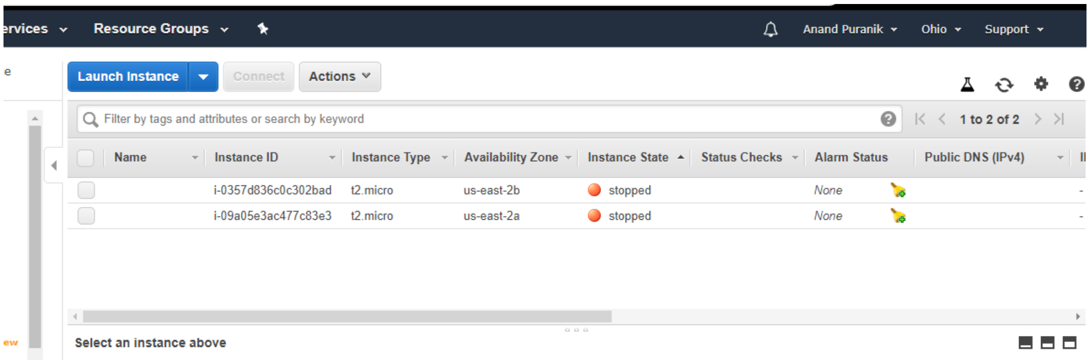
Cloudwatch logs Assiignment - 1

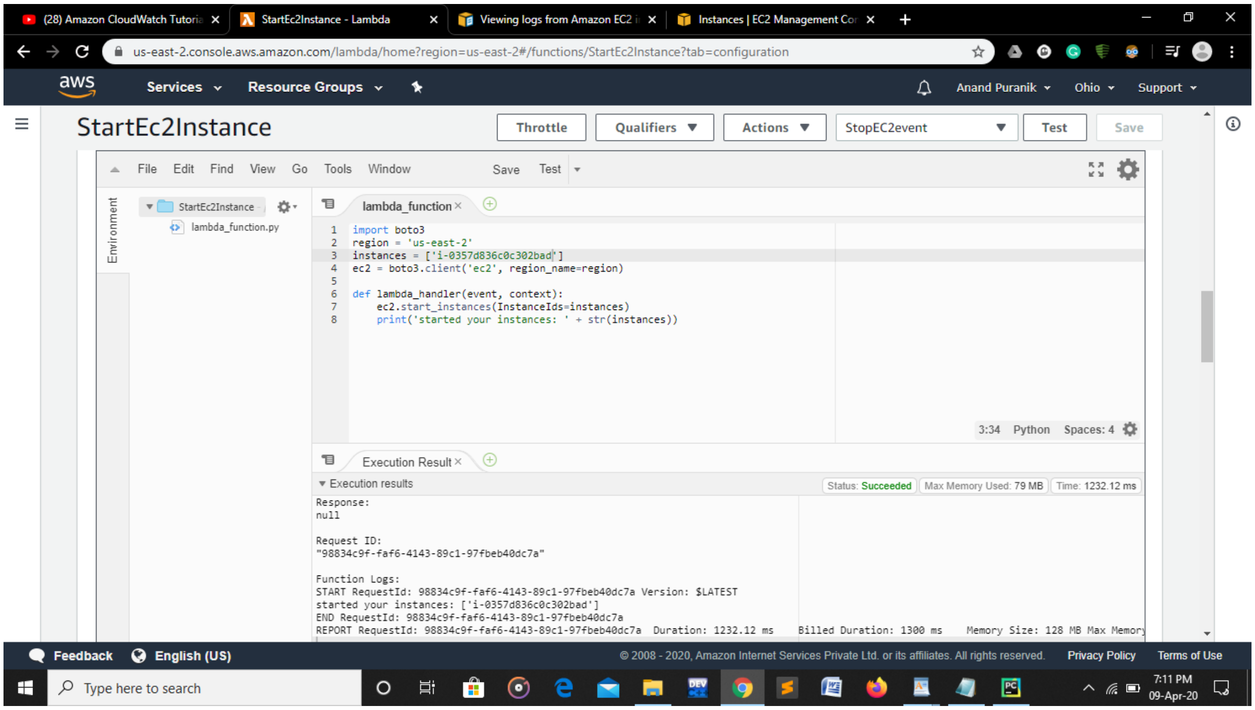
Create a lambda function to start, stop and terminate a running EC2 instance. Also, verify the logs generated by the lambda in the Cloudwatch logs to check whether the lambda execution is done properly.

1. Create an EC2 Instance



Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)
	i-0357d836c0c302bad	t2.micro	us-east-2b	● stopped		None	-
	i-09a05e3ac477c83e3	t2.micro	us-east-2a	● stopped		None	-

