# Serverless EC2 Instance Scheduler for Company Working Hours

This project provides a fully serverless solution to automate the scheduling of EC2 instances based on company working hours. Companies often do not require EC2 instances to run 24/7, instead needing them only during specific hours, such as 8:00 AM to 5:00 PM. This solution uses AWS Lambda functions and CloudWatch Events to start and stop instances according to a defined schedule.

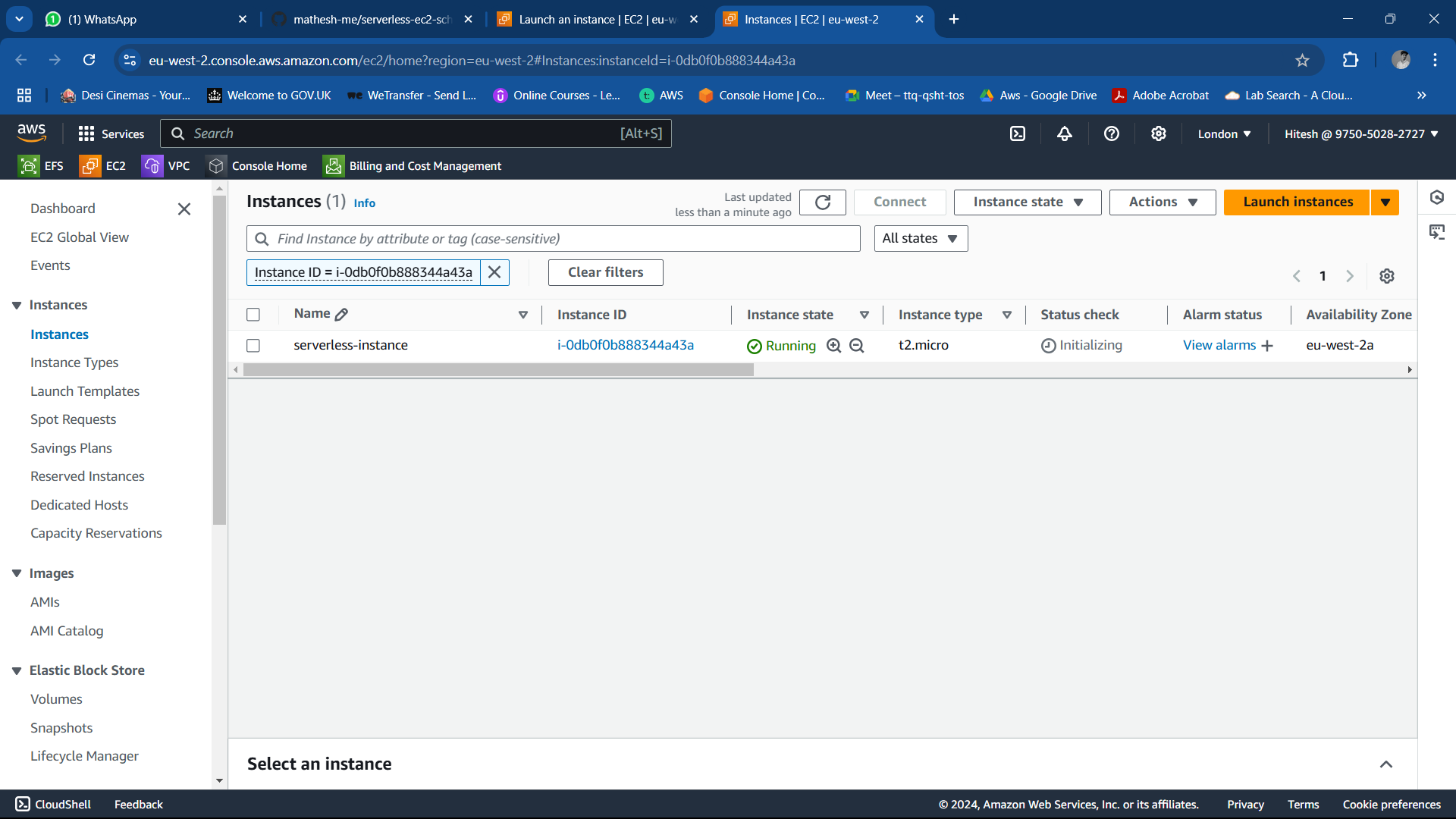
Key Features:

* Automates EC2 instance management to optimize costs.
* Implements serverless architecture with AWS Lambda.
* Leverages CloudWatch Events for precise scheduling.

Implementation Steps:

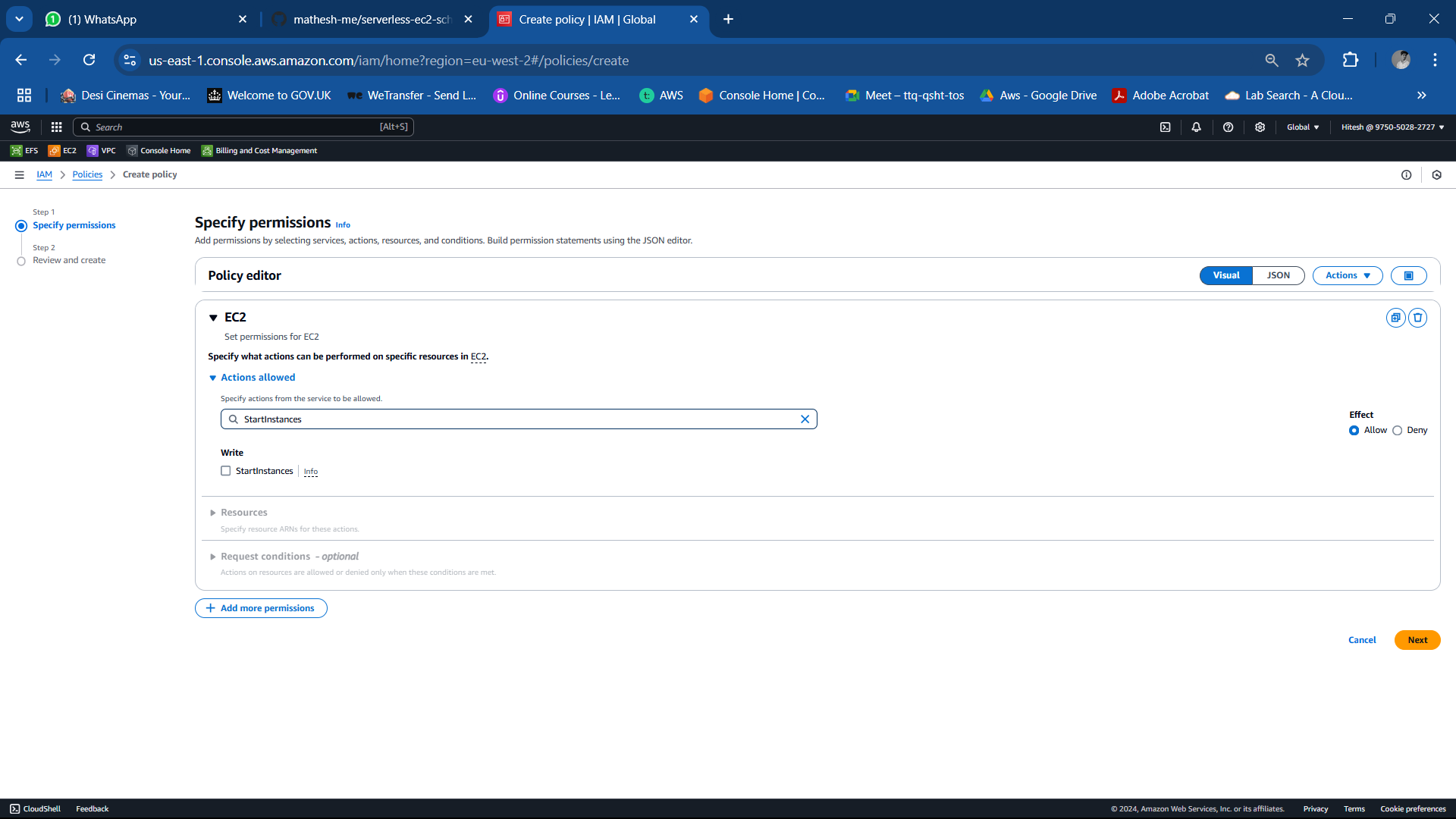
Step 1: **Create the EC2 Instance**

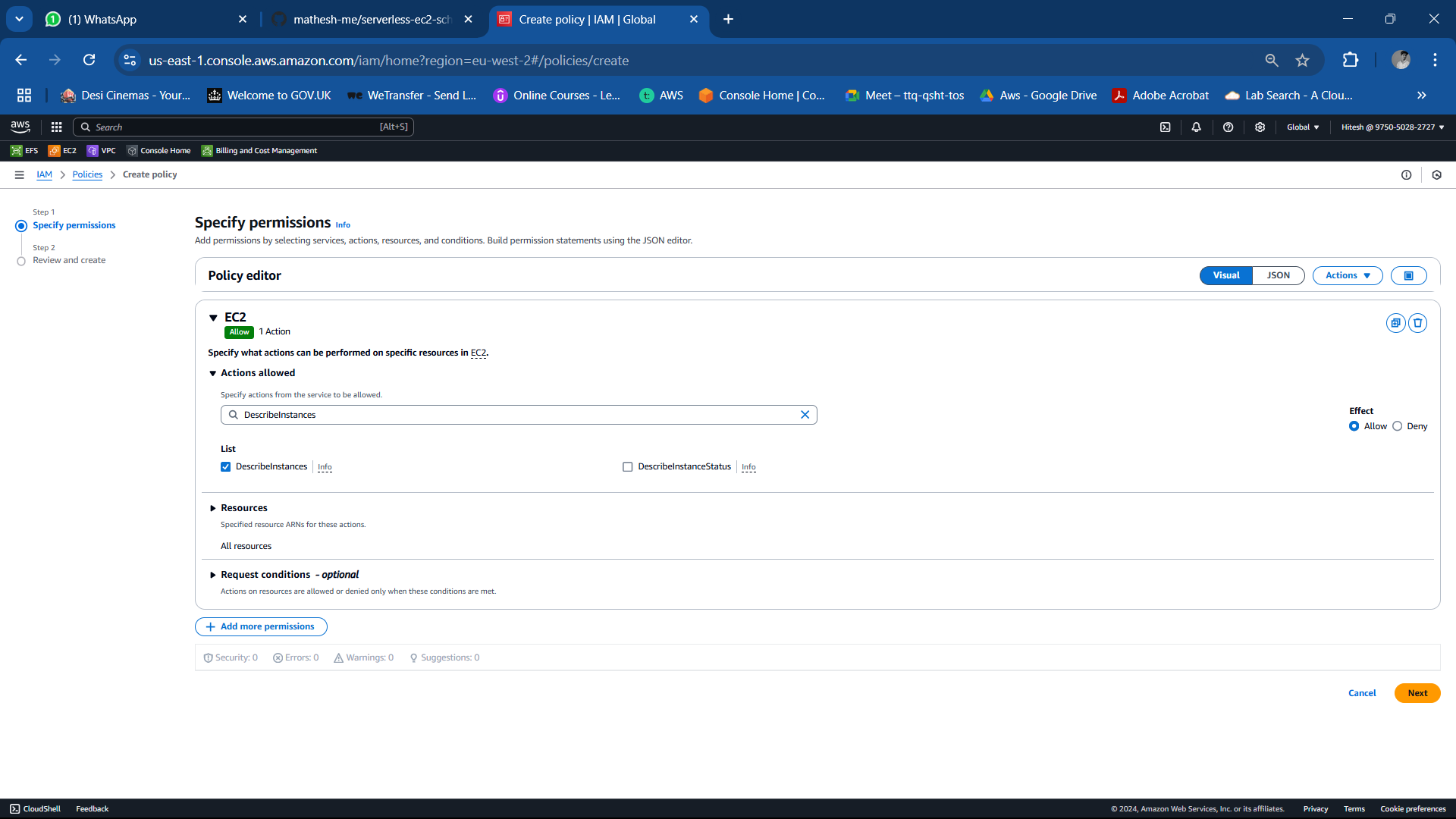
1. Navigate to the EC2 Console and create an instance following the standard setup procedure.

