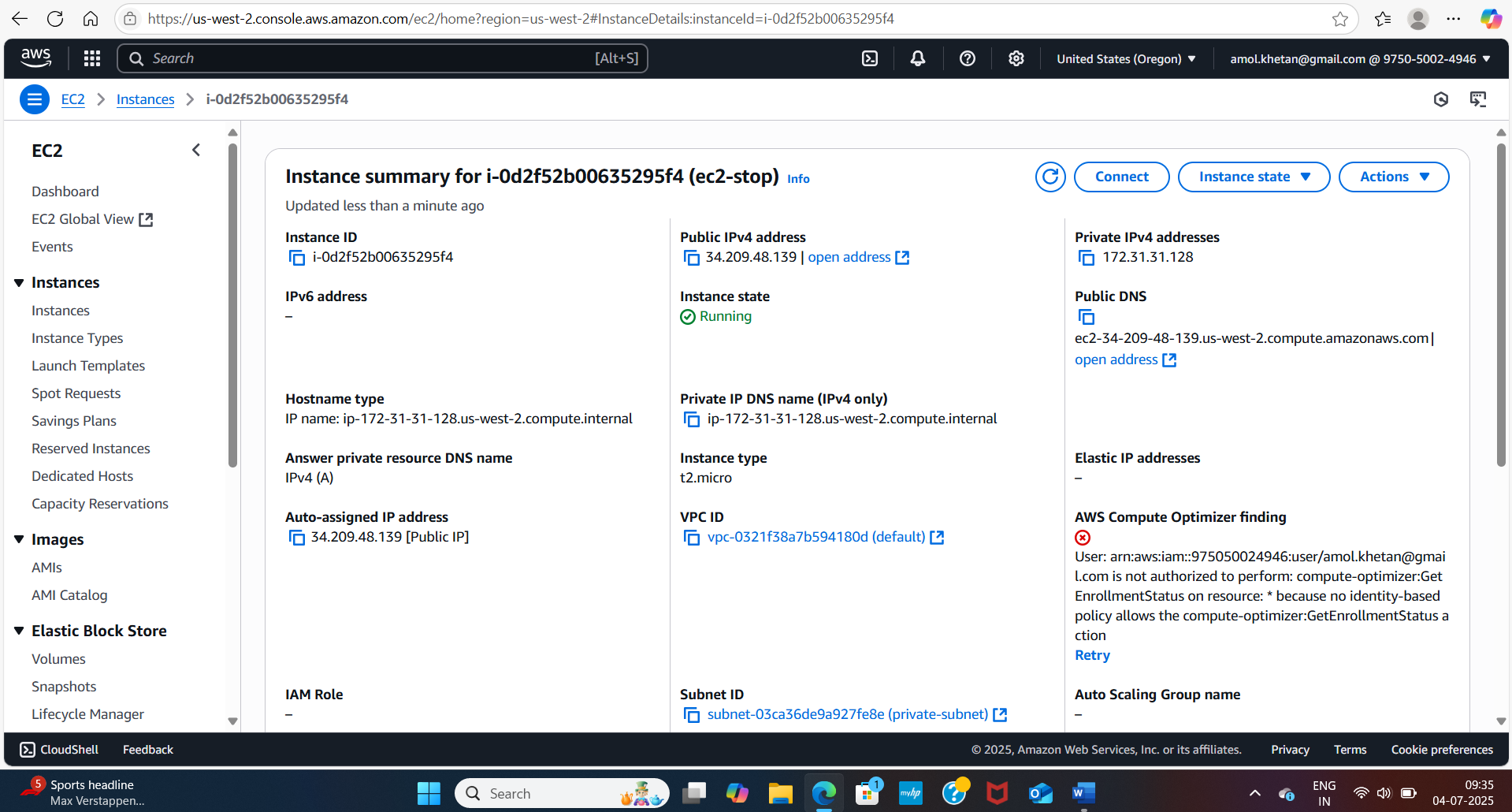
   - Tag one of them as `Auto-Stop` and the other as `Auto-Start`.

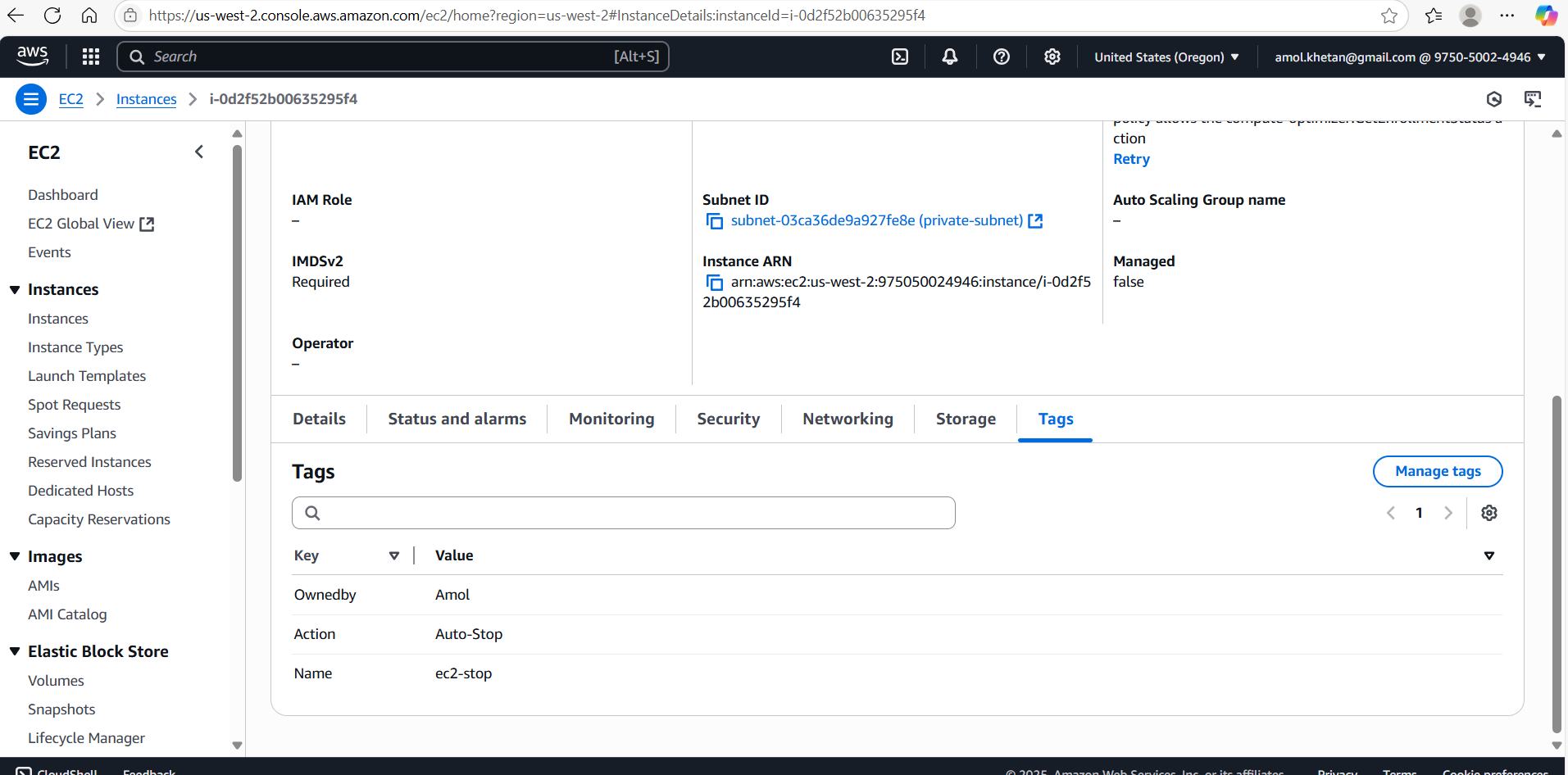
Start Instance Configs:





Stop Instance Configs:

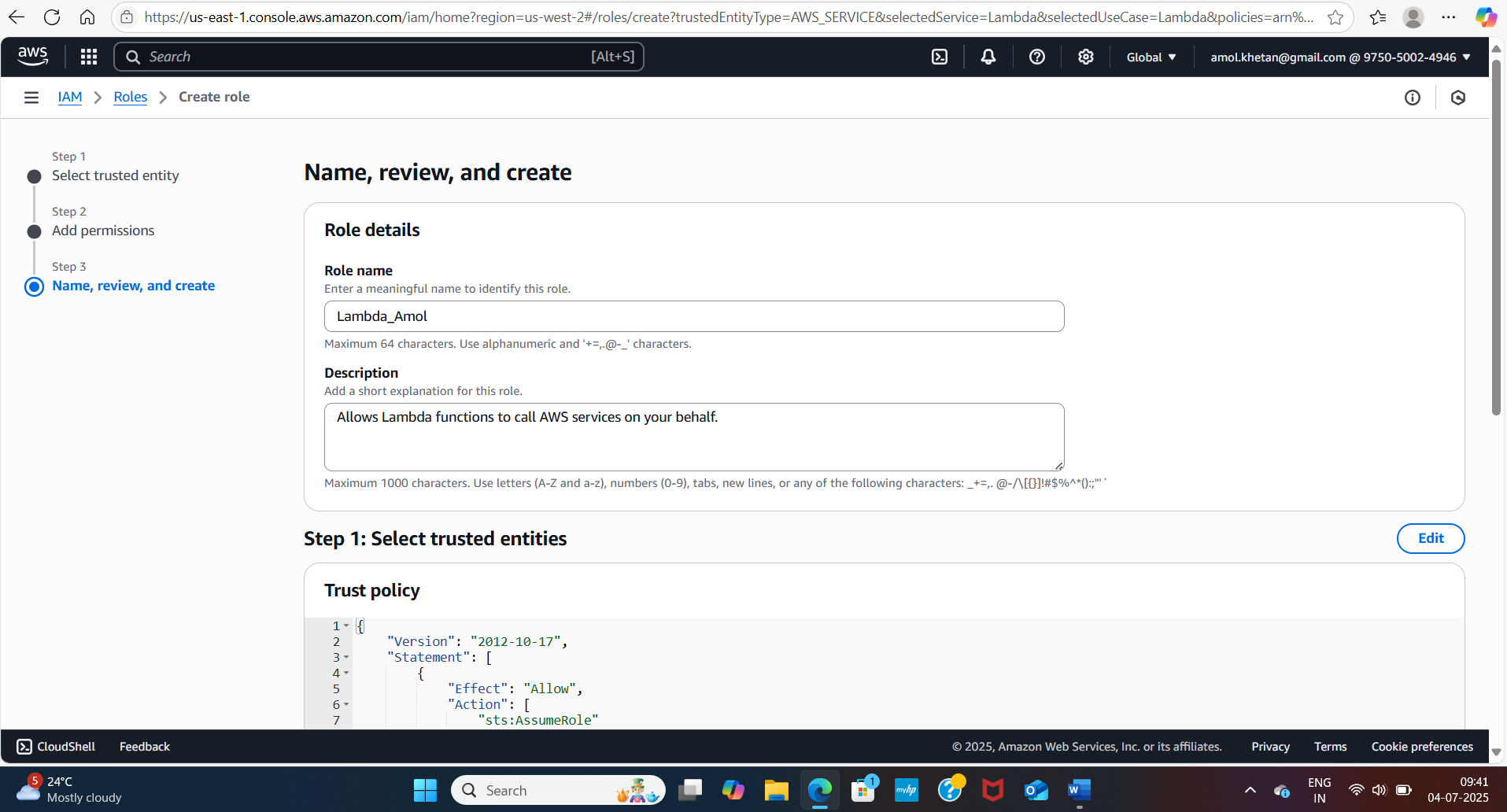


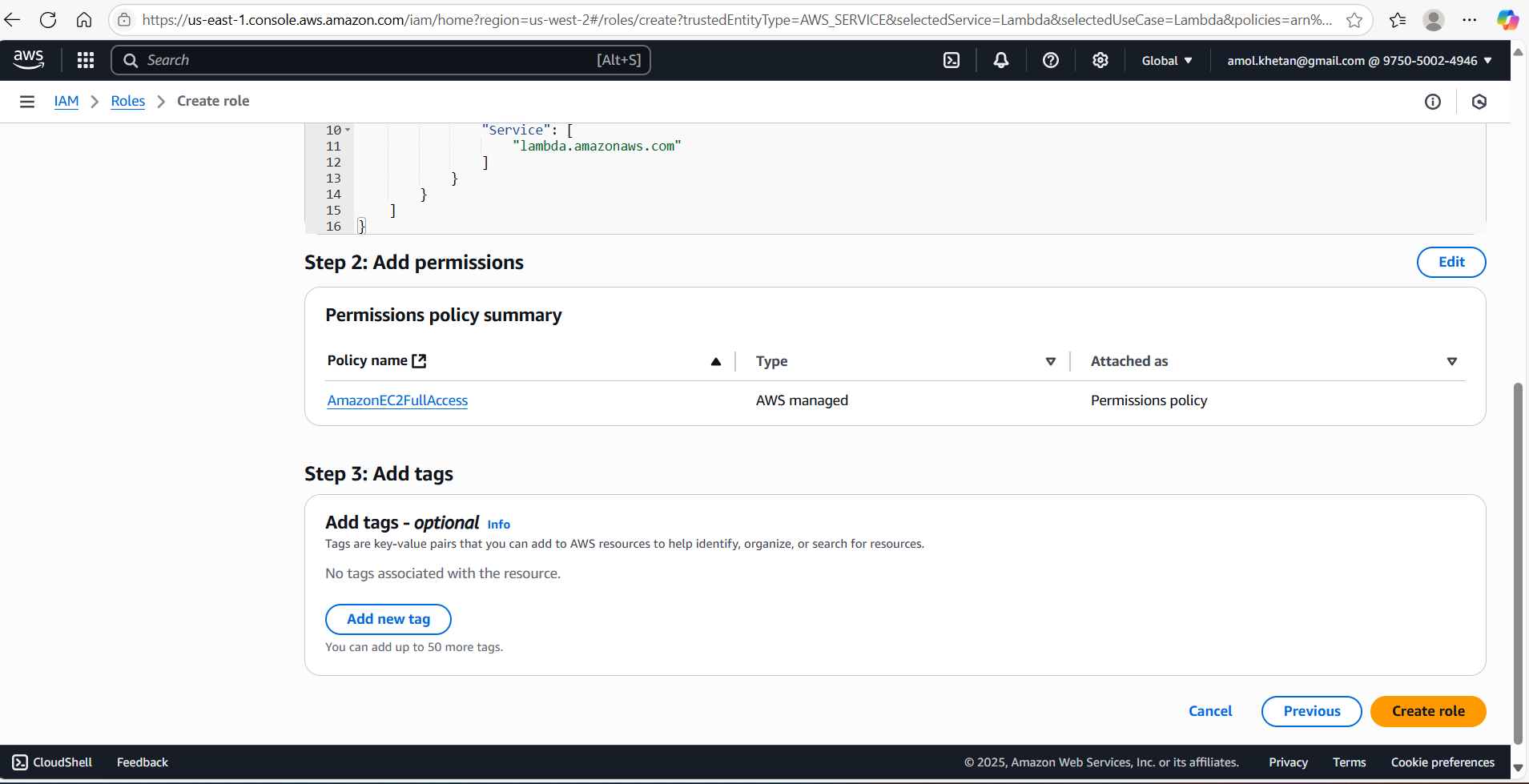