2. Create Two Lambda Functions to Start and Stop EC2 instance.



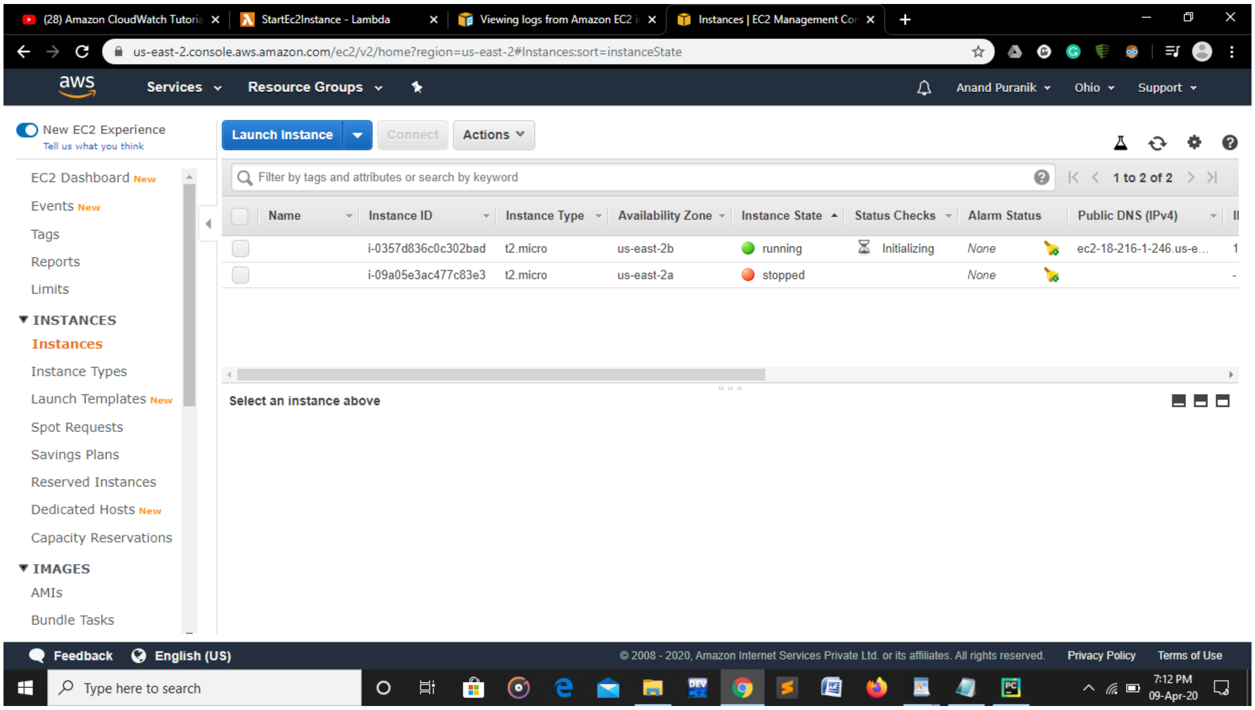
```python
import boto3
region = 'us-east-2'
instances = ['i-0357d836c0c302bad']
ec2 = boto3.client('ec2', region_name=region)

def lambda_handler(event, context):
    ec2.start_instances(InstanceIds=instances)
    print('started your instances: ' + str(instances))
```

Execution results

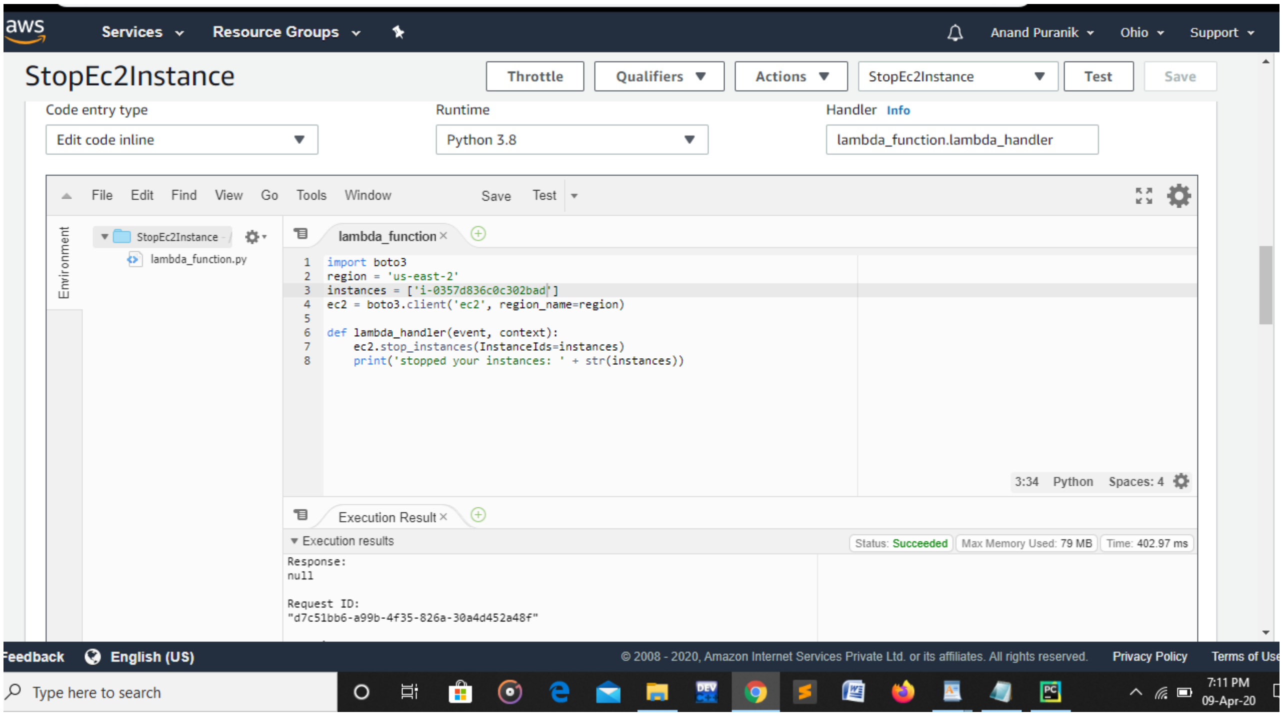
Status: Succeeded | Max Memory Used: 79 MB | Time: 1232.12 ms

```
Response:
null

Request ID:
"98834c9f-faf6-4143-89c1-97fbeb40dc7a"

Function Logs:
START RequestId: 98834c9f-faf6-4143-89c1-97fbeb40dc7a Version: $LATEST
started your instances: ['i-0357d836c0c302bad']
END RequestId: 98834c9f-faf6-4143-89c1-97fbeb40dc7a
REPORT RequestId: 98834c9f-faf6-4143-89c1-97fbeb40dc7a  Duration: 1232.12 ms    Billed Duration: 1300 ms    Memory Size: 128 MB Max Memory
```