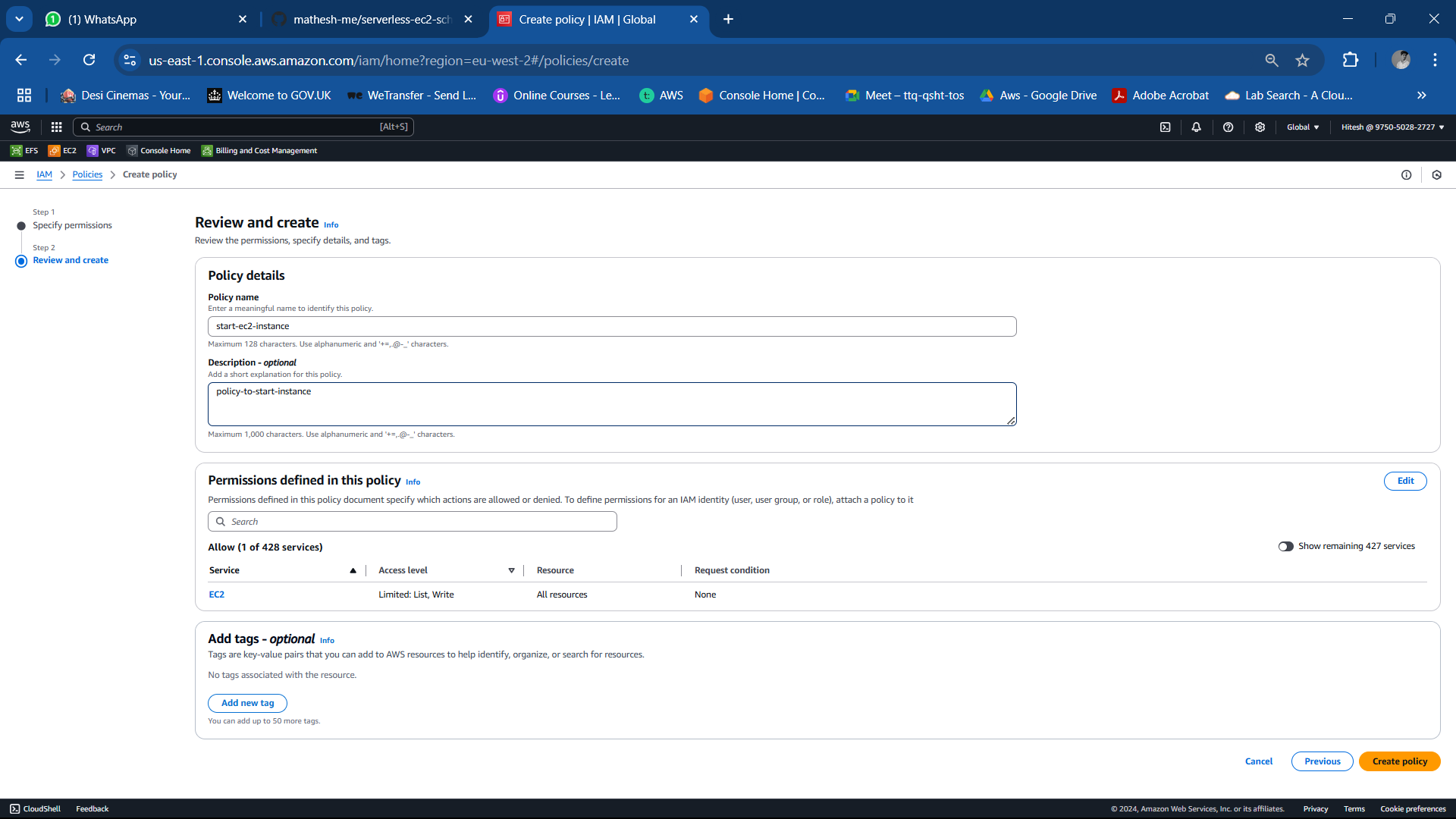


Step 2: **Define IAM Policies**

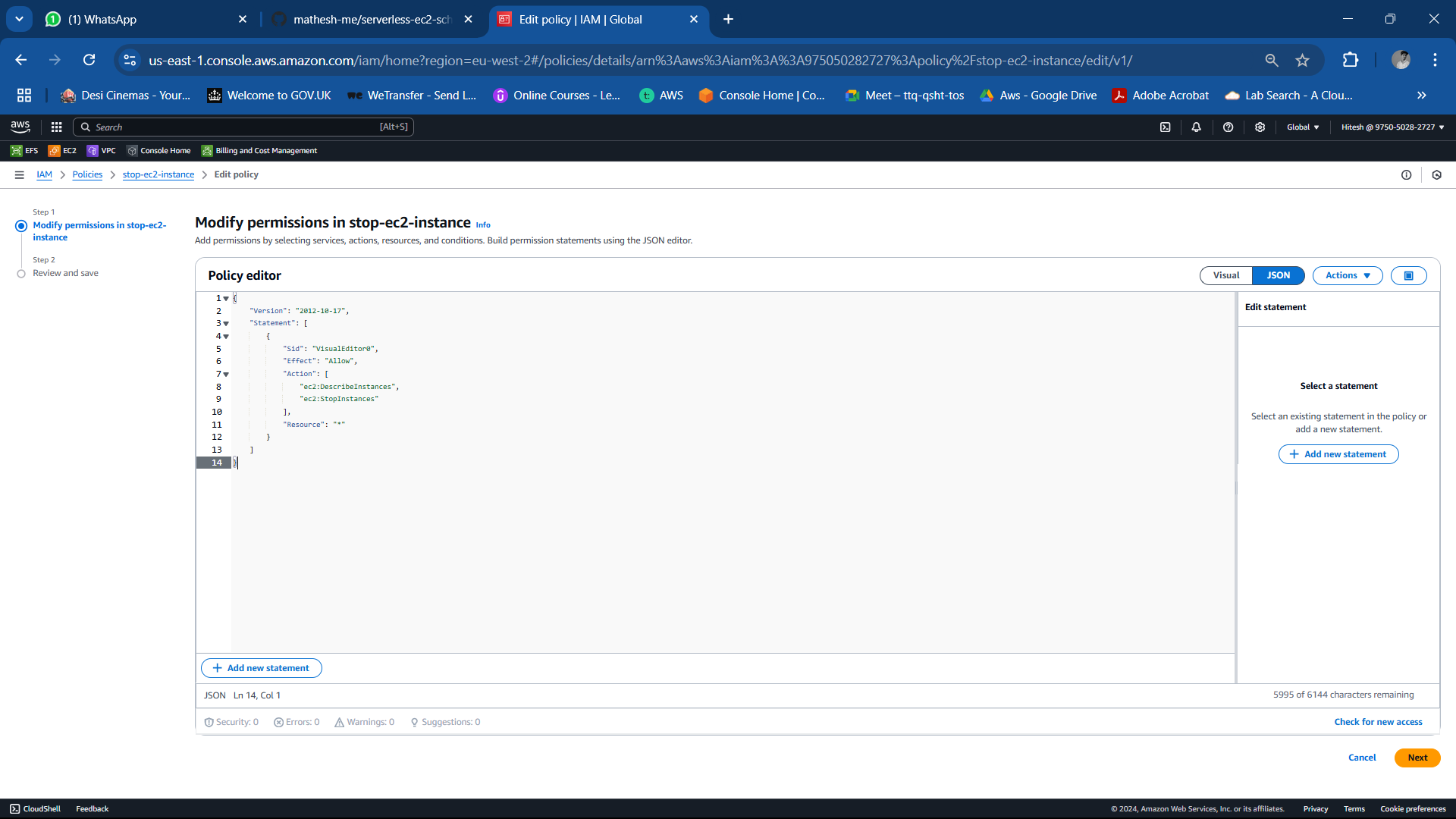
1. Go to the IAM Console and create policies for managing EC2 instances.
2. Create two separate policies:
   * **Start EC2 Policy**:
     + Actions: DescribeInstances, StartInstances
     + Name: start-ec2-instance

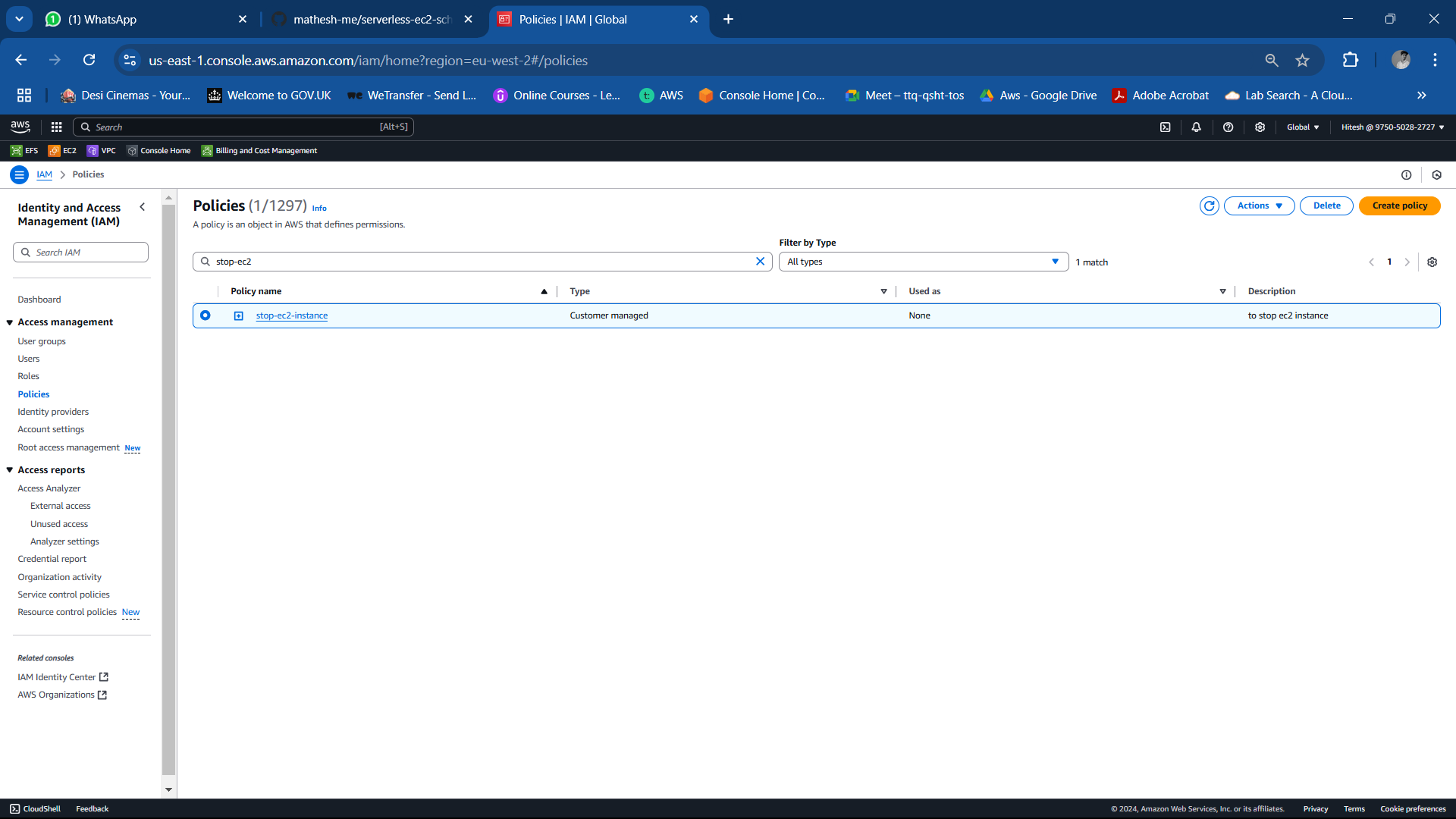






* + **Stop EC2 Policy**:
    - Actions: DescribeInstances, StopInstances
    - Name: stop-ec2-instance





