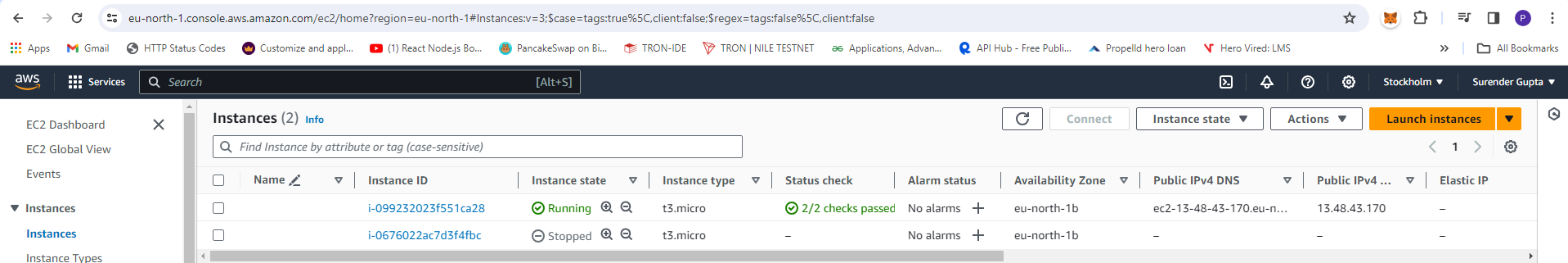
**Assignment On Serverless Architecture**

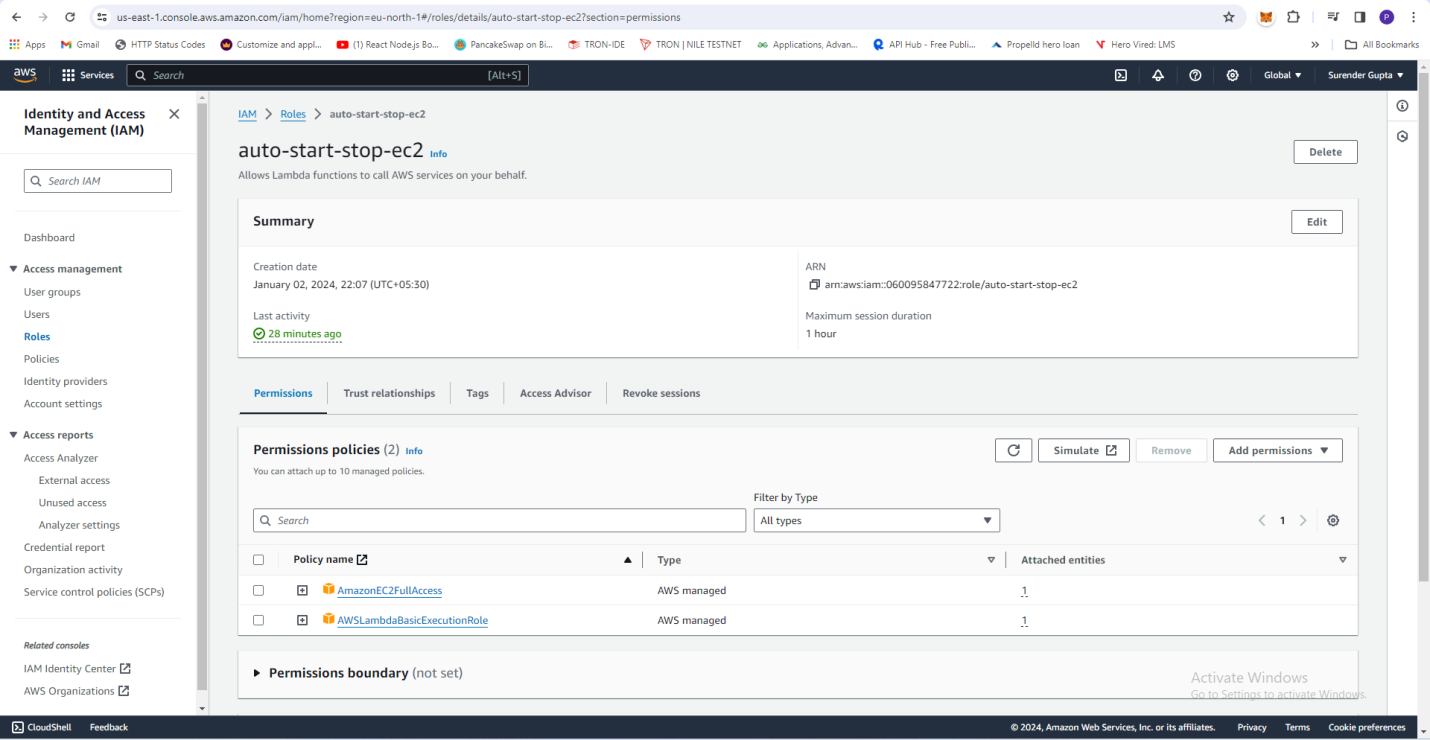
**Assignment 1: Automated Instance Management Using AWS Lambda and Boto3**