3. The Ec2 instance is initiated after running start function of lambda



Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)
	i-0357d836c0c302bad	t2.micro	us-east-2b	running	Initializing	None	ec2-18-216-1-246.us-e...
	i-09a05e3ac477c83e3	t2.micro	us-east-2a	stopped		None	



Code entry type: Edit code inline

Runtime: Python 3.8

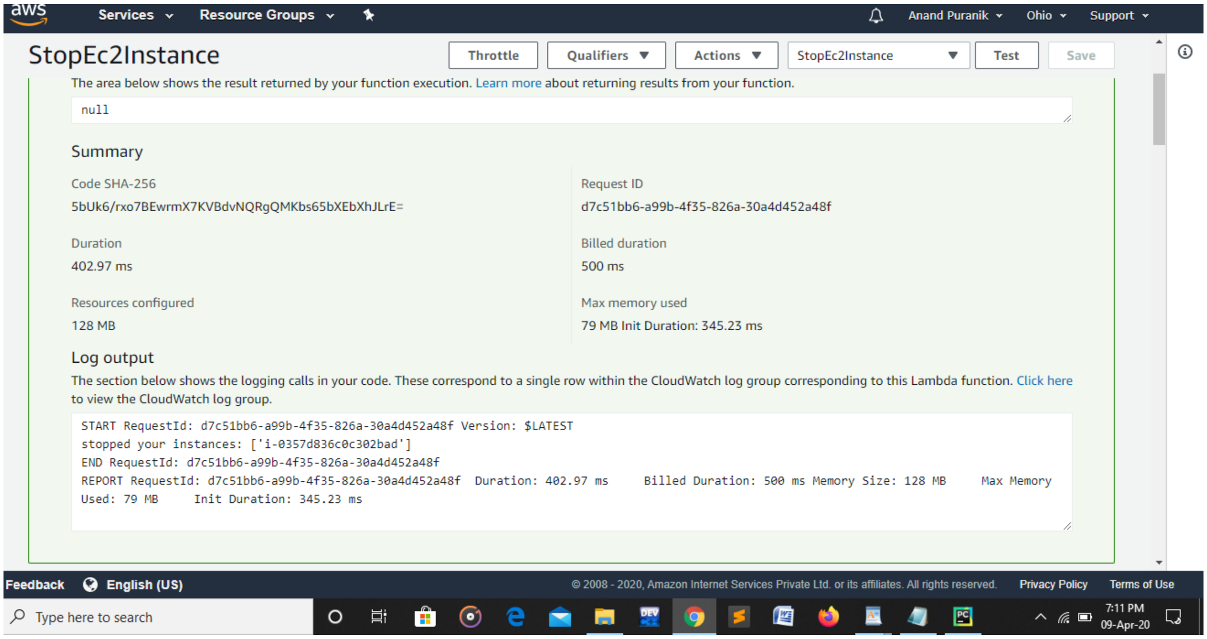
Handler: lambda_function.lambda_handler

```python
import boto3
region = 'us-east-2'
instances = ['i-0357d836c0c302bad']
ec2 = boto3.client('ec2', region_name=region)

def lambda_handler(event, context):
    ec2.stop_instances(InstanceIds=instances)
    print('stopped your instances: ' + str(instances))
```

3:34 Python Spaces: 4

Execution results — Status: Succeeded | Max Memory Used: 79 MB | Time: 402.97 ms

```
Response:
null

Request ID:
"d7c51bb6-a99b-4f35-826a-30a4d452a48f"
```



StopEc2Instance

The area below shows the result returned by your function execution. Learn more about returning results from your function.

```
null
```