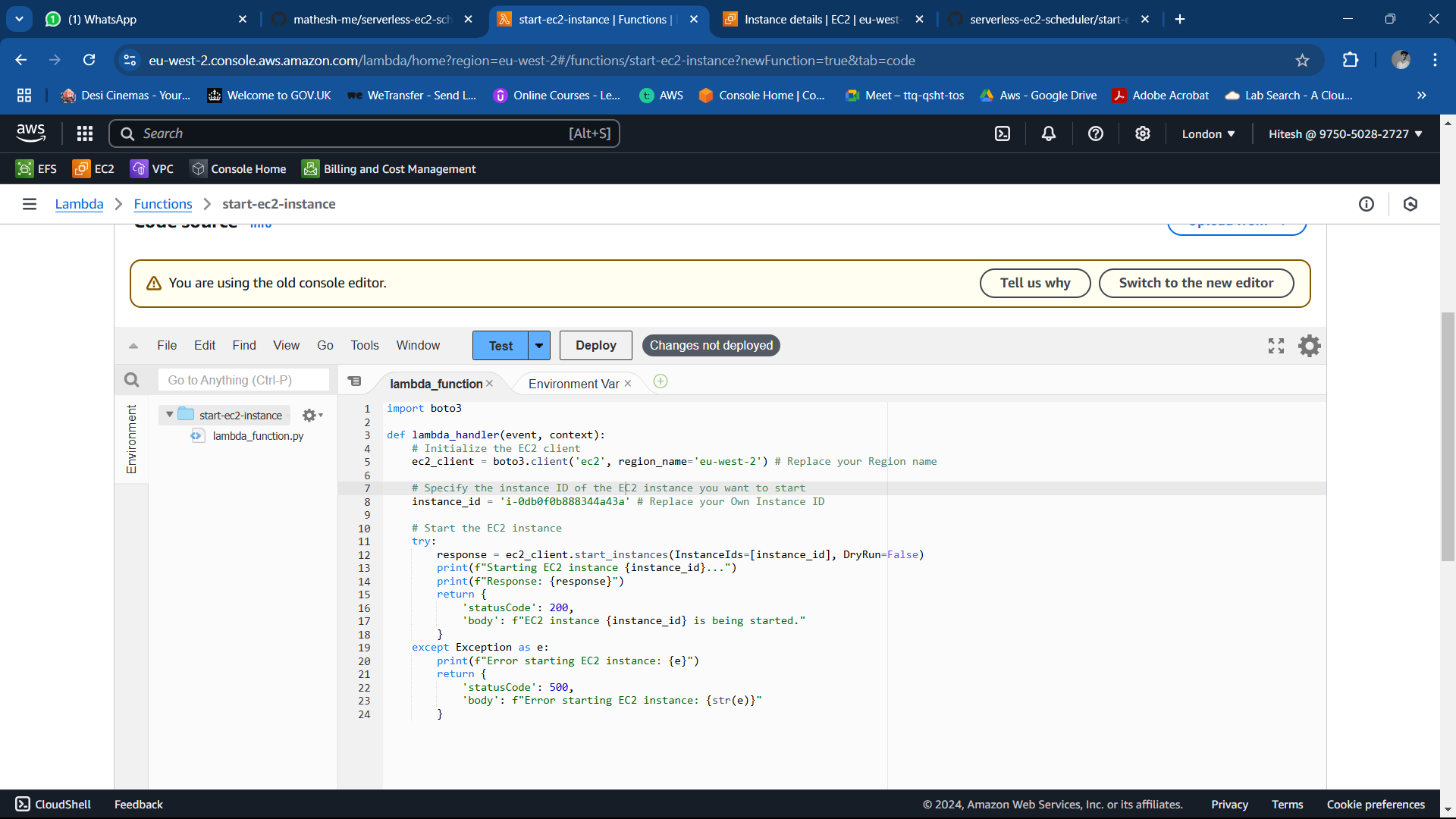
1. These policies will be attached to separate roles for the Lambda functions.

Step 3: **Create Lambda Functions**

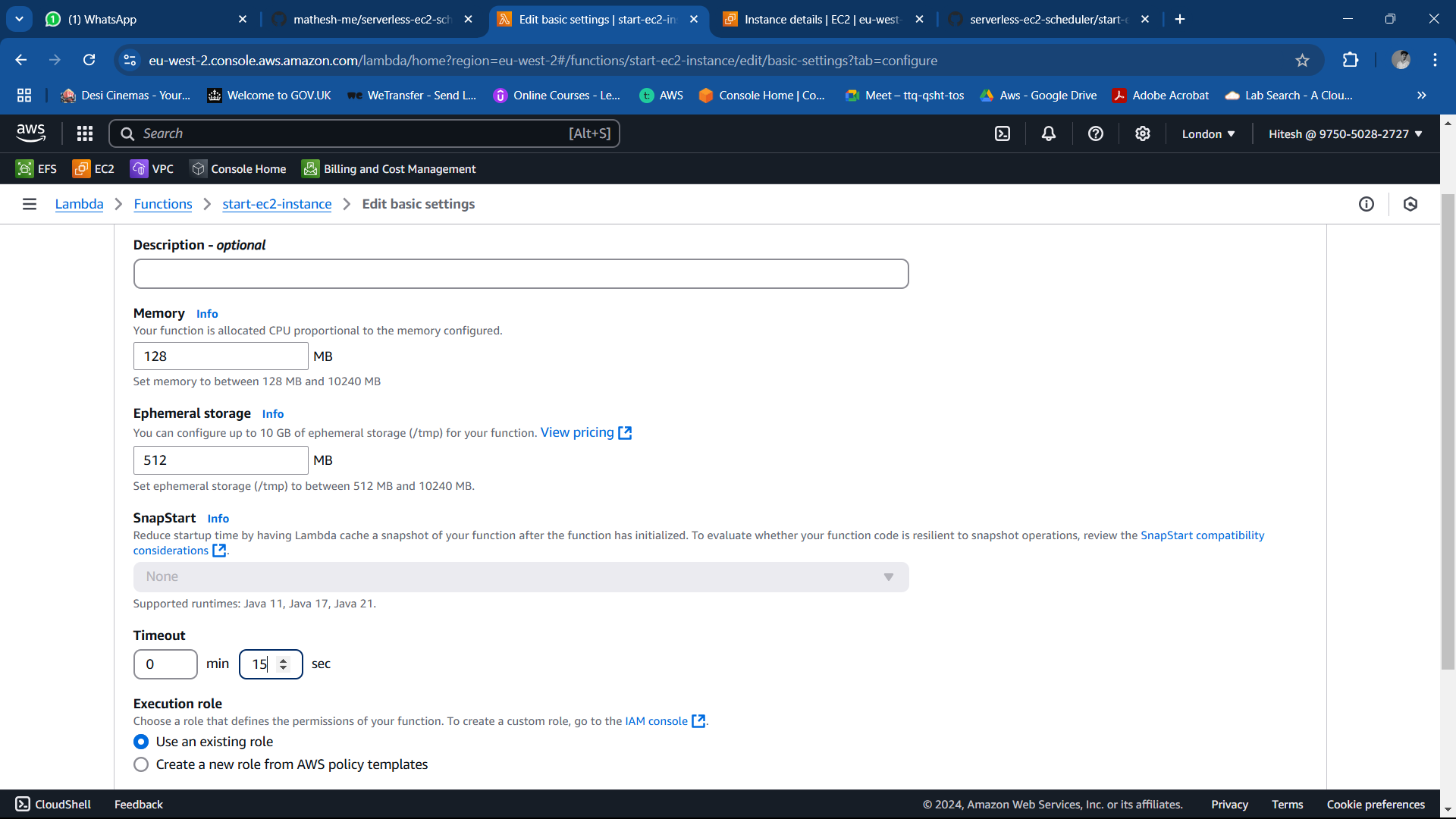
1. Navigate to the Lambda Console.



1. Create two functions:
   * **Function1 : Start Instance Function**:
     + Use the script start-ec2-instance.py.
     + Attach the start-ec2-instance policy.



Once you changed the region name and instance id with the code click deploy and go to configure

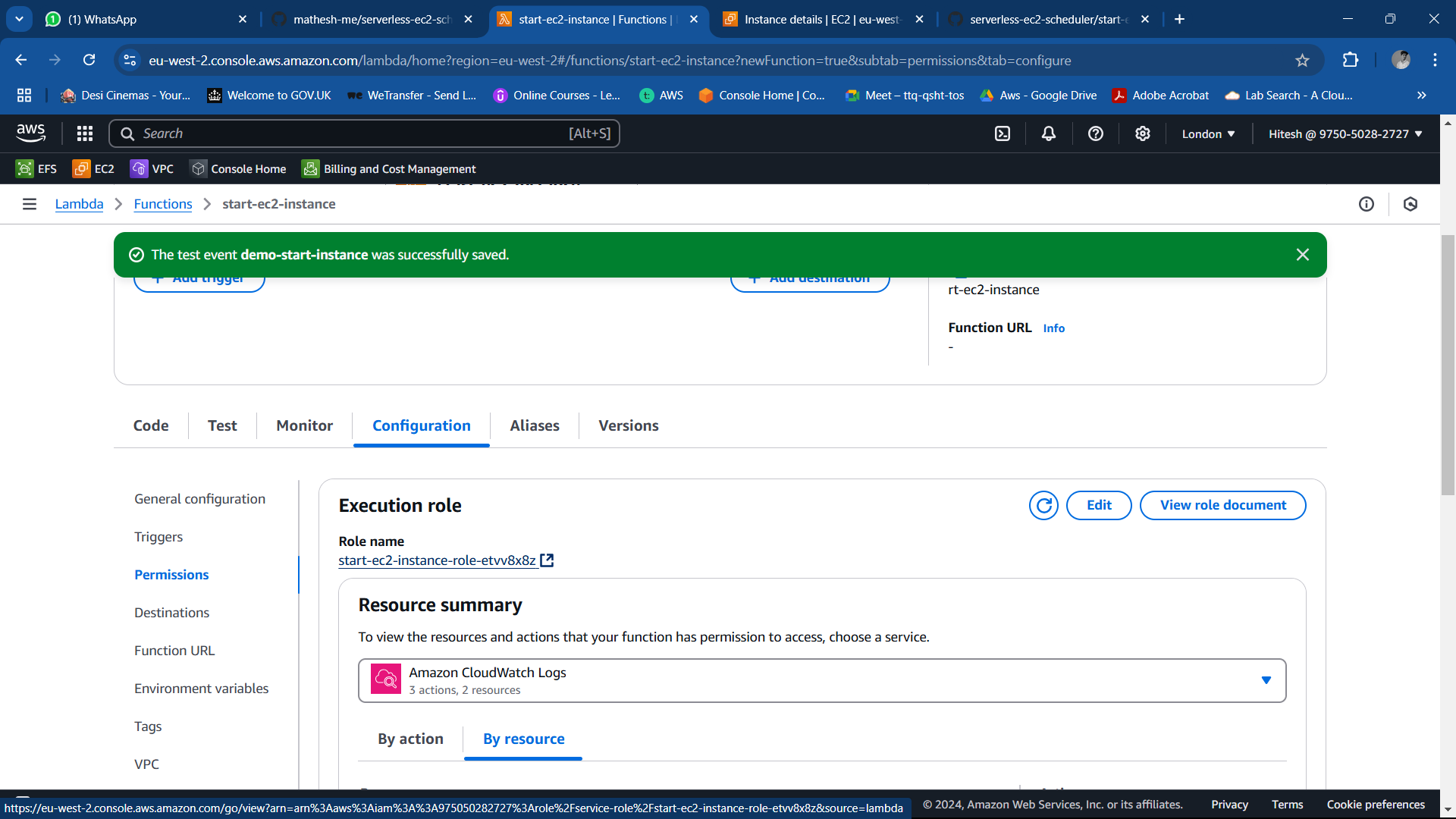
The click on edit and scroll down and change the **Timeout** to 15 sec