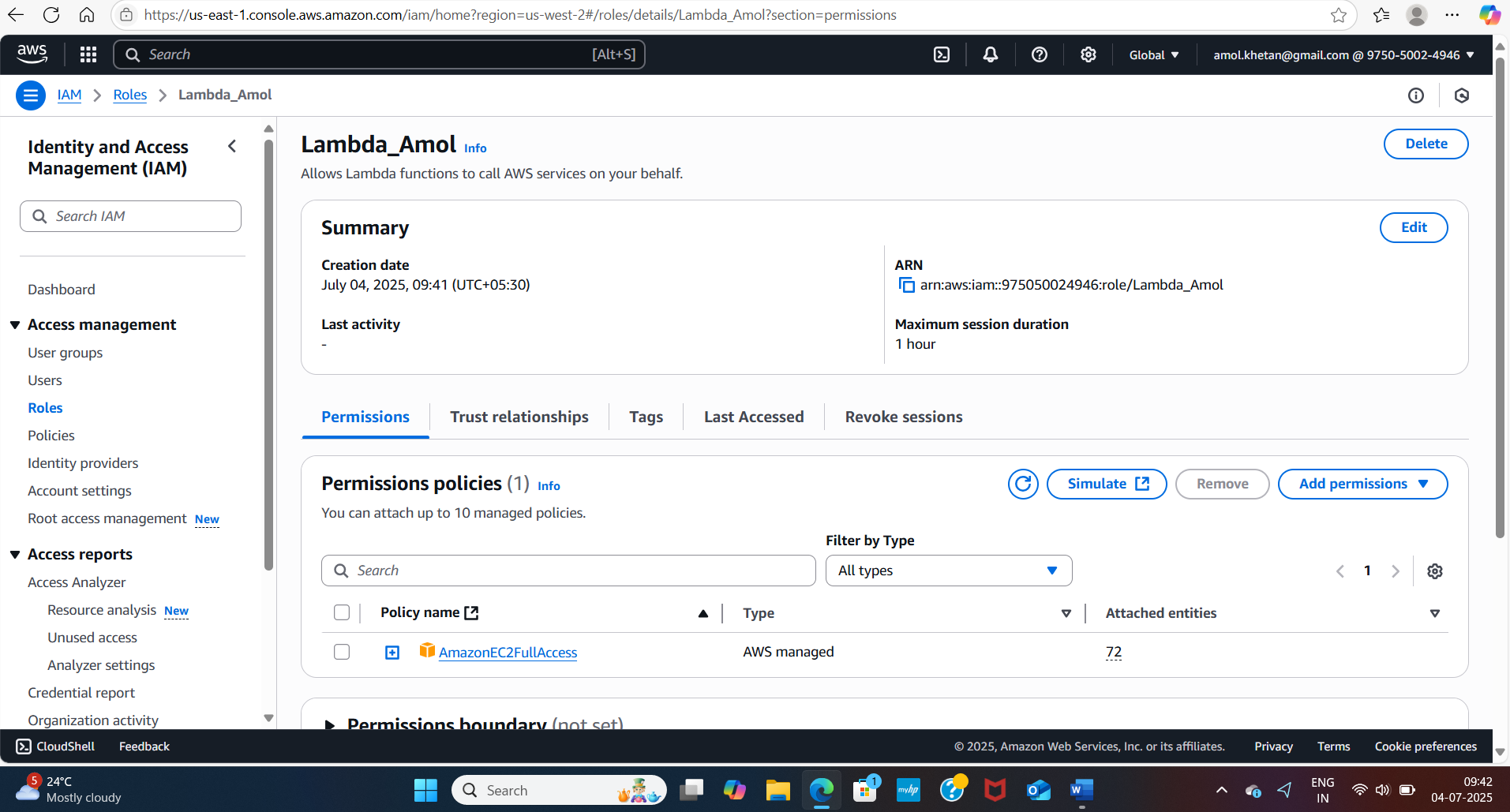
**2. Lambda IAM Role:**

   - 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.)