Summary

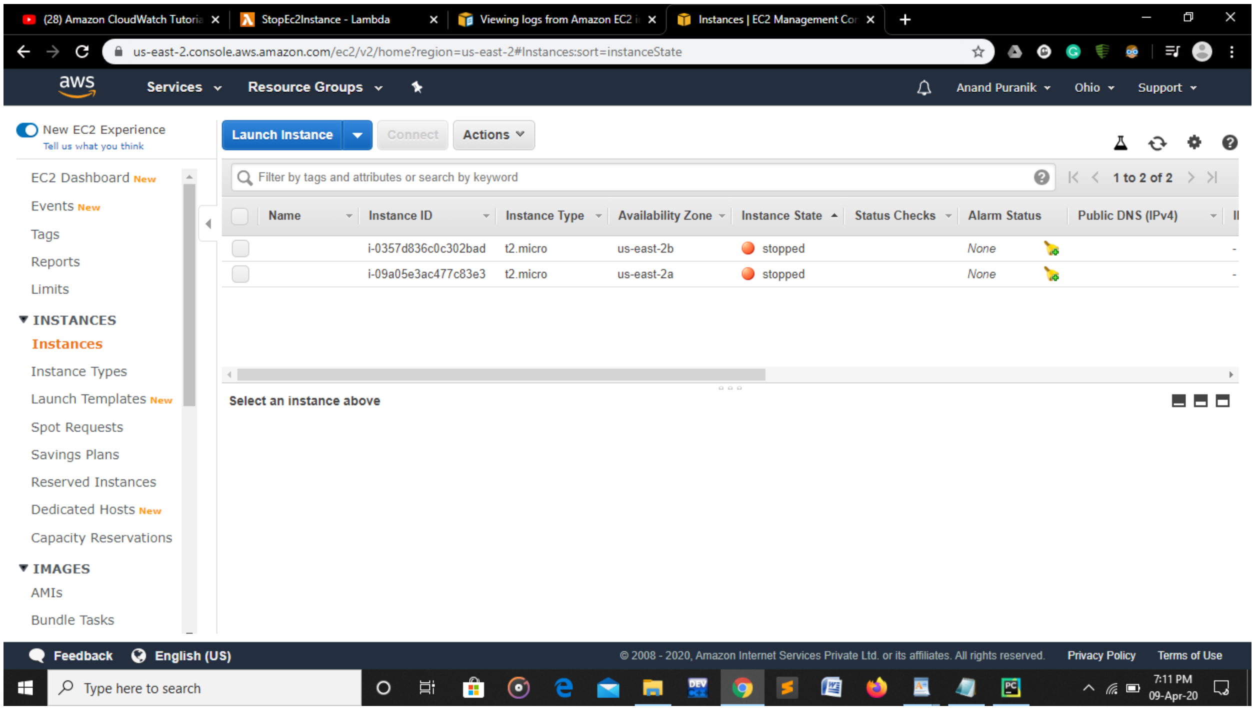
Code SHA-256 5bUk6/rxo7BEwrmX7KVBdvNQRgQMKbs65bXEbXhJLrE=	Request ID d7c51bb6-a99b-4f35-826a-30a4d452a48f
Duration 402.97 ms	Billed duration 500 ms
Resources configured 128 MB	Max memory used 79 MB Init Duration: 345.23 ms

Log output

The section below shows the logging calls in your code. These correspond to a single row within the CloudWatch log group corresponding to this Lambda function. Click here to view the CloudWatch log group.

```
START RequestId: d7c51bb6-a99b-4f35-826a-30a4d452a48f Version: $LATEST
stopped your instances: ['i-0357d836c0c302bad']
END RequestId: d7c51bb6-a99b-4f35-826a-30a4d452a48f
REPORT RequestId: d7c51bb6-a99b-4f35-826a-30a4d452a48f  Duration: 402.97 ms     Billed Duration: 500 ms Memory Size: 128 MB     Max Memory
Used: 79 MB     Init Duration: 345.23 ms
```

Ec2 instance is topped



Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)
	i-0357d836c0c302bad	t2.micro	us-east-2b	● stopped		None	-
	i-09a05e3ac477c83e3	t2.micro	us-east-2a	● stopped		None	-

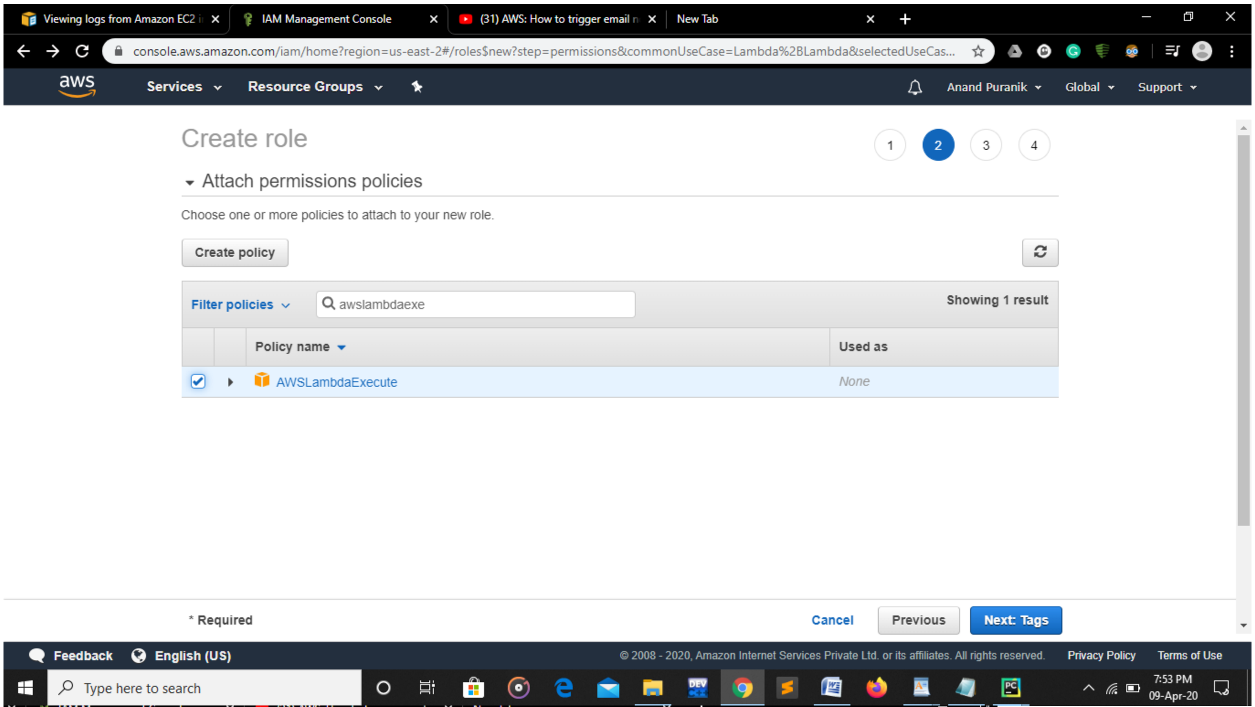
- Start Stop function Lambda
- start

-
  ```
  import boto3
  region = 'us-east-2'
  instances = ['i-0357d836c0c302bad']
  ec2 = boto3.client('ec2', region_name=region)

  def lambda_handler(event, context):
      ec2.start_instances(InstanceIds=instances)
      print('started your instances: ' + str(instances))
  ```