**Solution1.**

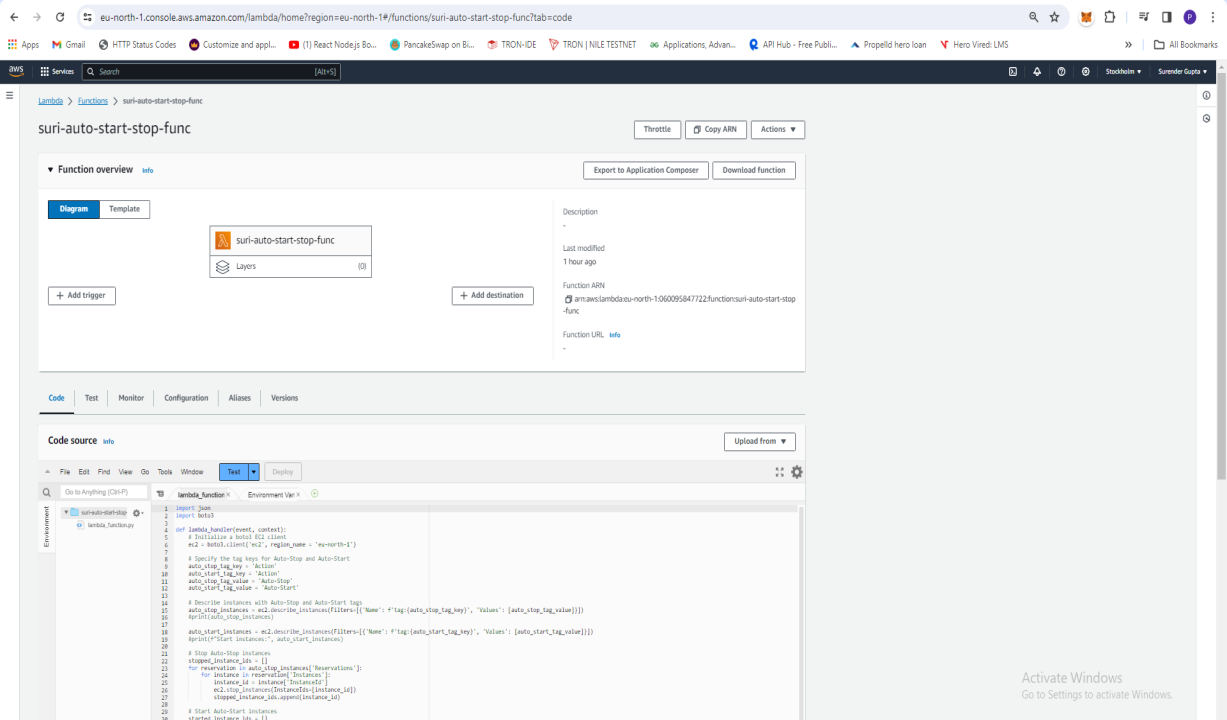
1. First Create Two EC2 Instance
2. Steps are below for creating EC2 Instance.
3. In Section of **Name and tags** set **Key** is **Action** and **Value** is **Auto-Start**
4. In Section of **Application and OS Images (Amazon Machine Image)** select **Quick Start tab** and **Ubuntu** option after select **Amazon Machine Image (AMI)** select any option as required but we are select **Ubuntu Server 22.04 LTS (HVM), SSD Volume Type** now **Architecture** option select **64-Bit (x86)**
5. In Section of **Instance type** select option as you need but we are choose **t3.micro**
6. In Section of **Key pair (login)** if you not have any key pair create new key pair but we have already created key pair just selecting option of my **auto-start-stop**
7. In Section of **Network settings** you are **just create new security group** or **existing security group** select as you need but we are **create security group** also check **Allow SSH traffic from** option of IP Address but we are selecting **Anywhere 0.0.0.0/0** also checkbox tick on **Allow HTTPS traffic from the internet** and **Allow HTTP traffic from the internet**
8. In Section of **Configure storage** and **Advanced details** do not need to set anything required in this project.
9. In Section of **Summary** if required set **Number of instances** as much you required type but we set **2** for now
10. Now click on **Launch instance** button.
11. After Success message , click on instance-id in message
12. Back to Instance list page edit tag name **Auto-Start** change to **Auto-Stop**.
13. Two Instances are Screenshot below:



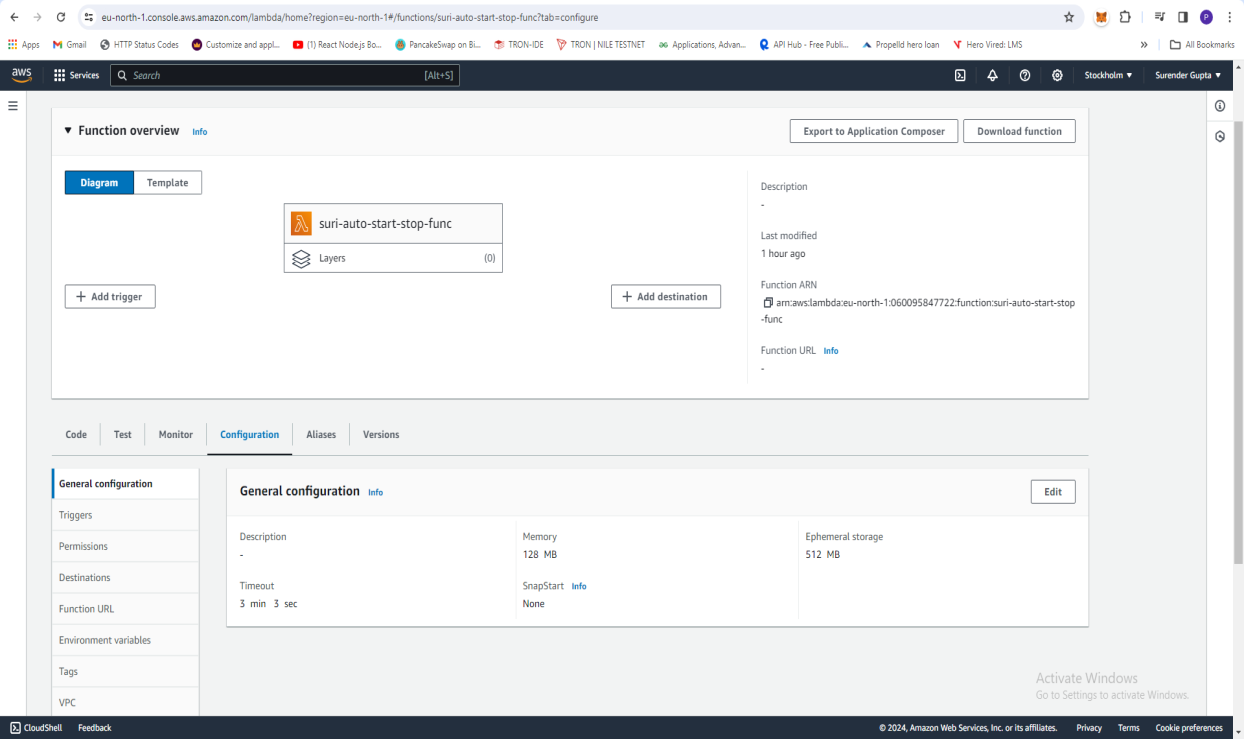
1. Create New Role for Lambda Function
2. First go to **IAM Dashboard** left sidebar under **Access Management -> Role** click it.
3. Create new Role button then **Trusted entity type** AWS Service and Use Case Lambda and click on Next Button
4. Search permission policy **AmazonEC2FullAccess** Checked it and **AWSLambdaBasicExecutionRole** also checked it click on Next
5. Enter Role Name **auto-start-stop-ec2** and click on create role button at bottom.