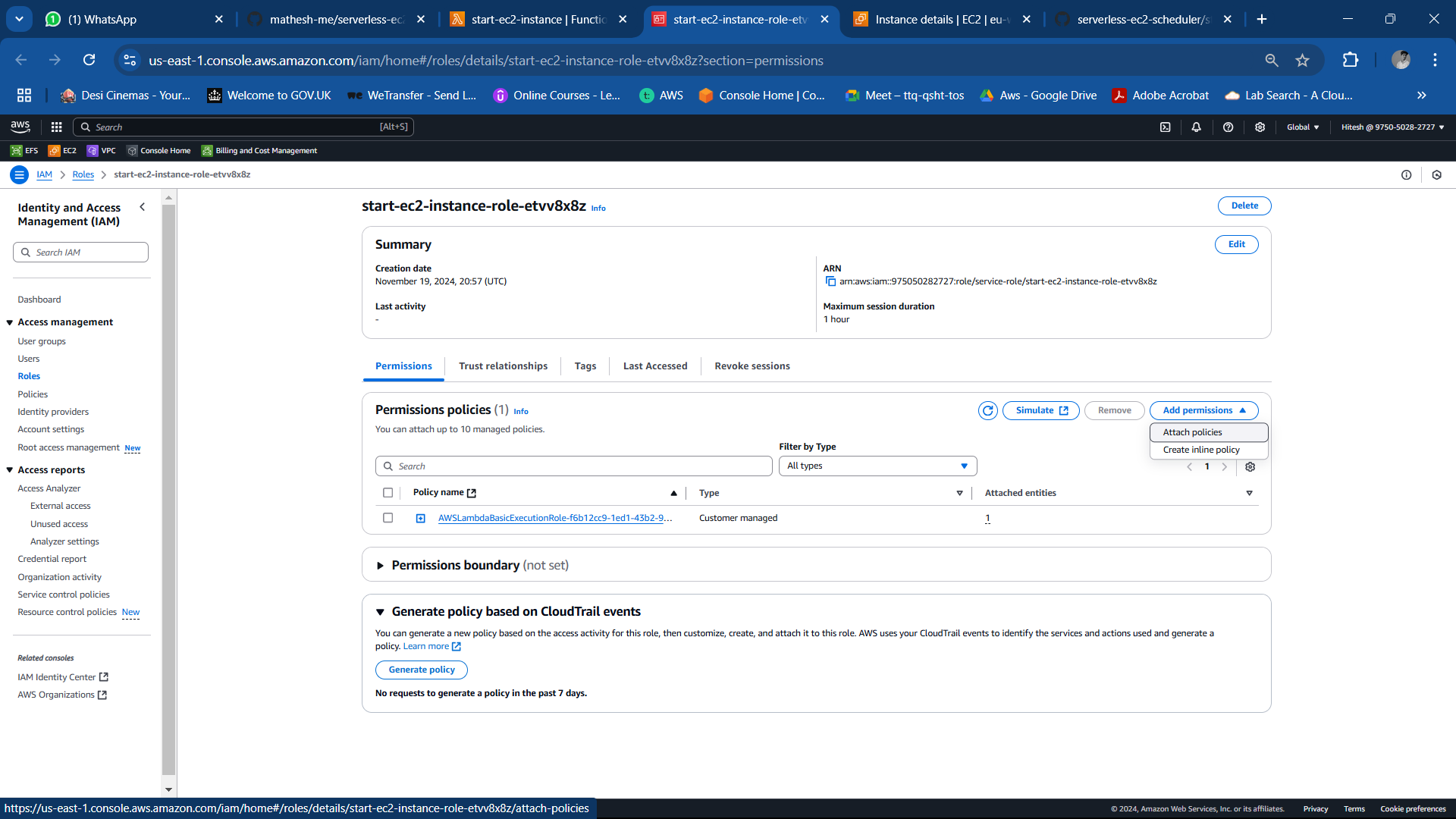
* + **Function 2 : Stop Instance Function**:
    - Use the script stop-ec2-instance.py.
    - Attach the stop-ec2-instance policy.

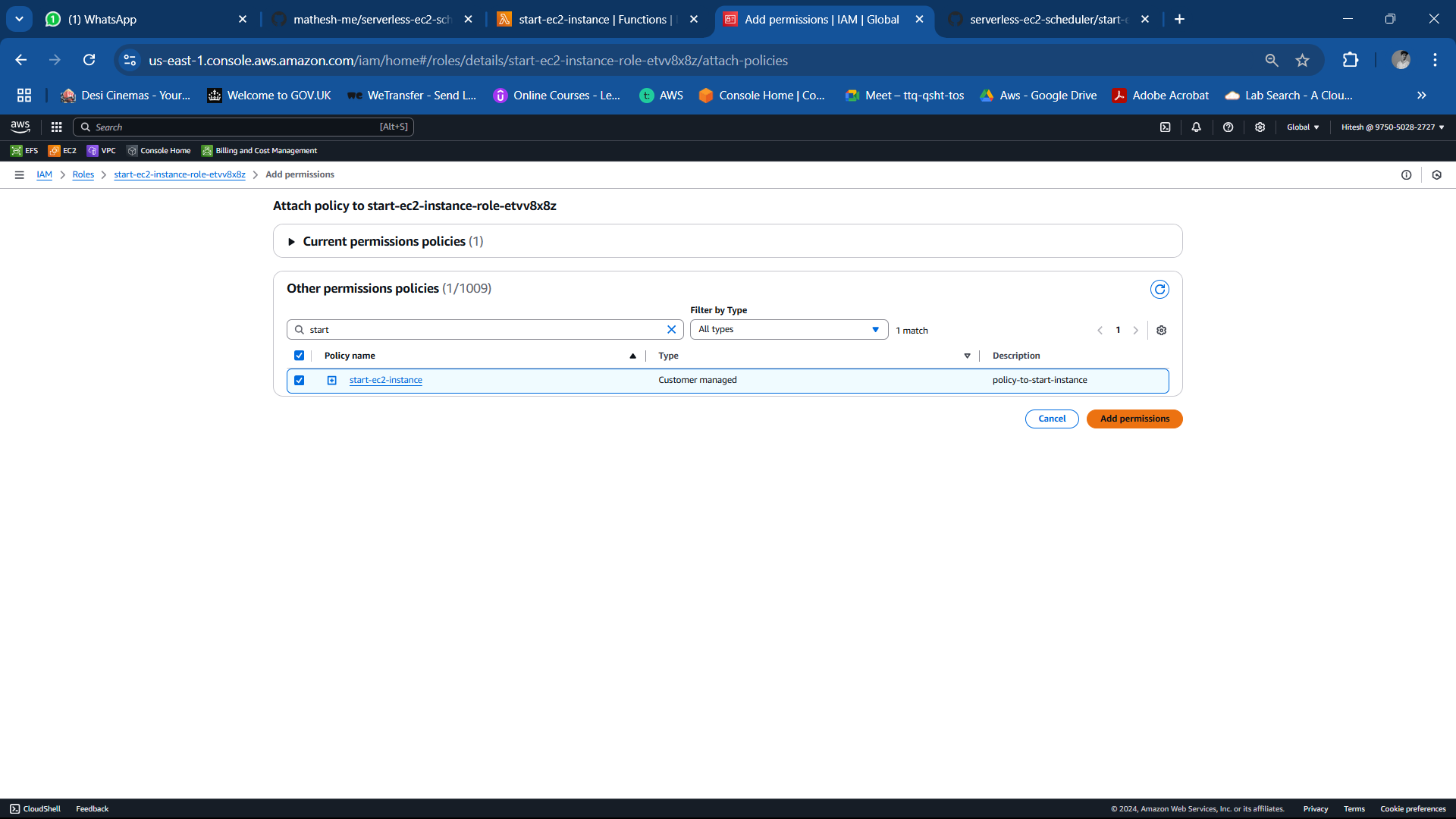
Once you change to timeout save it and click on test the code



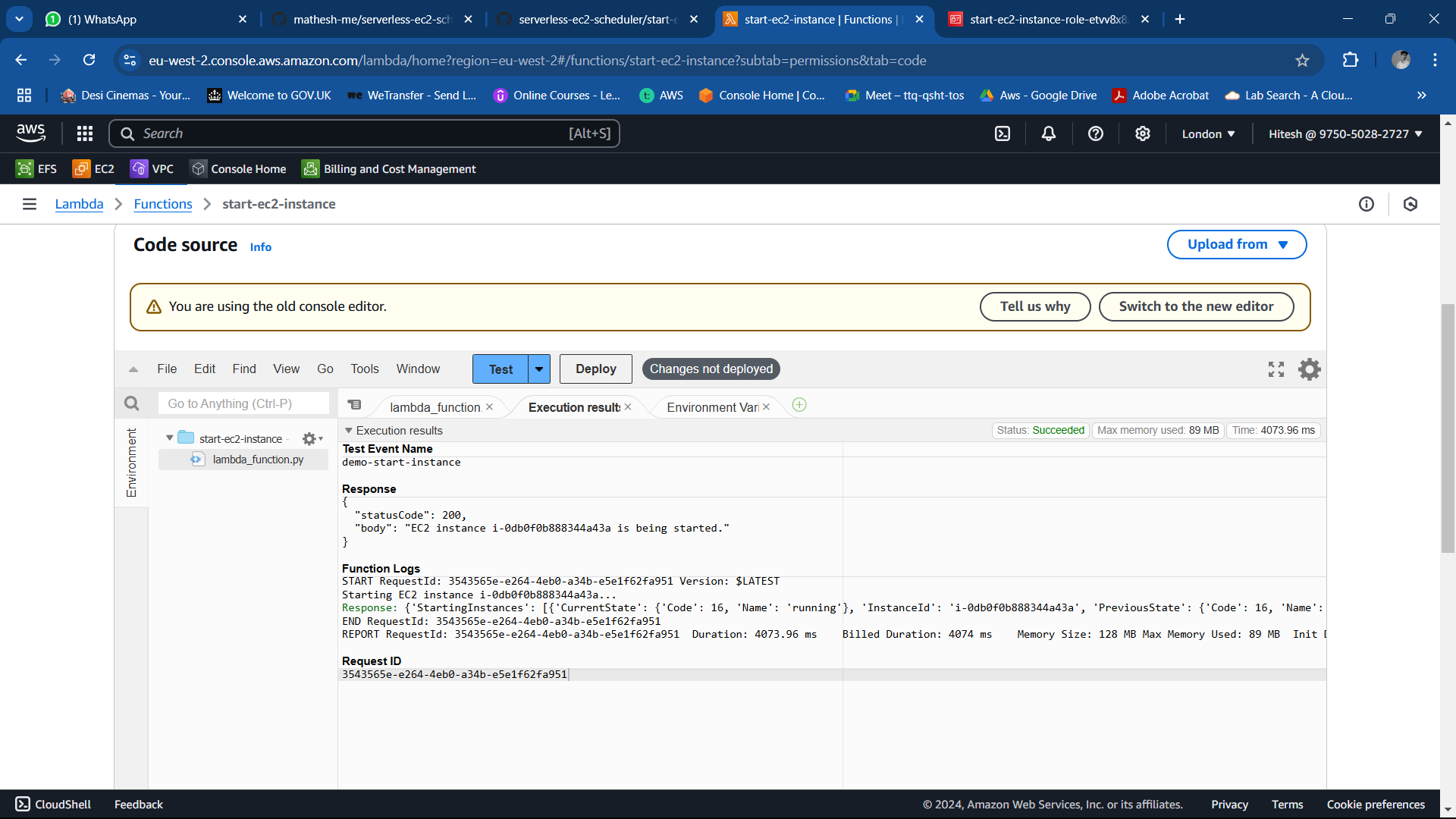
Click on configure and click on role name it will take you to the iam permission now attach the (start-ec2-instance ) permission to event