- stop
  ```
  import boto3
  region = 'us-east-2'
  instances = ['i-0357d836c0c302bad']
  ec2 = boto3.client('ec2', region_name=region)

  def lambda_handler(event, context):
      ec2.stop_instances(InstanceIds=instances)
      print('stopped your instances: ' + str(instances))
  ```

Assignment #3- Create a lambda function to send an email notification to your email id of your AWS account, as soon as you delete a file from an S3 bucket, mentioning the file name which is deleted.

1. Create a role and attach lambda function to it.



Viewing logs from Amazon EC2
IAM Management Console
(31) AWS: How to trigger email
New Tab
console.aws.amazon.com/iam/home?region=us-east-2#/roles$new?step=permissions&commonUseCase=Lambda%2BLambda&selectedUseCas...
aws
Services
Resource Groups
Anand Puranik
Global
Support

Create role

1 2 3 4

Attach permissions policies

Choose one or more policies to attach to your new role.

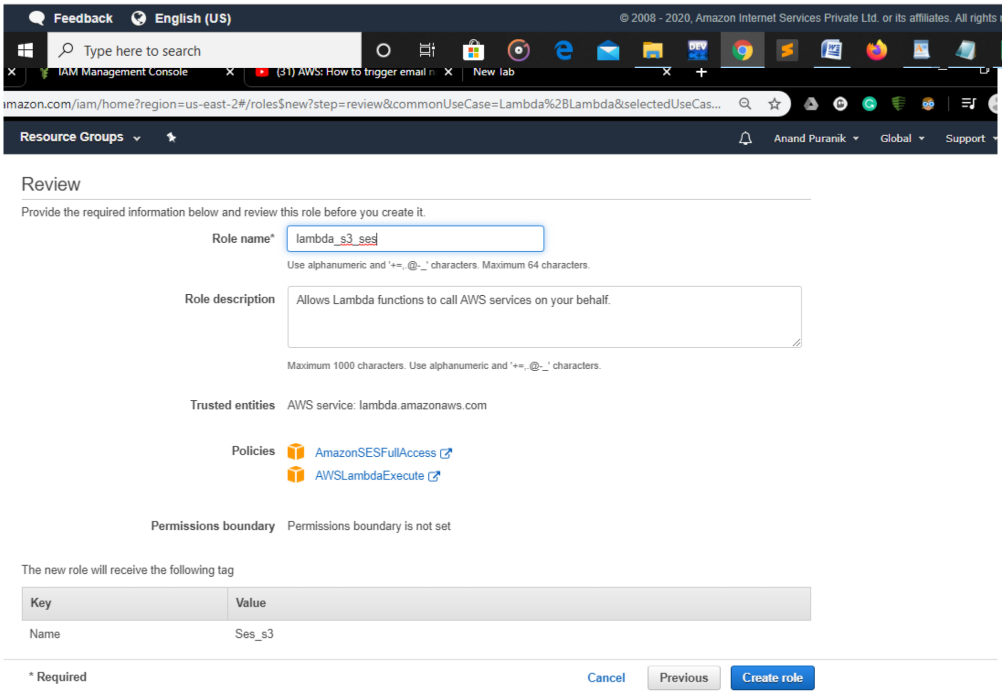
Create policy

Filter policies | awslambdaexe | Showing 1 result

		Policy name	Used as
☑	▸	AWSLambdaExecute	None

* Required

Cancel
Previous
Next: Tags
Feedback
English (US)
© 2008 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use
Type here to search
7:53 PM
09-Apr-20



IAM Management Console
(31) AWS: How to trigger email
New tab
amazon.com/iam/home?region=us-east-2#/roles$new?step=review&commonUseCase=Lambda%2BLambda&selectedUseCas...
Resource Groups
Anand Puranik
Global
Support

Review

Provide the required information below and review this role before you create it.

Role name* lambda_s3_ses

Use alphanumeric and '+=,.@-_' characters. Maximum 64 characters.

Role description Allows Lambda functions to call AWS services on your behalf.

Maximum 1000 characters. Use alphanumeric and '+=,.@-_' characters.