1. Create Lambda Function
2. Lambda Dashboard left sidebar **Function** click it
3. Click on **Create function** after choose **Author from Scratch**
4. Function Name **suri-auto-start-stop-func** and Runtime select **Python 3.8** and Architecture **x86\_64** selected.
5. **Change default execution role** Opted **Use an existing role** and select **Existing role** is previously created **auto-start-stop-ec2** and click on **create function** at bottom.



1. Going to Configuration Tab Edit Timeout to 3 second to add Minute.

****

1. Lambda function below:

import json

import boto3

def lambda\_handler(event, context):

# Initialize a boto3 EC2 client

ec2 = boto3.client('ec2', region\_name = 'eu-north-1')

# Specify the tag keys for Auto-Stop and Auto-Start

auto\_stop\_tag\_key = 'Action'

auto\_start\_tag\_key = 'Action'

auto\_stop\_tag\_value = 'Auto-Stop'

auto\_start\_tag\_value = 'Auto-Start'

# Describe instances with Auto-Stop and Auto-Start tags

auto\_stop\_instances = ec2.describe\_instances(Filters=[{'Name': f'tag:{auto\_stop\_tag\_key}', 'Values': [auto\_stop\_tag\_value]}])

auto\_start\_instances = ec2.describe\_instances(Filters=[{'Name': f'tag:{auto\_start\_tag\_key}', 'Values': [auto\_start\_tag\_value]}])

# Stop Auto-Stop instances

stopped\_instance\_ids = []

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

for instance in reservation['Instances']:

instance\_id = instance['InstanceId']

ec2.stop\_instances(InstanceIds=[instance\_id])

stopped\_instance\_ids.append(instance\_id)

# Start Auto-Start instances

started\_instance\_ids = []

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

for instance in reservation['Instances']:

instance\_id = instance['InstanceId']

ec2.start\_instances(InstanceIds=[instance\_id])

started\_instance\_ids.append(instance\_id)

# Print affected instance IDs for logging purposes

print("Stopped instances:", stopped\_instance\_ids)

print("Started instances:", started\_instance\_ids)