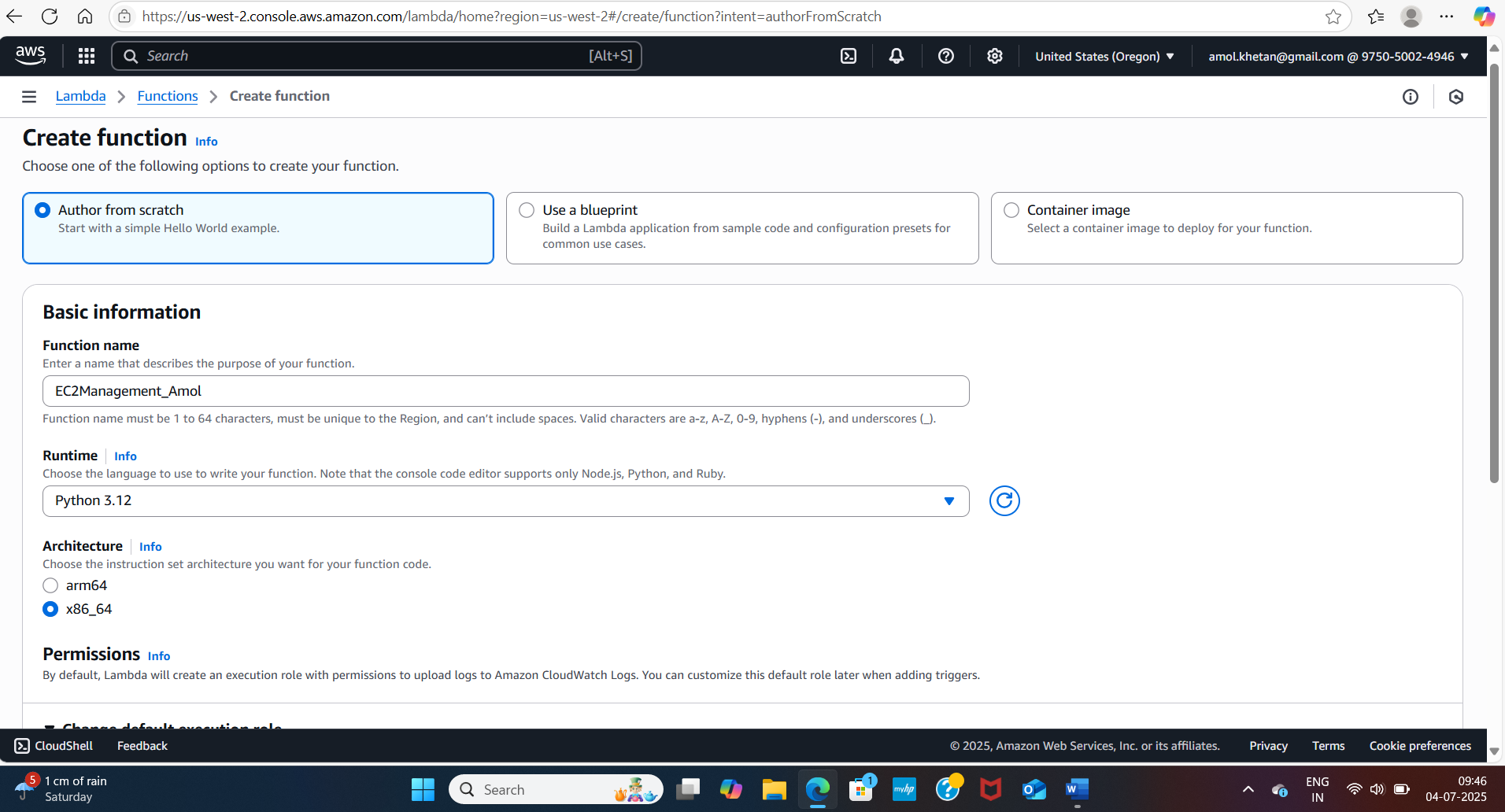


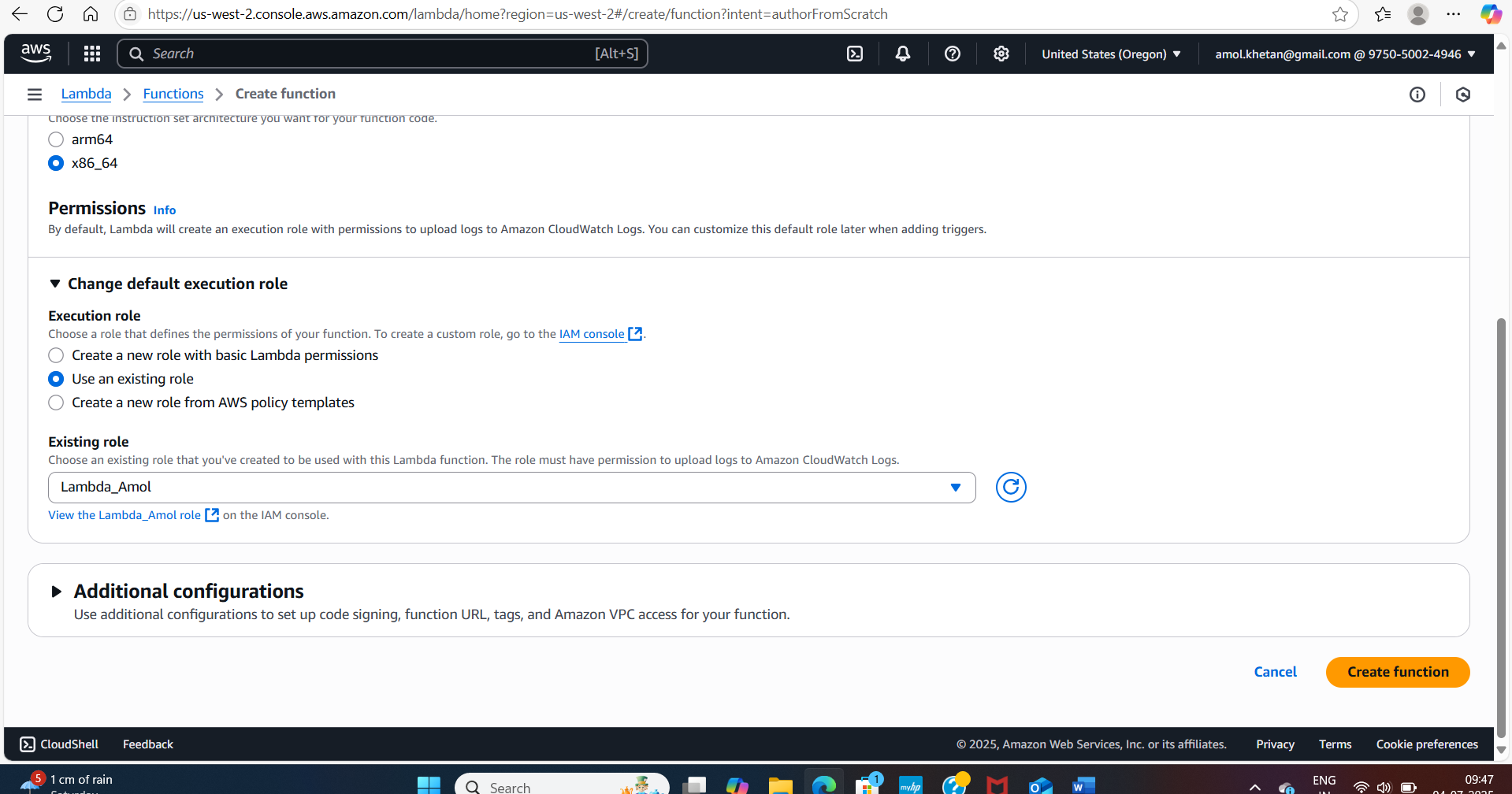


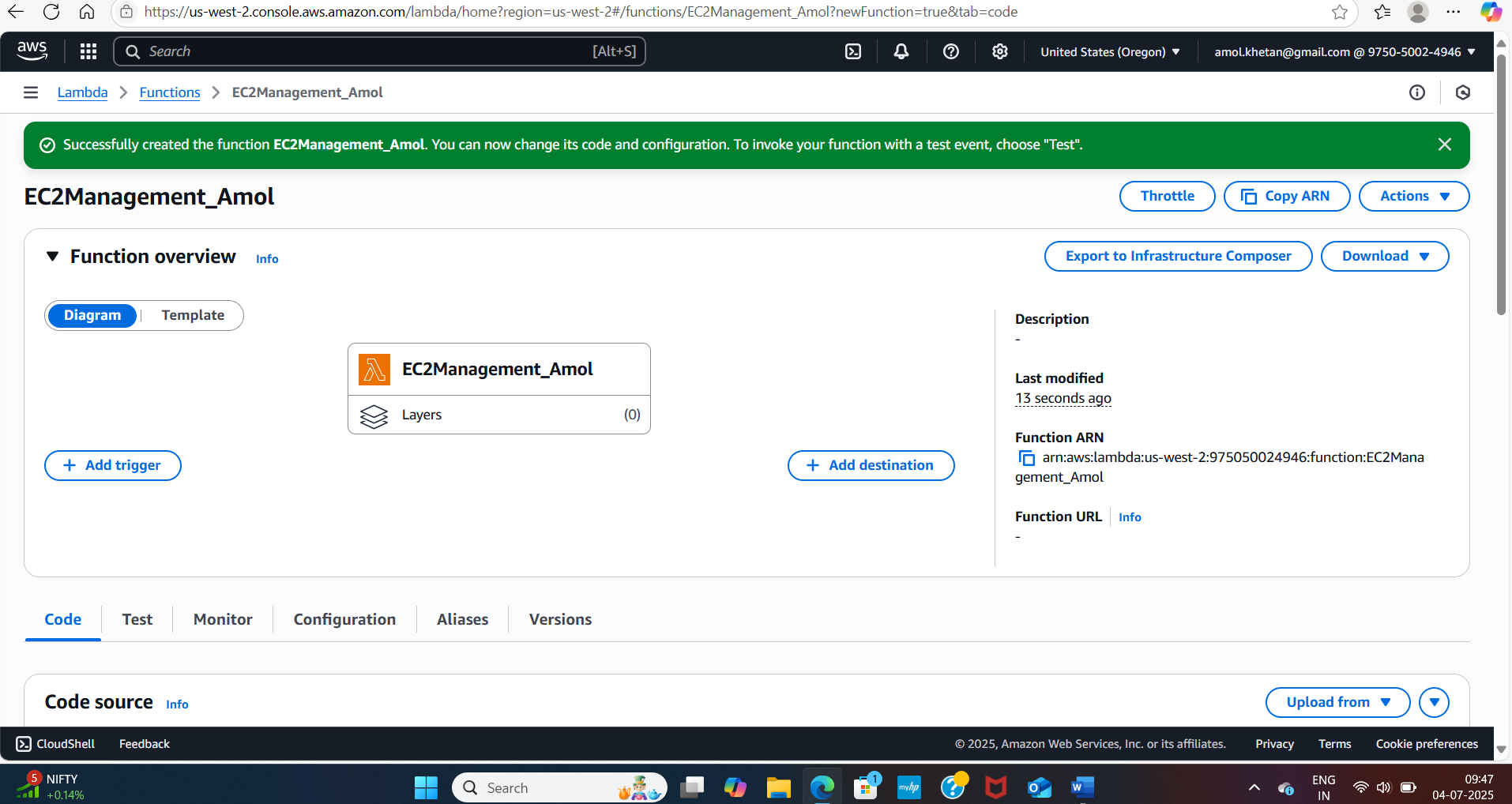
**3. Lambda Function Creation:**

 - Set up an AWS Lambda function.