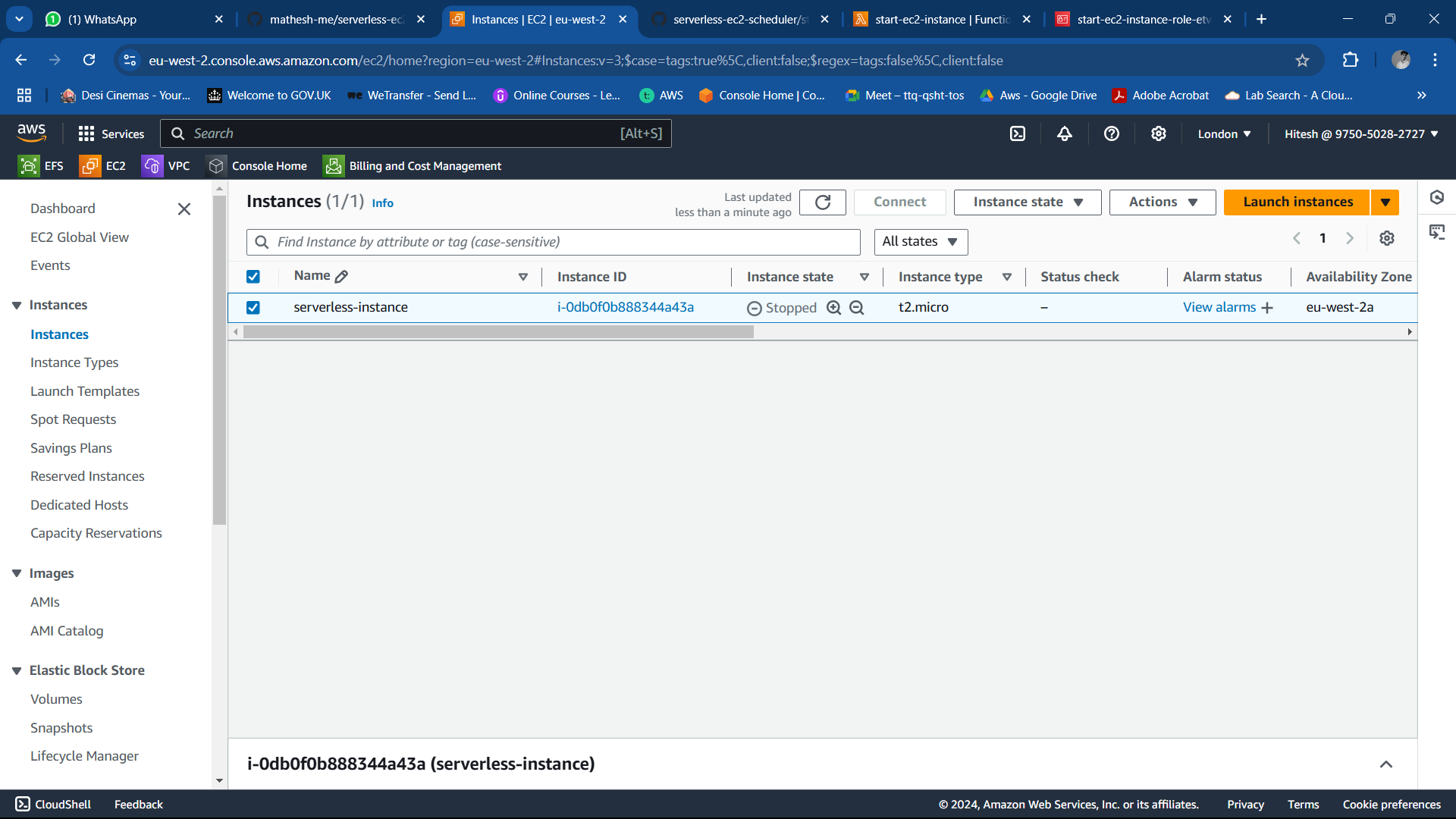




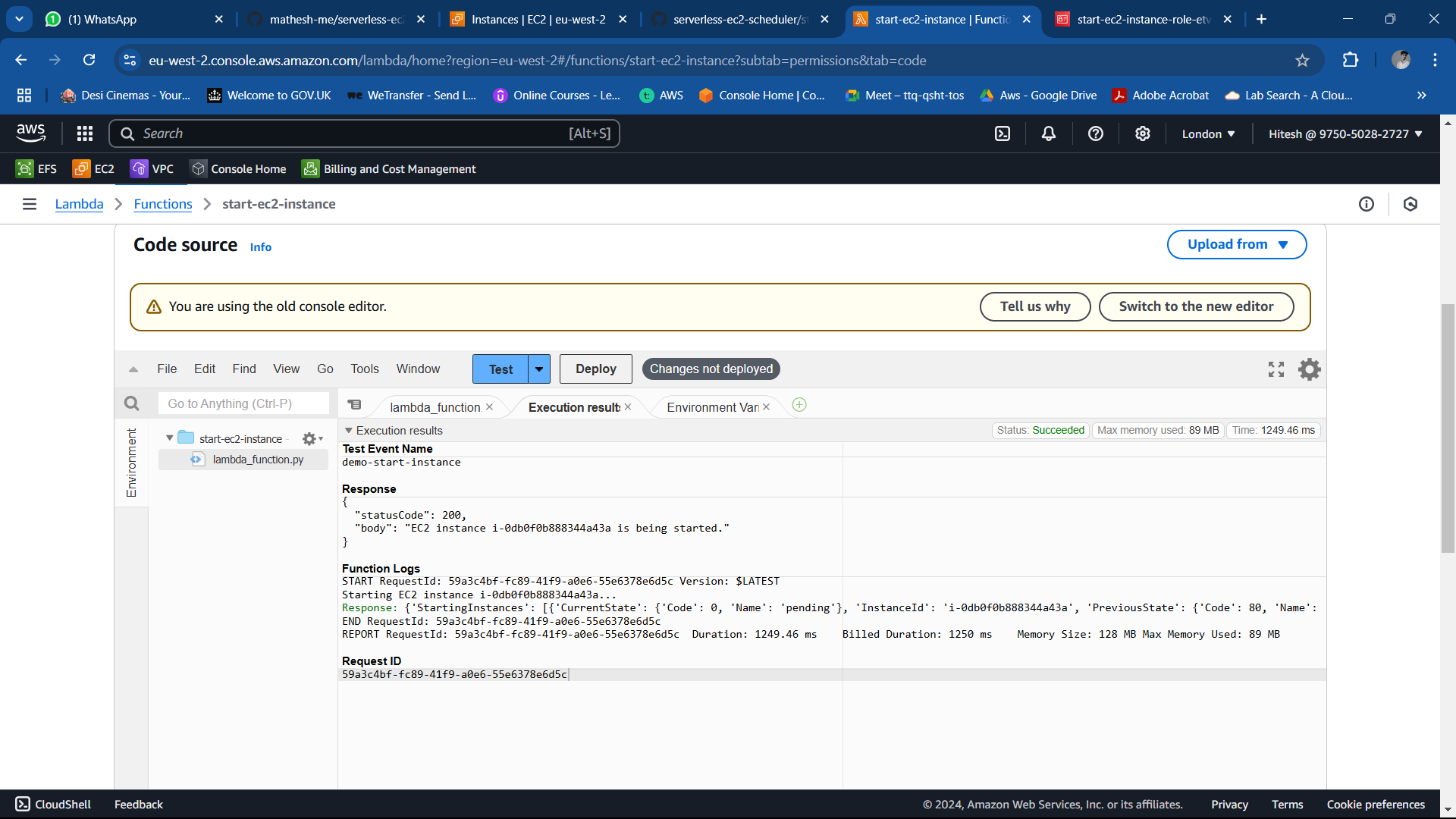
Once we attach policy to each function accordingly then after test the code in each function to see our setup is working or not.

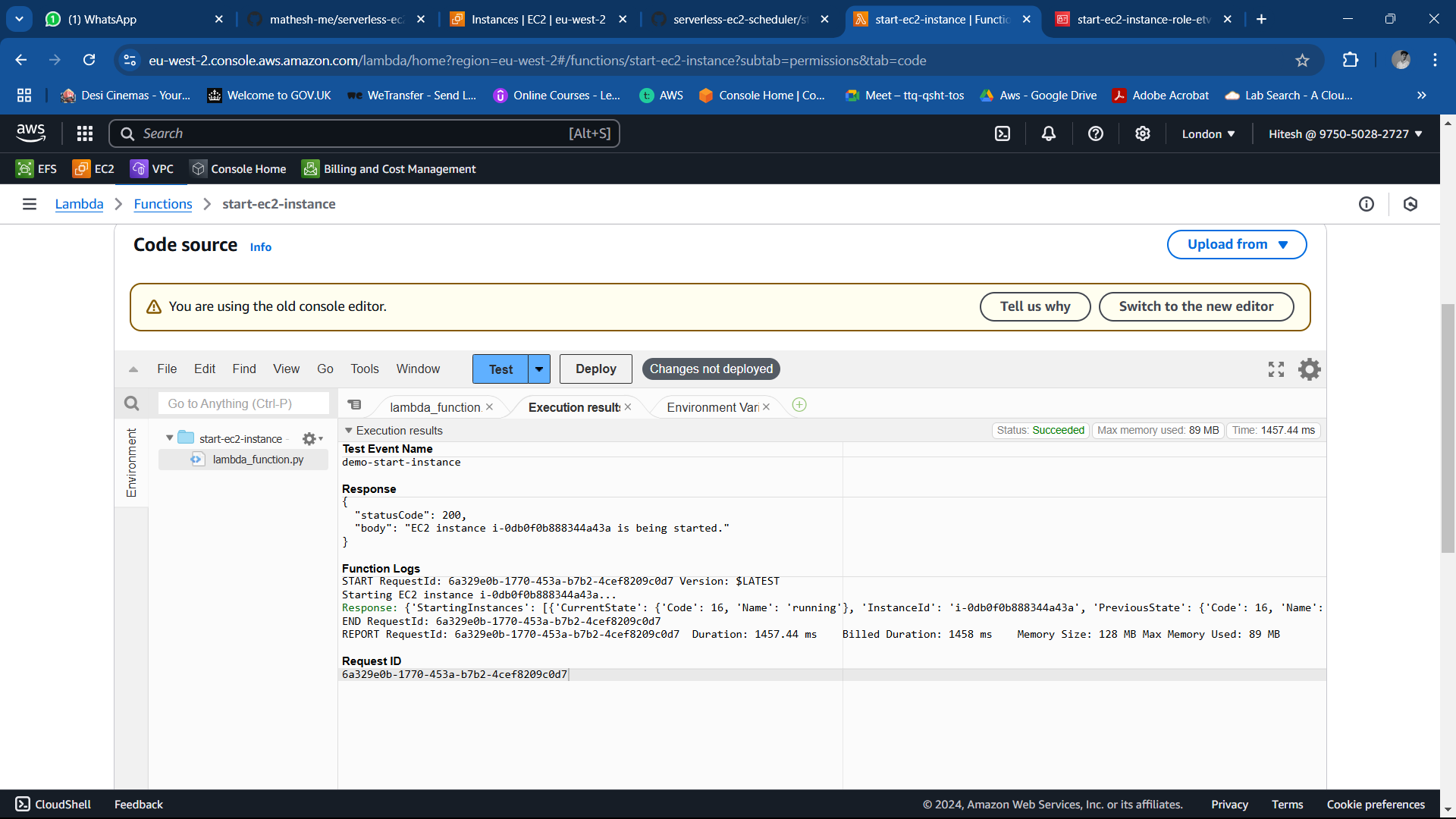


As we can see my instance is already in running state. So let’s double check my setup is working or not by stopping the instance manually and test the code again and see lambda will start my instance or not with iam policy



We can see in the status that instance is again getting start and now instance state is pending it will be running state shortly