Trusted entities AWS service: lambda.amazonaws.com

Policies AmazonSESFullAccess
AWSLambdaExecute

Permissions boundary Permissions boundary is not set

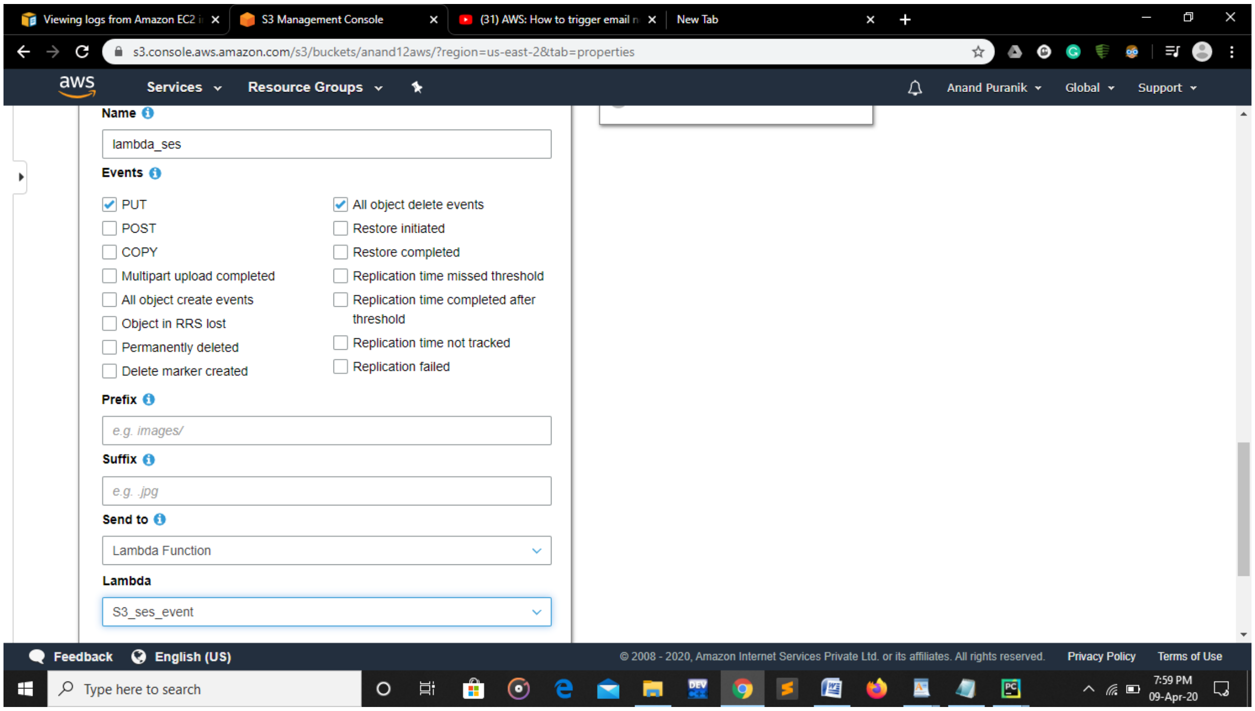
The new role will receive the following tag

Key	Value
Name	Ses_s3

* Required

A role is created

2. Choose a bucket -> Goto Properties -> Advance settings and choose Events


Viewing logs from Amazon EC2
S3 Management Console
(31) AWS: How to trigger email
New Tab
s3.console.aws.amazon.com/s3/buckets/anand12aws/?region=us-east-2&tab=properties
Services
Resource Groups
Anand Puranik
Global
Support

Name

lambda_ses

Events

☑ PUT	☑ All object delete events
□ POST	□ Restore initiated
□ COPY	□ Restore completed
□ Multipart upload completed	□ Replication time missed threshold
□ All object create events	□ Replication time completed after threshold
□ Object in RRS lost	□ Replication time not tracked
□ Permanently deleted	□ Replication failed
□ Delete marker created	

Prefix

e.g. images/

Suffix

e.g. .jpg

Send to

Lambda Function

Lambda

S3_ses_event

Feedback
English (US)
© 2008 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use
Type here to search
7:59 PM
09-Apr-20