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





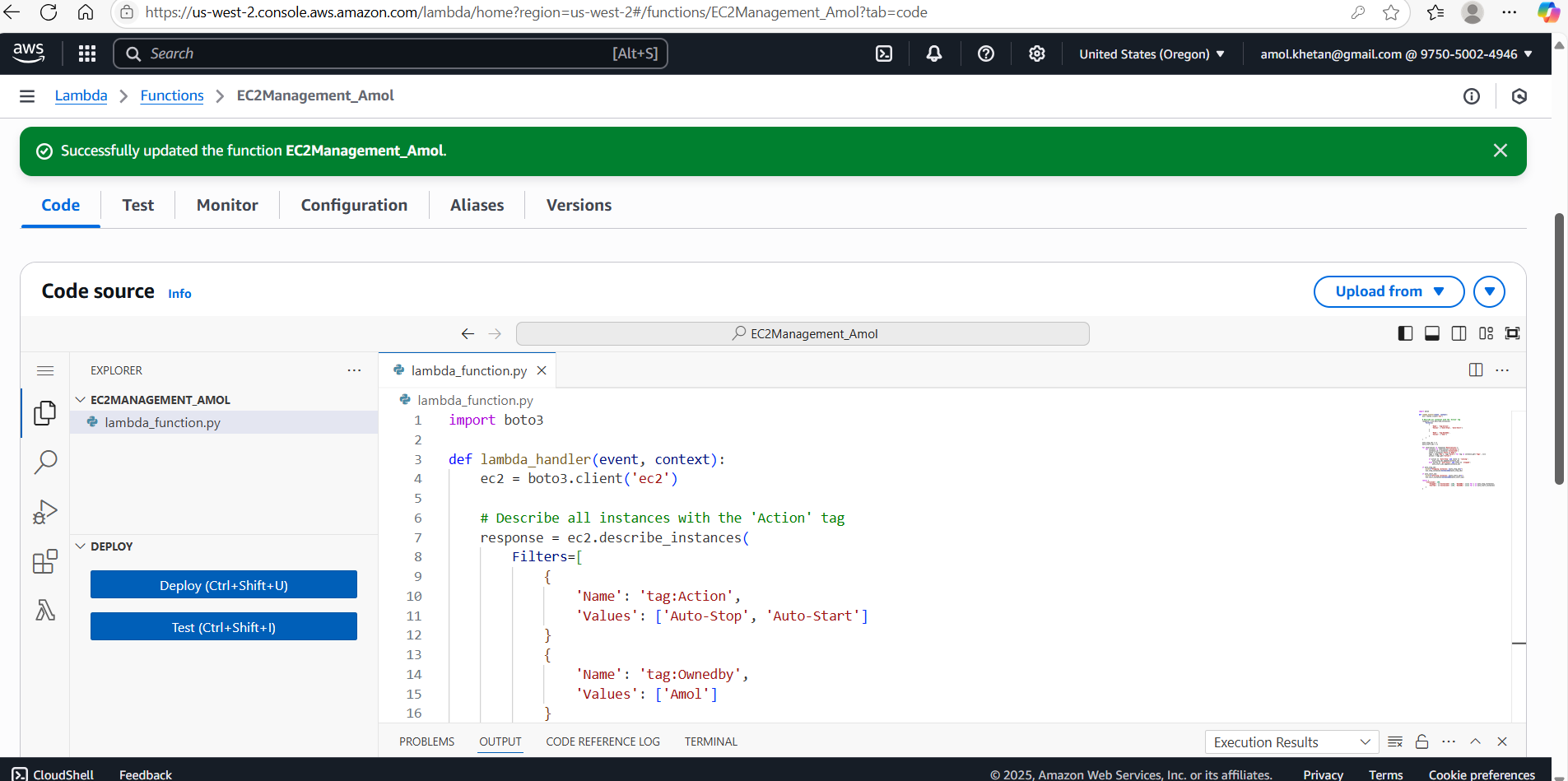


**5. Coding:**

   - Using Boto3 in the Lambda function:

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

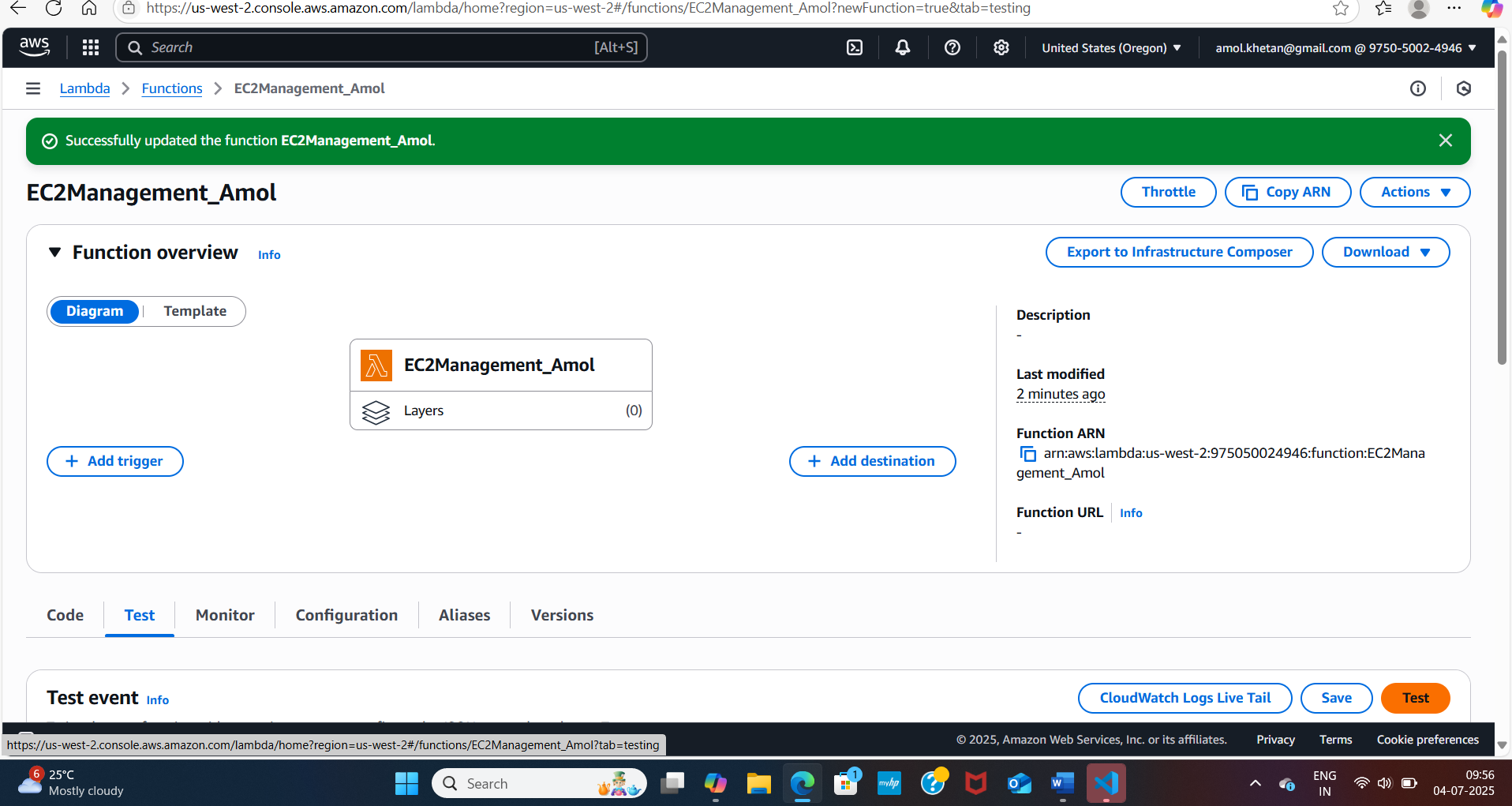
     - Detect all EC2 instances with the `Auto-Start` tag and start them.



**5. Testing:**

 - Manually invoke the Lambda function.

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

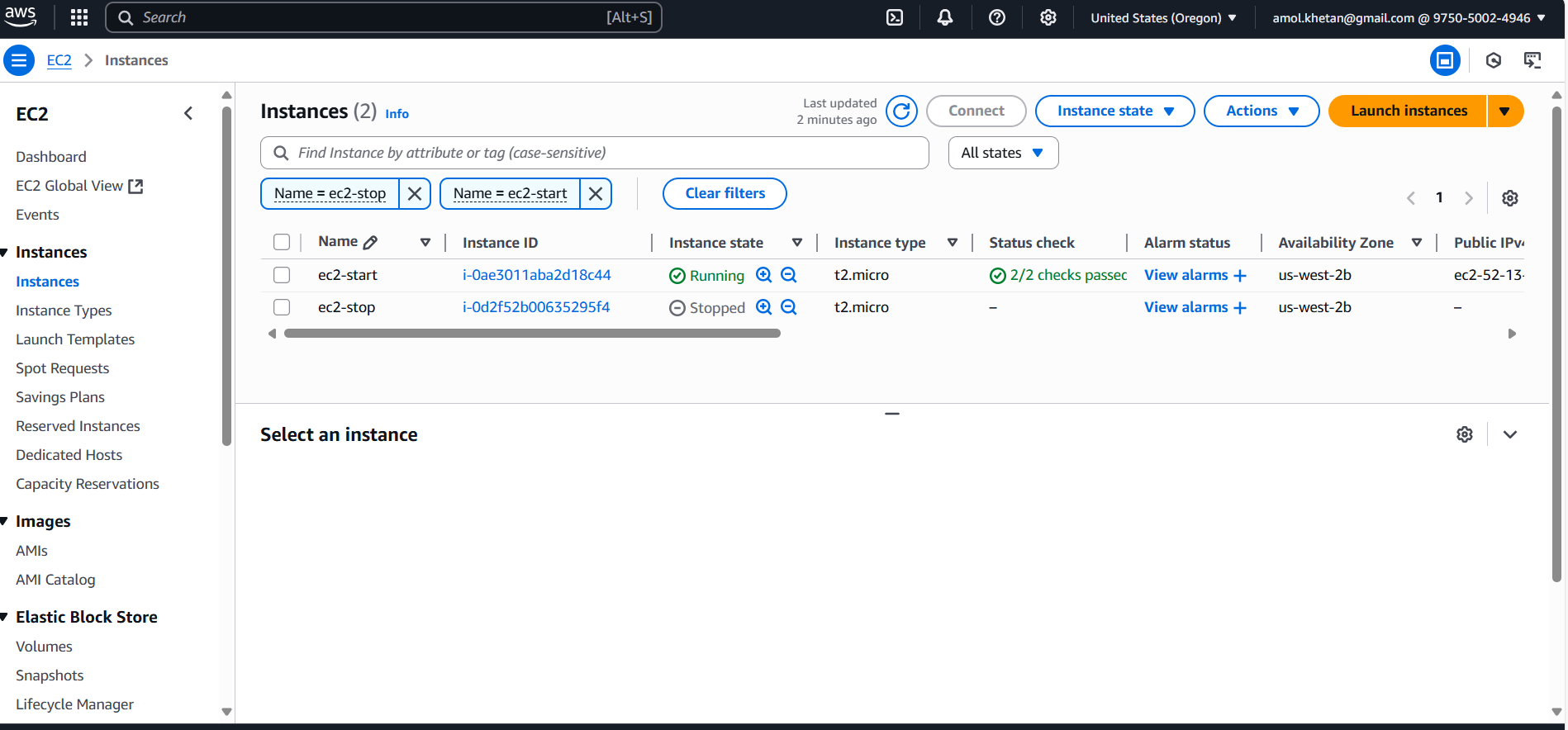


In 1st test it failed as timeout was 3 sec only, changed it to 30 sec and it worked