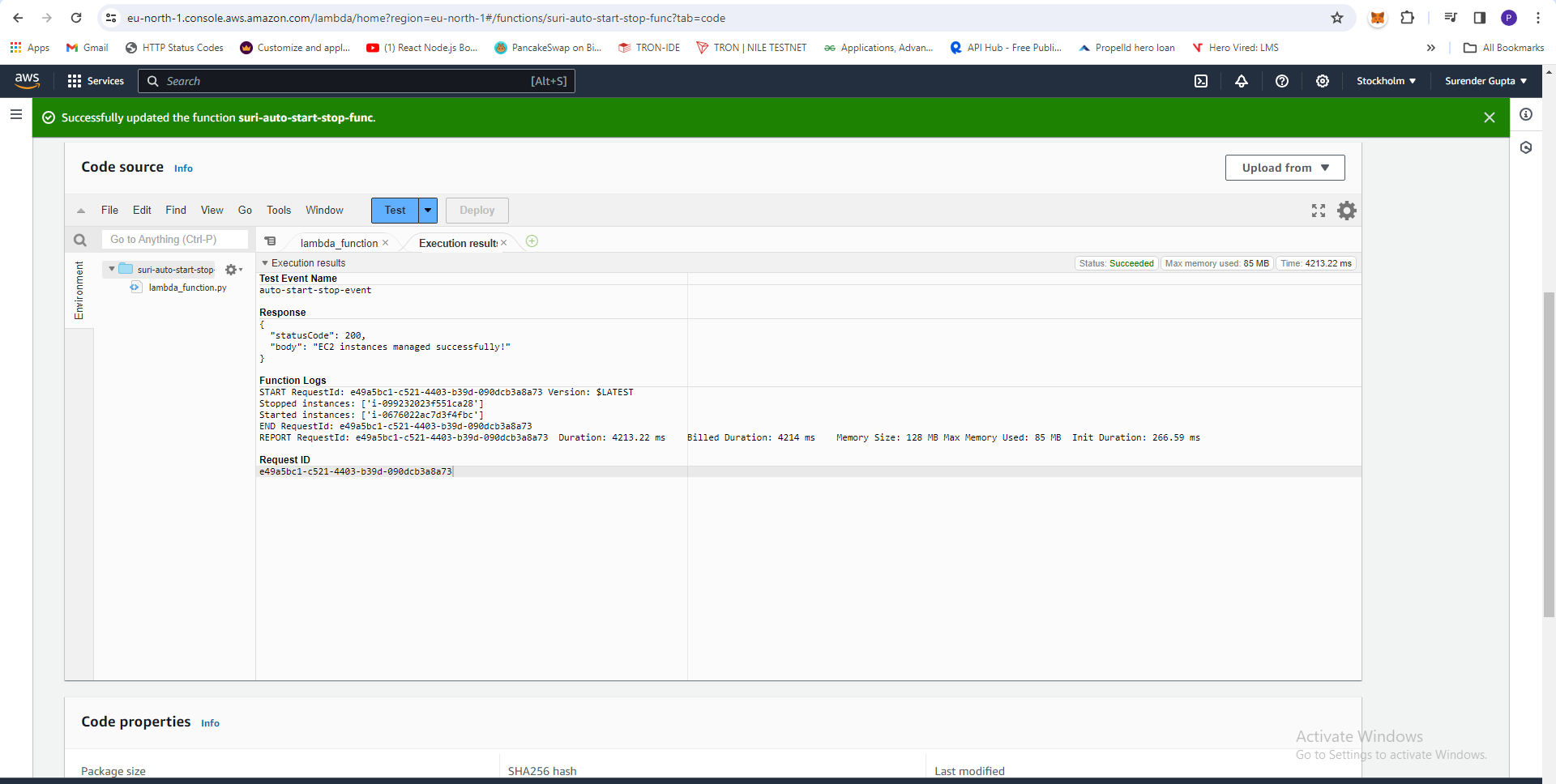
return {

'statusCode': 200,

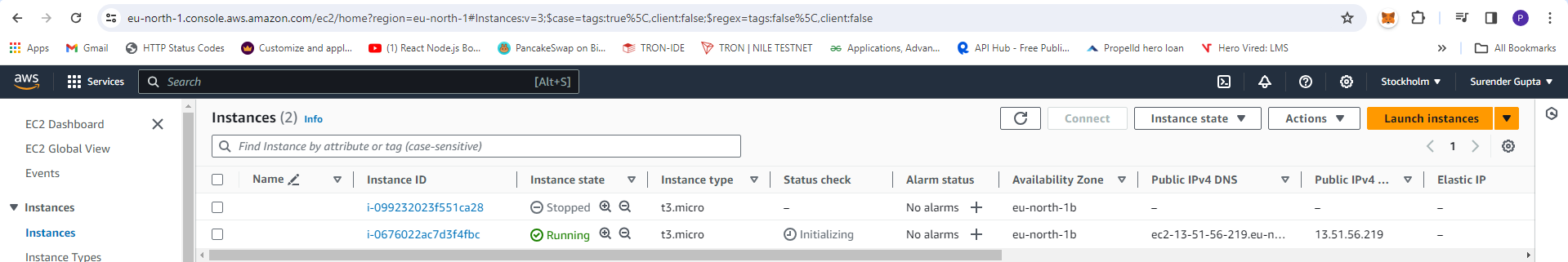
'body': 'EC2 instances managed successfully!'

}

1. Click on Deploy to save the python script and then click on Test script
2. After Test result show its Output below:



1. EC2 Instance update its state see below:



GitHub Link: https://github.com/surendergupta/Assignment-On-Serverless-Architecture