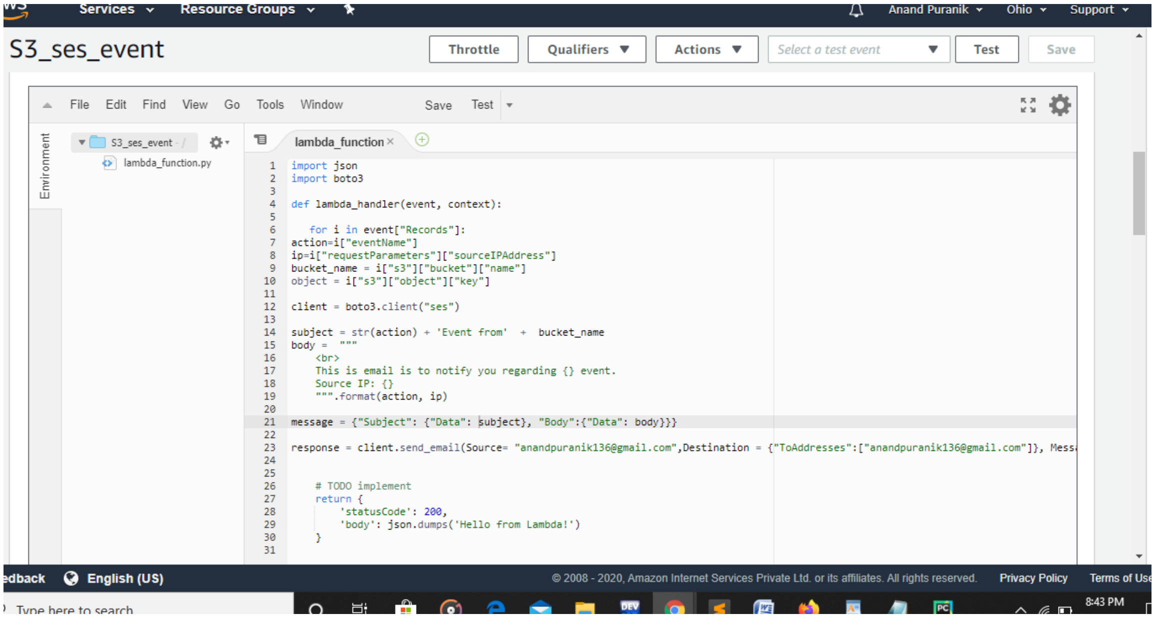
In events choose

Events- PUT

All objects delete event

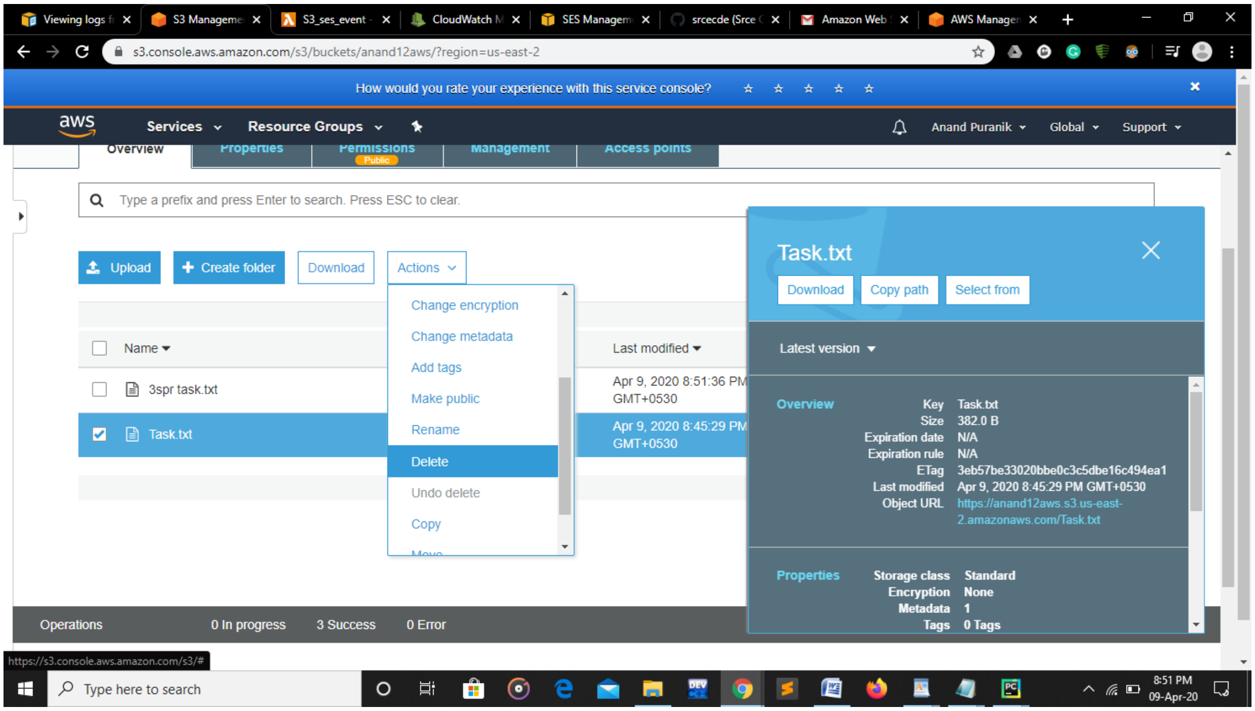
Attach lambda event

then go to lamda function and write the access code for the notifying mail for deleting any files from the s3 bucket:



```
import json
import boto3

def lambda_handler(event, context):

   for i in event["Records"]:
action=i["eventName"]
ip=i["requestParameters"]["sourceIPAddress"]
bucket_name = i["s3"]["bucket"]["name"]
object = i["s3"]["object"]["key"]

client = boto3.client("ses")

subject = str(action) + 'Event from'  +  bucket_name
body =  """
    <br>
    This is email is to notify you regarding {} event.
    Source IP: {}
    """.format(action, ip)

message = {"Subject": {"Data": subject}, "Body":{"Data": body}}}

response = client.send_email(Source= "anandpuranik136@gmail.com",Destination = {"ToAddresses":["anandpuranik136@gmail.com"]}, Mess


    # TODO implement
    return {
        'statusCode': 200,
        'body': json.dumps('Hello from Lambda!')
    }

```

Now save the code and test it by deleting any element from s3 storage and check for clouwatch logs to check execution of code



	Name	Last modified
□	3spr task.txt	Apr 9, 2020 8:51:36 PM GMT+0530
☑	Task.txt	Apr 9, 2020 8:45:29 PM GMT+0530