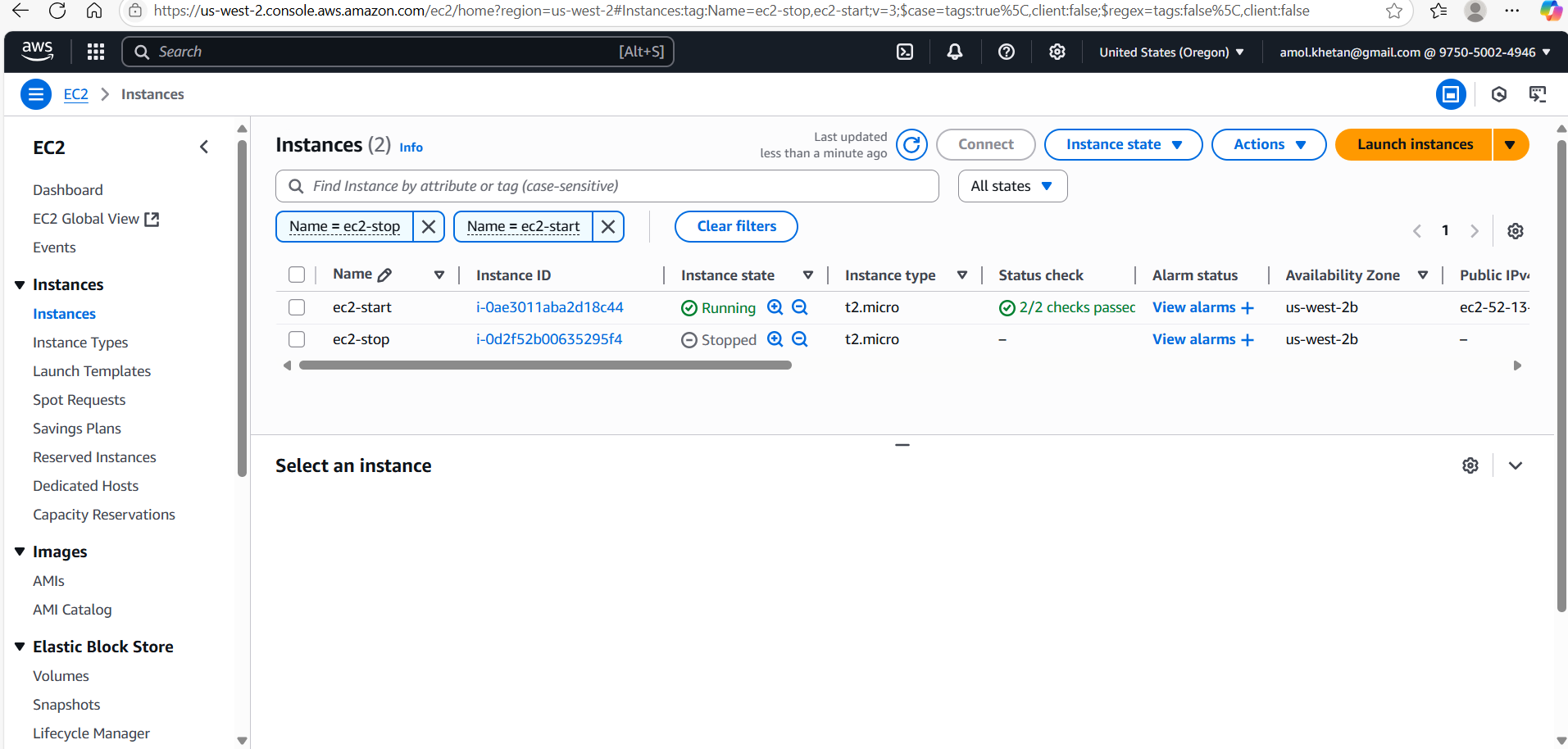
**Test Scenario 1: Stop Instance is Stopped and Start is in Running State.**

Expected Result: Nothing Should Happen

Screenshot before testing:



Post Testing:



Lambda Function Output:

Status: Succeeded

Test Event Name: testevent

Response:

{

  "statusCode": 200,

  "body": {

    "Stopped": [],

    "Started": []

  }

}

Function Logs:

START RequestId: 5cb5e10e-2d20-4028-8091-e638976e22d1 Version: $LATEST

END RequestId: 5cb5e10e-2d20-4028-8091-e638976e22d1

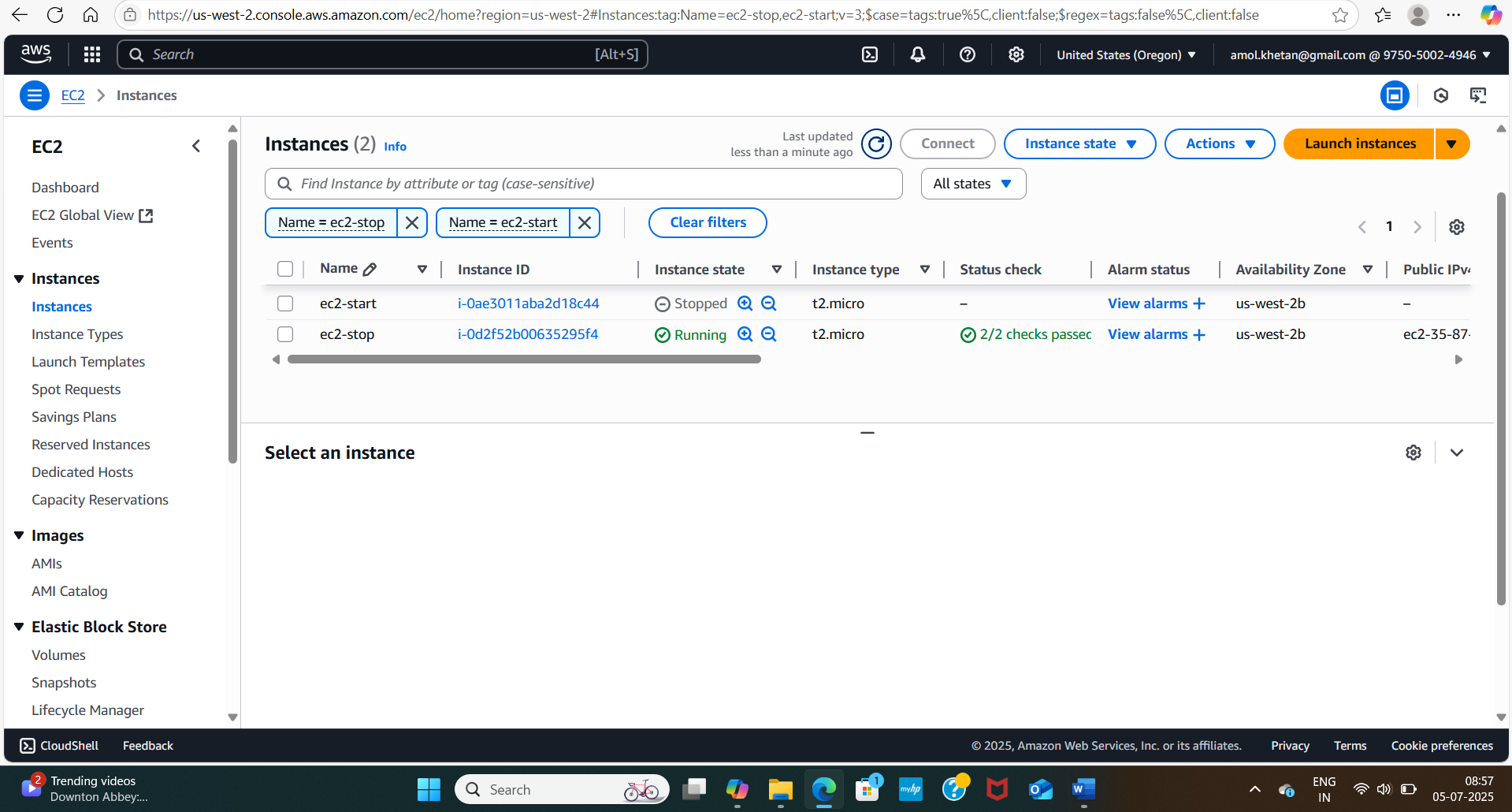
REPORT RequestId: 5cb5e10e-2d20-4028-8091-e638976e22d1  Duration: 3622.92 ms    Billed Duration: 3623 ms    Memory Size: 128 MB Max Memory Used: 94 MB  Init Duration: 279.73 ms