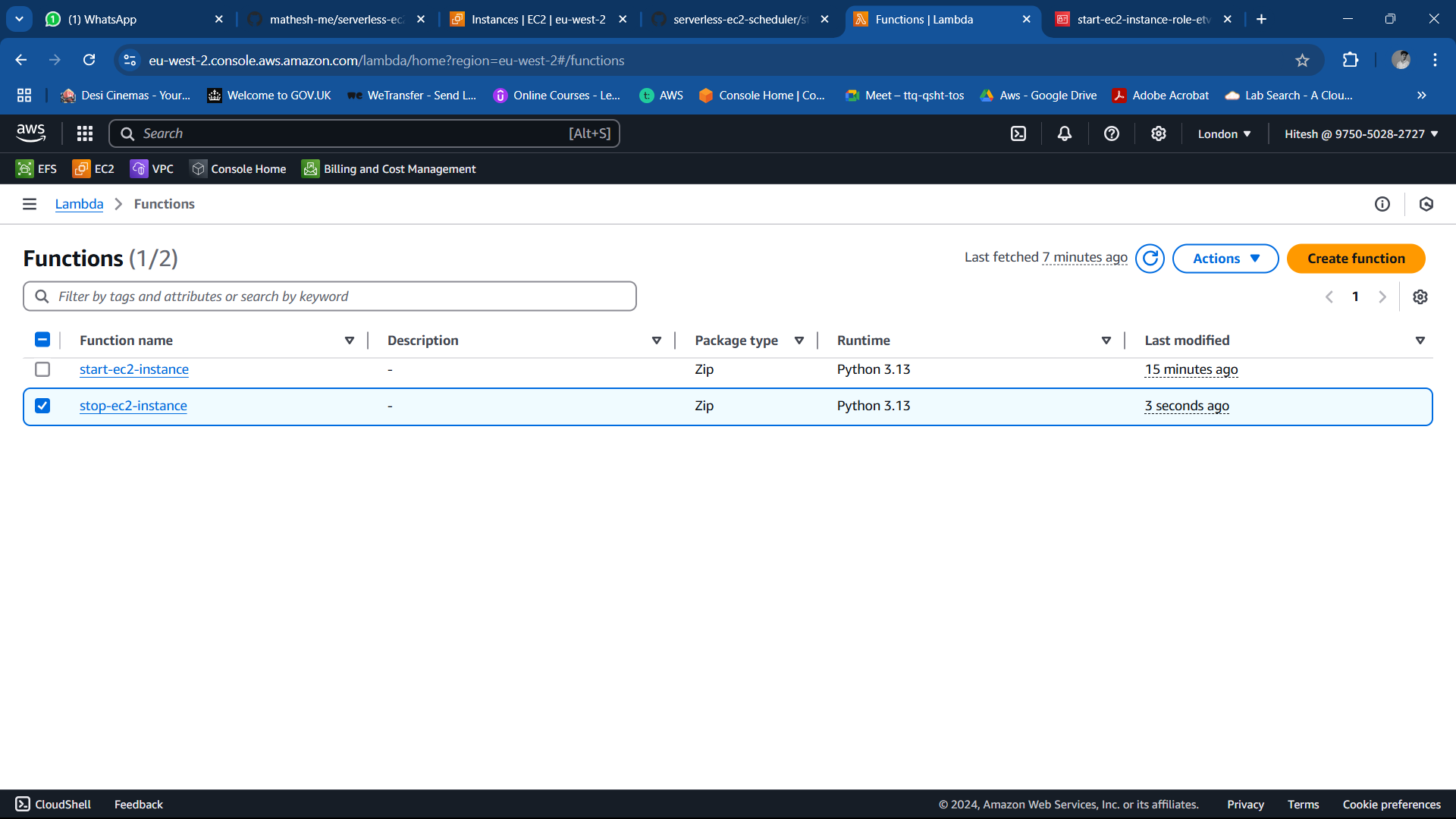


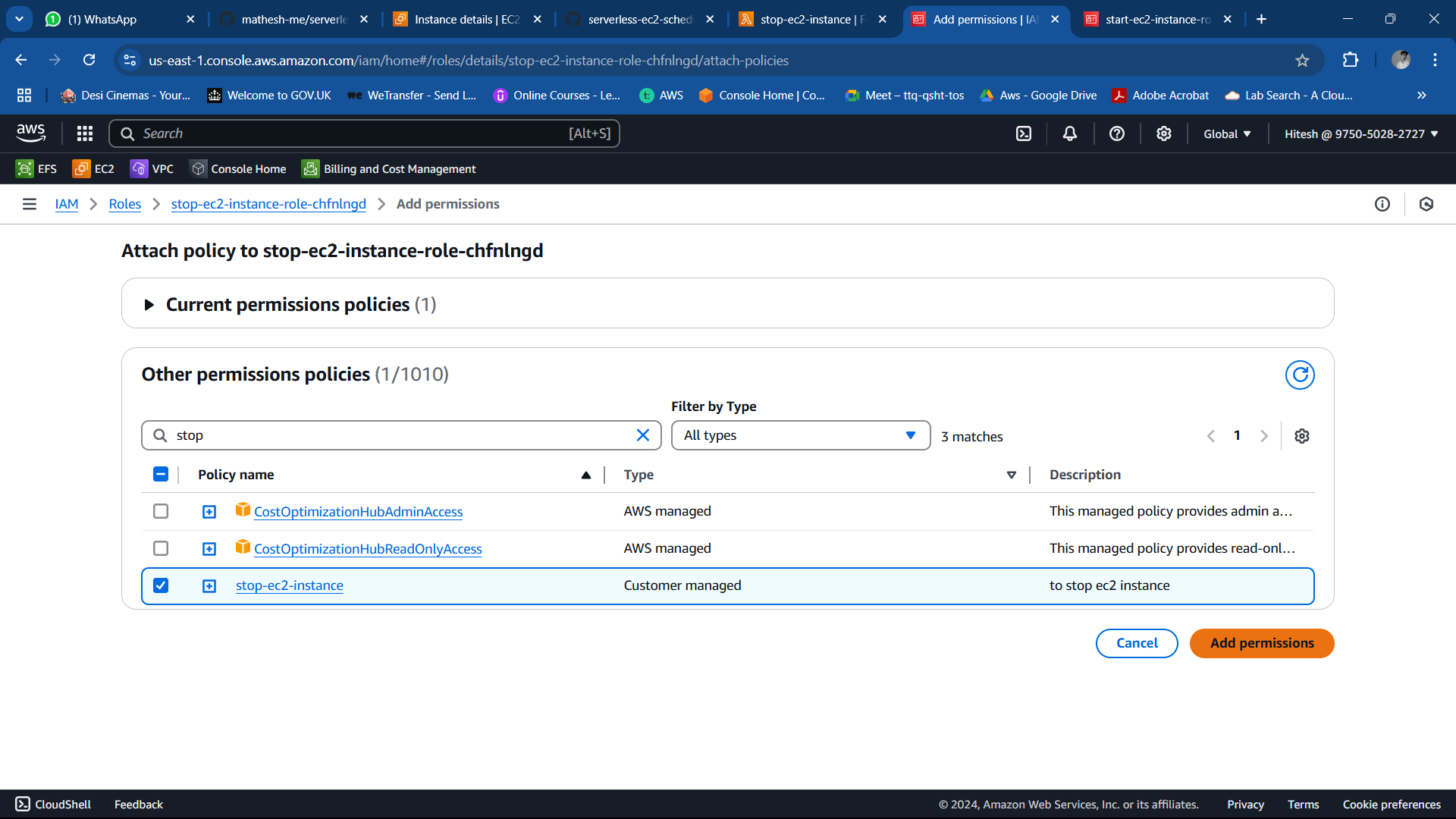


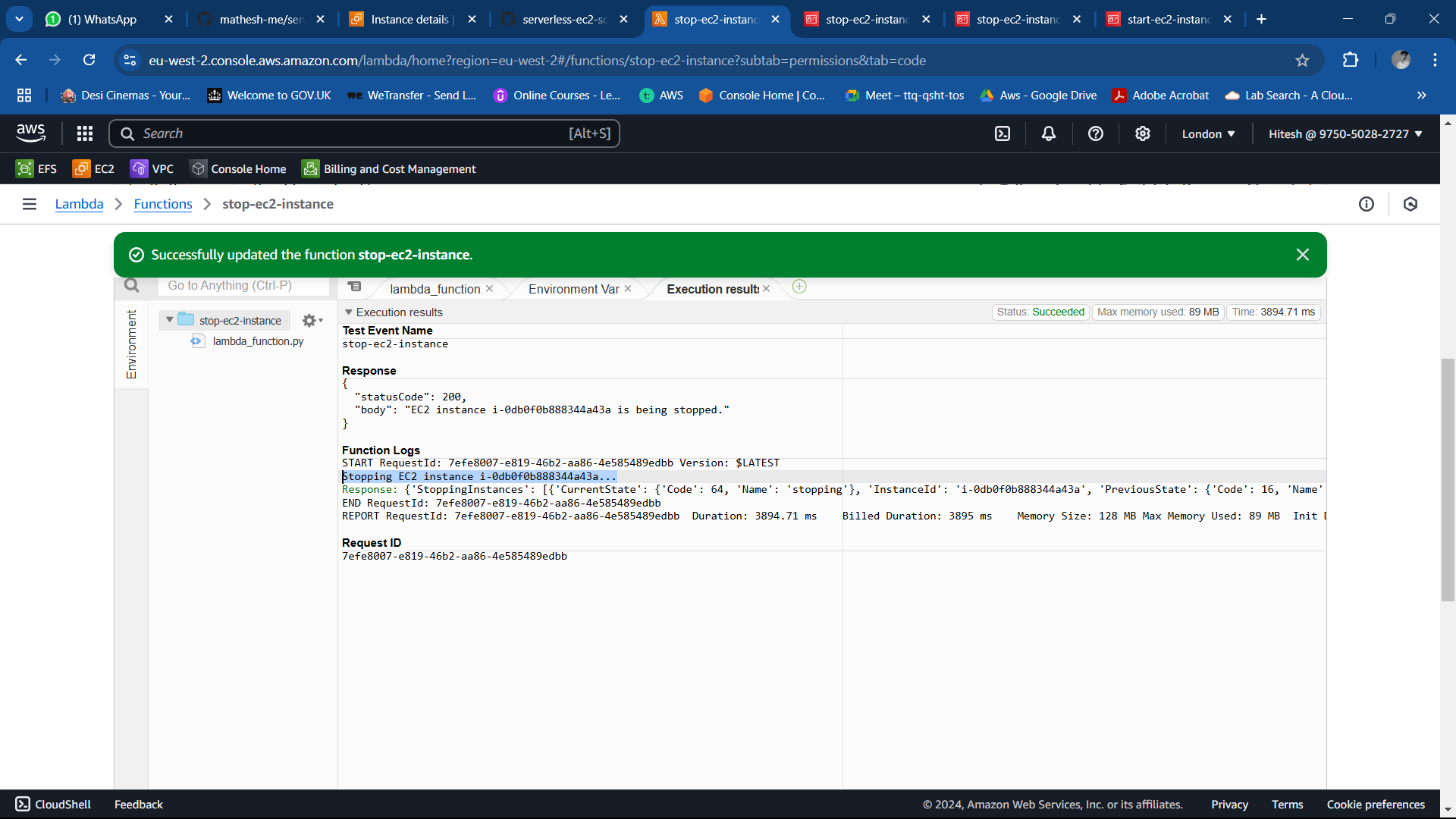
Now perform the same steps to stop the instance using lambda

Create another function named stop-ec2-instance >>> attach the code for stoping instance >> deploy the code >>> >> set the timeout for 15 sec>>attach Iam policy(stop-ec2-policy)