Request ID: 5cb5e10e-2d20-4028-8091-e638976e22d1

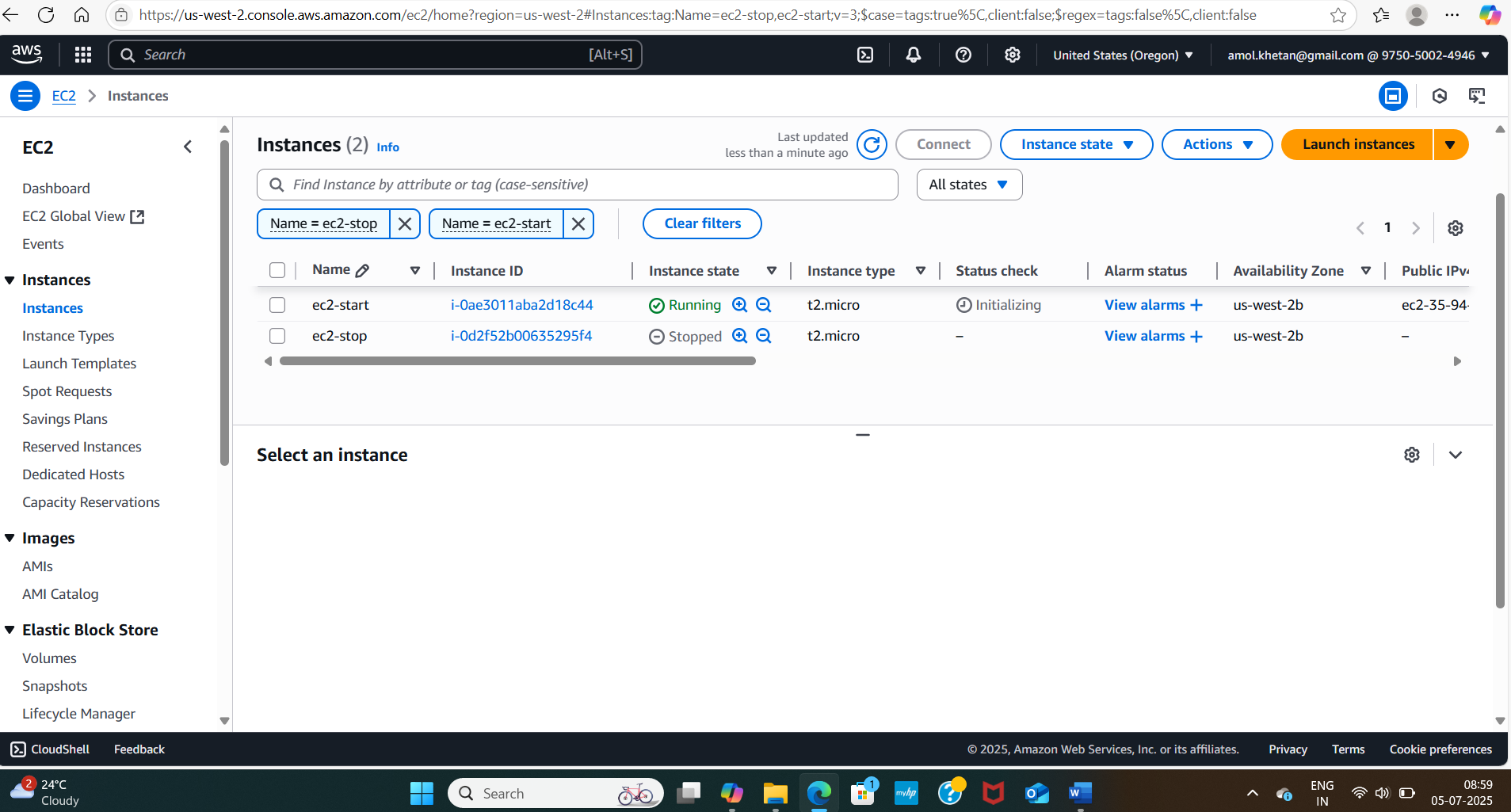
**Test Scenario 2: Stop Instance is Running and Start is in Stop State.**

Expected Result: Stop Instance should Stop and Start one should Start Running

Screenshot before testing:



After Lambda Invocation



Lambda Function Output:

Status: Succeeded

Test Event Name: testevent

Response:

{

  "statusCode": 200,

  "body": {

    "Stopped": [

      {

        "InstanceId": "i",

        "OwnedBy": "-"

      }

    ],

    "Started": [

      {

        "InstanceId": "i",

        "OwnedBy": "-"

      }

    ]

  }

}

Function Logs:

START RequestId: 19b22db4-7dad-4f8e-8848-19a504eba9fa Version: $LATEST

Stopping instances: ['i-0d2f52b00635295f4']

Starting instances: ['i-0ae3011aba2d18c44']

END RequestId: 19b22db4-7dad-4f8e-8848-19a504eba9fa

REPORT RequestId: 19b22db4-7dad-4f8e-8848-19a504eba9fa  Duration: 4470.13 ms    Billed Duration: 4471 ms    Memory Size: 128 MB Max Memory Used: 96 MB  Init Duration: 281.65 ms

Request ID: 19b22db4-7dad-4f8e-8848-19a504eba9fa