And test the code it will stop the instance





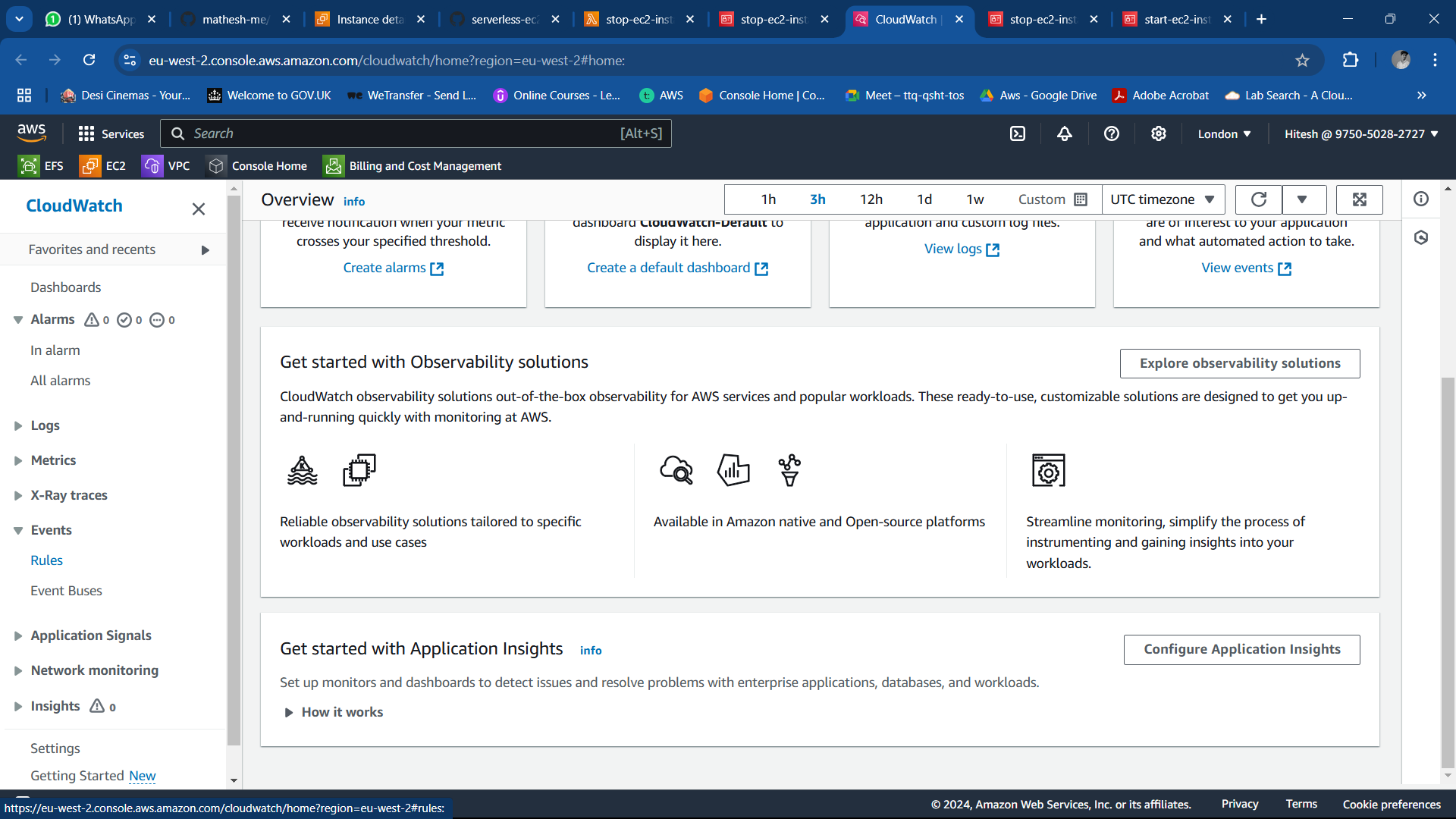


Test each Lambda function to ensure proper execution.

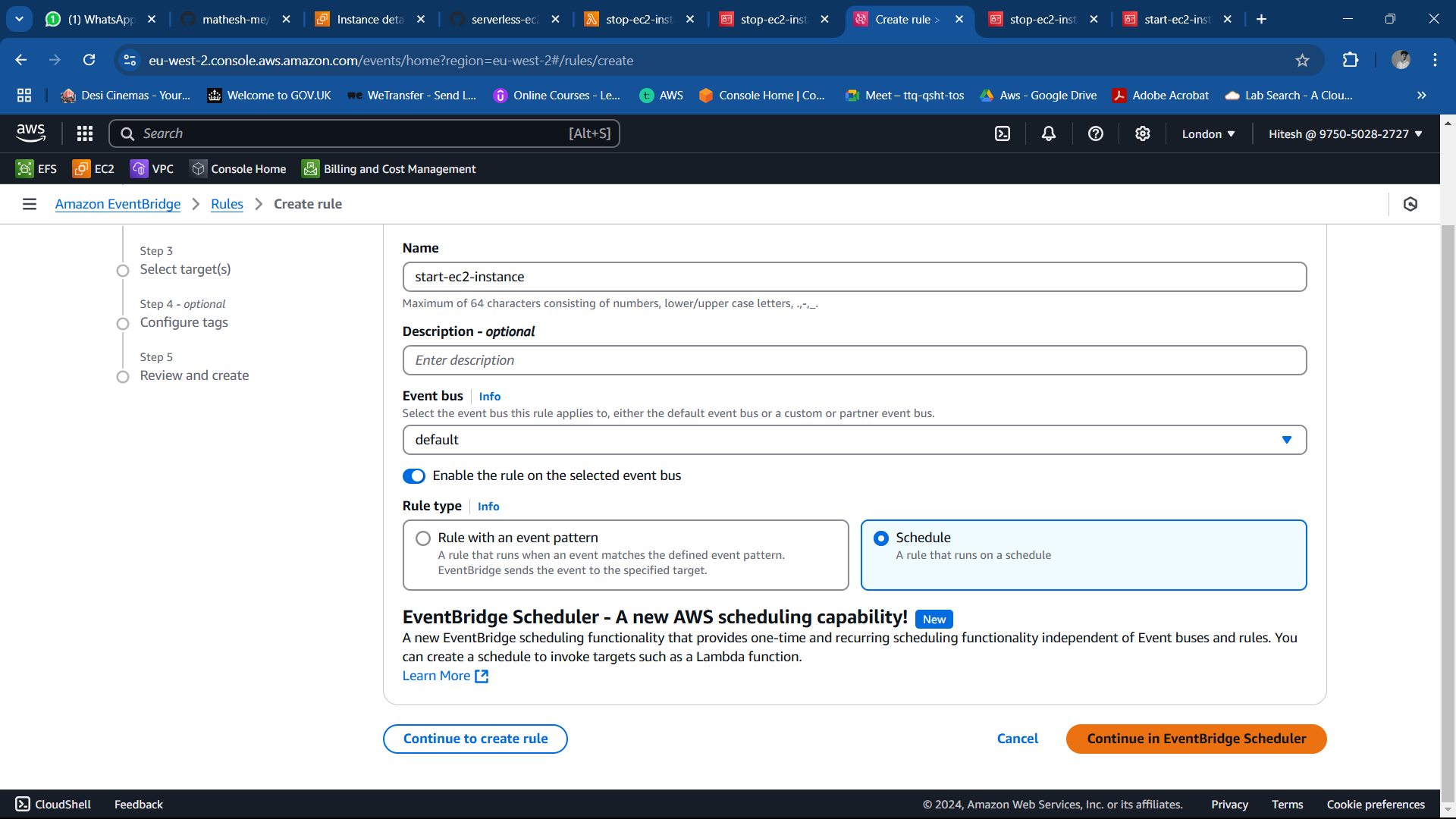
Step 4: **Schedule Using CloudWatch**

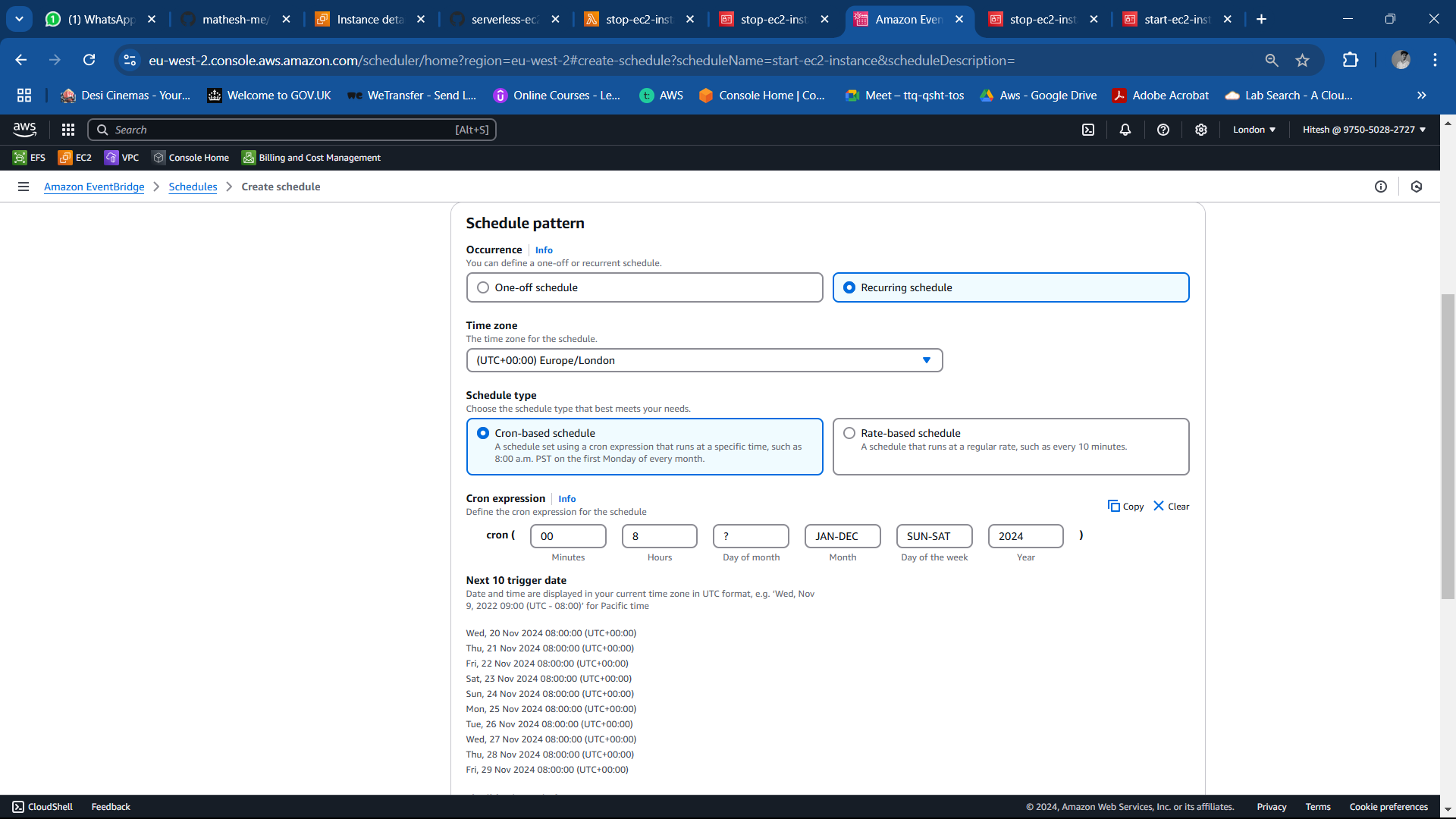
1. Navigate to the CloudWatch Console.

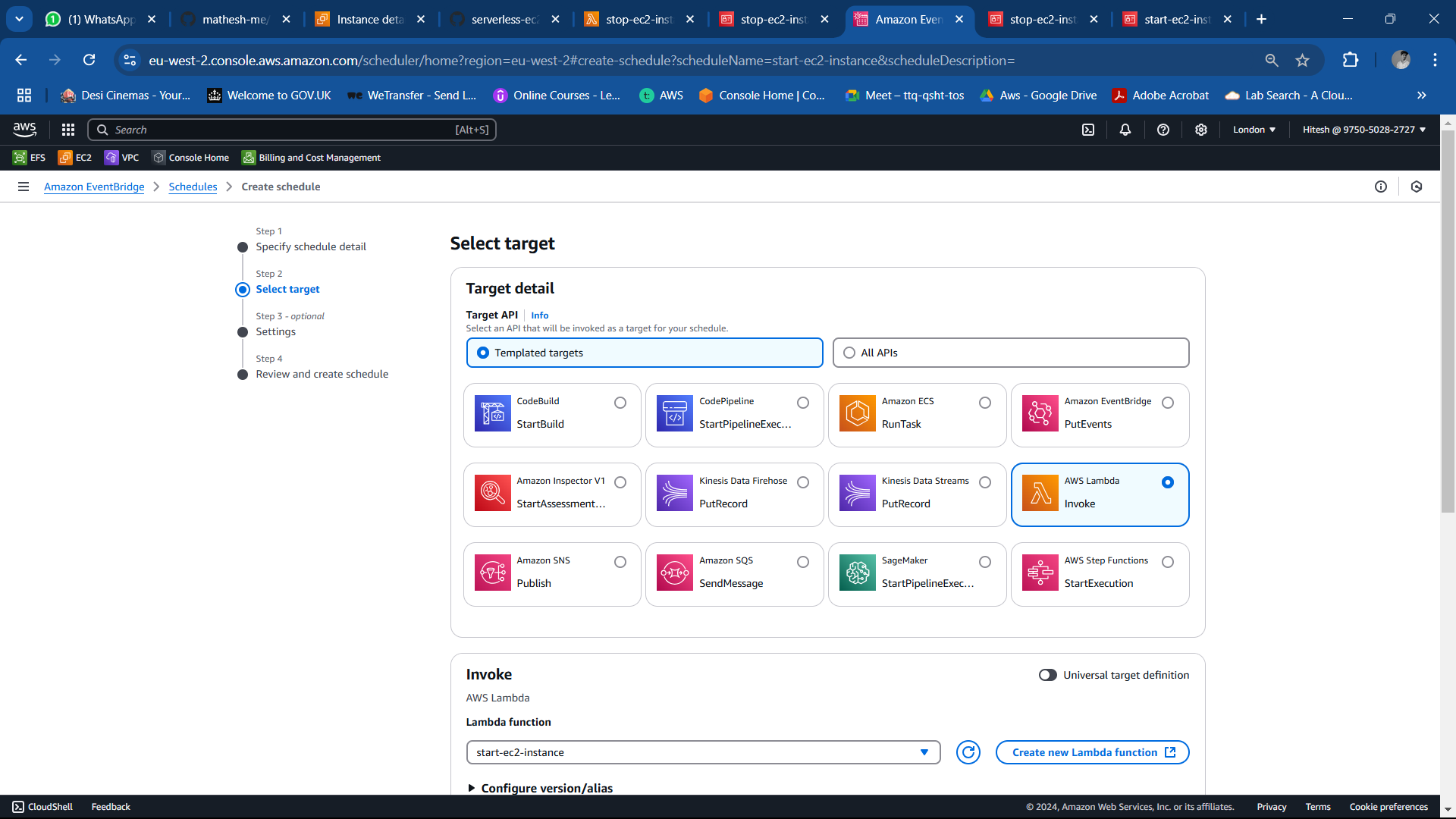
Go to cloud watch >>> navigate to event >> role

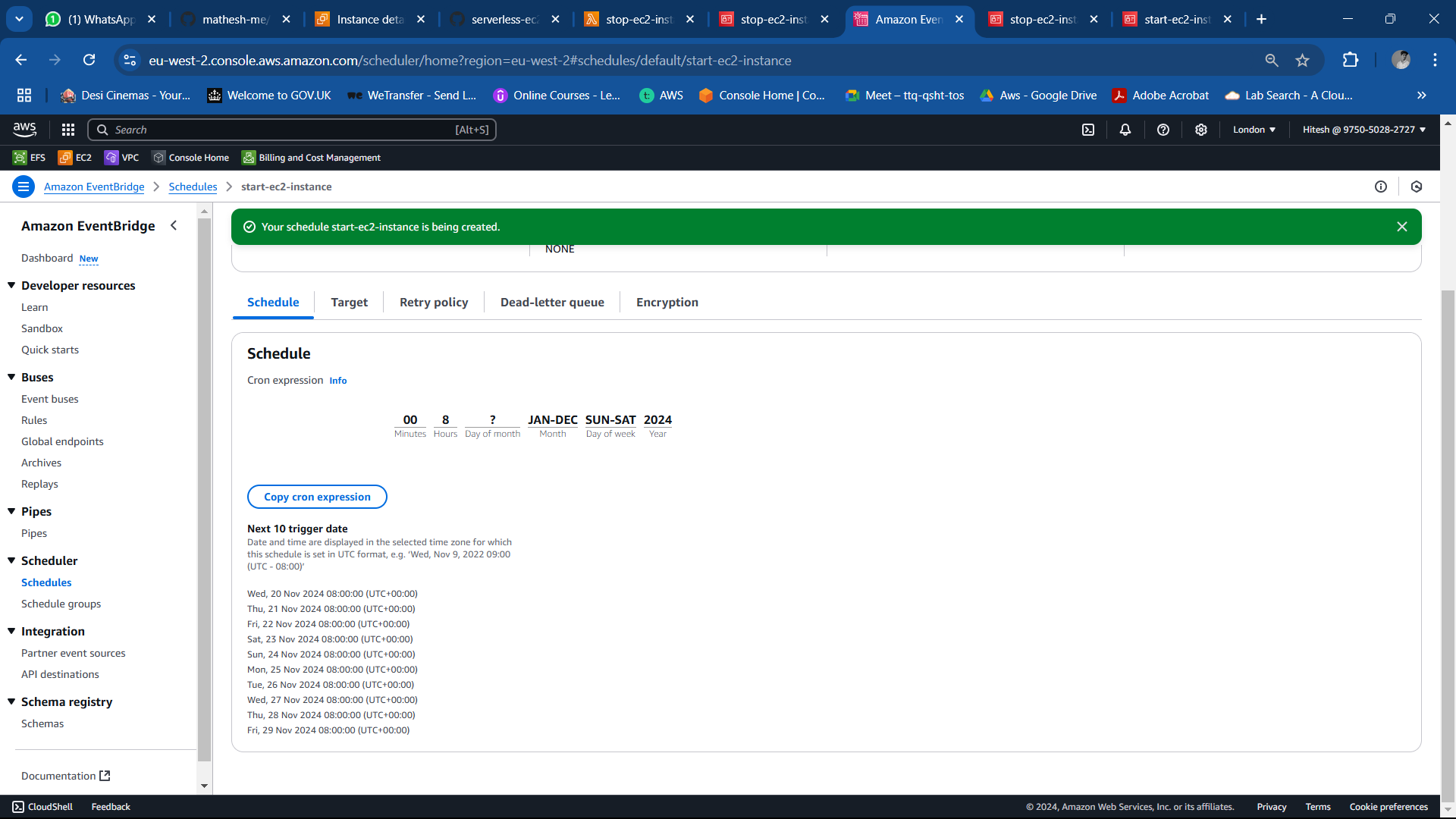


1. Create two schedules:
   * **Start Schedule**:
     + Rule Name: start-ec2-instance
     + Trigger: Lambda function for starting the instance.
     + Schedule Time: 8:00 AM daily.







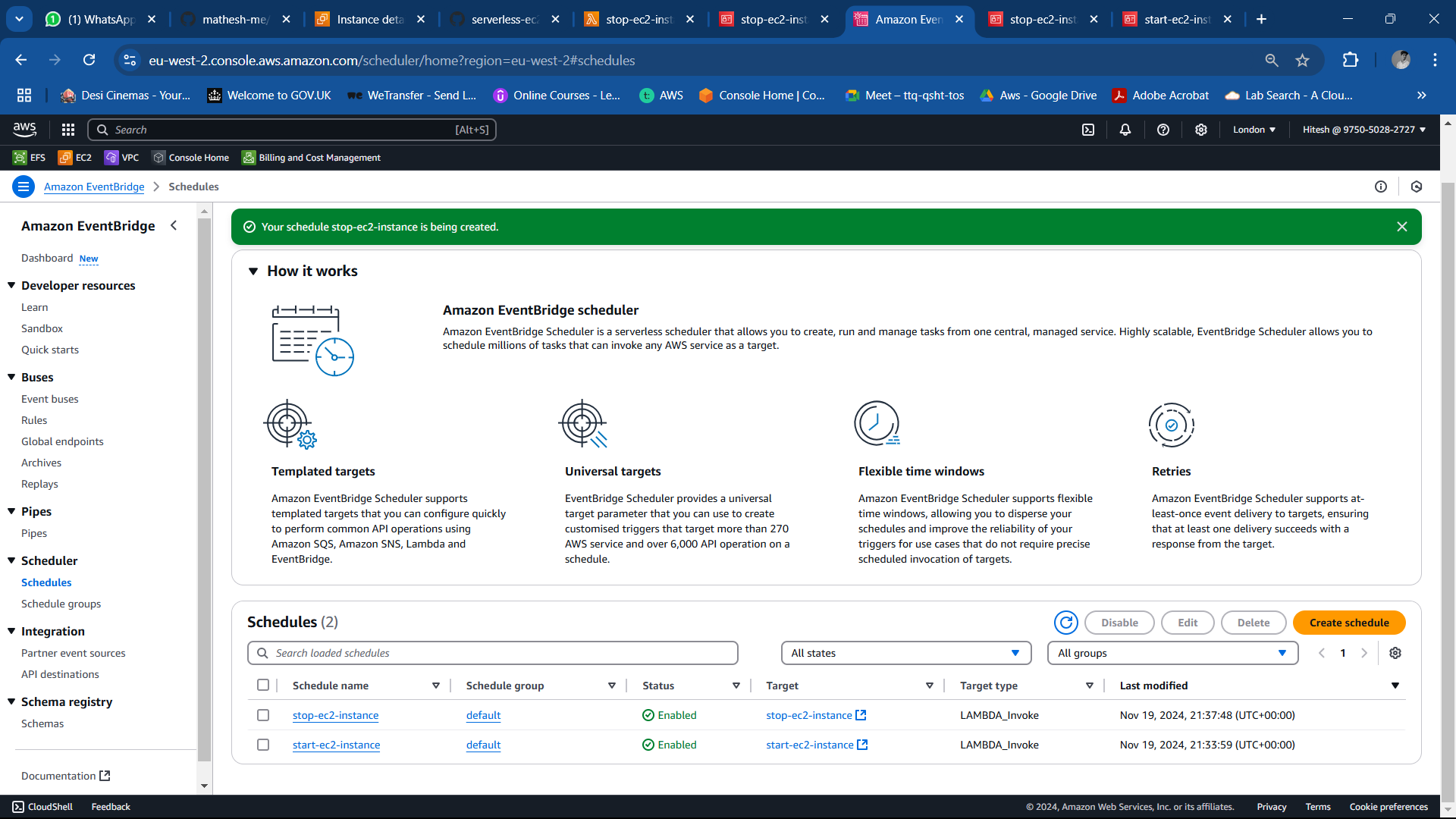


AS we can see I have created schedule to start my ec2 instance everyday at 08:00am

* + **Stop Schedule**:

Now we need to same step to stop the instance everyday at 05:00pm

* + - Rule Name: stop-ec2-instance
    - Trigger: Lambda function for stopping the instance.
    - Schedule Time: 5:00 PM daily.



### Final Outcome:

This setup ensures EC2 instances automatically start at 8:00 AM and stop at 5:00 PM, aligning with working hours. The serverless approach minimizes manual intervention and reduces operational costs.

LinkedIn Post

Using a Serverless EC2 Instance Scheduler to Reduce AWS Expenses! 🚀

It can be difficult to control cloud expenses, particularly when resources like EC2 instances are running continuously without need. In order to address this, I put in place a Serverless EC2 Instance Scheduler that makes sure instances only run from 8 AM to 5 PM.

This is how I did it:  
✅ Two AWS Lambda functions were developed to start and stop instances.  
✅ IAM policies were created specifically for EC2 actions such as StartInstances, StopInstances, and DescribeInstances.  
✅ CloudWatch Events was used for automated scheduling, which set off the functions at predetermined times.

Why this matters: This serverless solution is scalable and effective while lowering expenses, optimising resource usage, and doing away with manual intervention.

Technology Stack:  
  
Policies and Roles for AWS Lambda IAM  
Python for CloudWatch Events 🌟 While maintaining operational effectiveness, automating cloud resource management can result in significant cost savings. Please feel free to connect or leave a comment below if you're interested in investigating or putting similar solutions into practice!

#AWS #CloudComputing #Serverless #Automation #CostOptimization #CloudWatch #LambdaFunctions #CloudEngineering #DevOps #Python #CloudManagement #TechInnovation