Task.txt

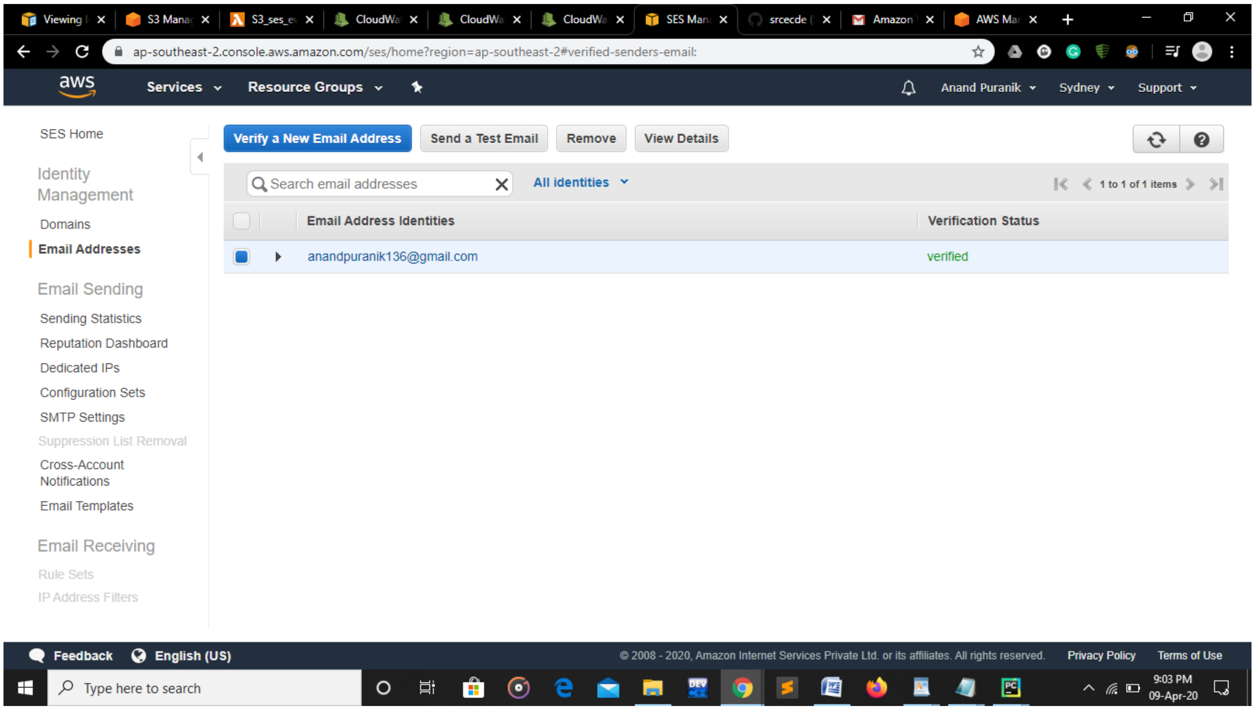
Overview

Key	Task.txt
Size	382.0 B
Expiration date	N/A
Expiration rule	N/A
ETag	3eb57be33020bbe0c3c5dbe16c494ea1
Last modified	Apr 9, 2020 8:45:29 PM GMT+0530
Object URL	https://anand12aws.s3.us-east-2.amazonaws.com/Task.txt

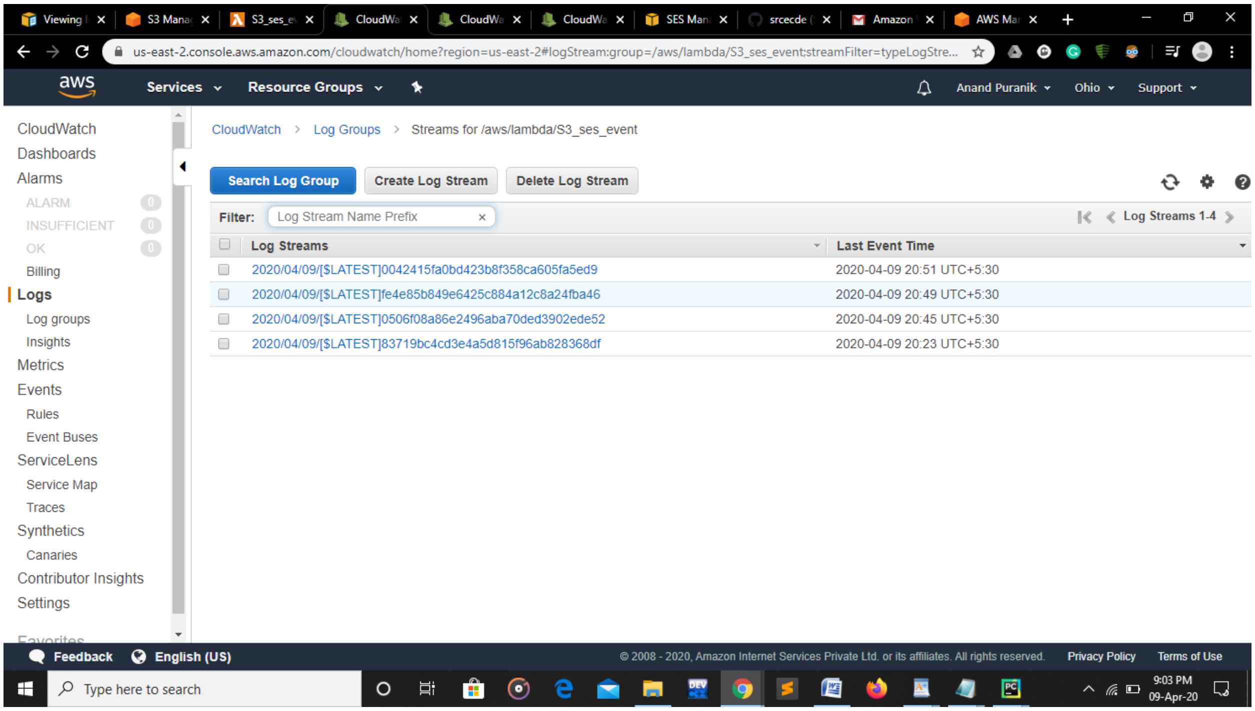
Properties

Storage class	Standard
Encryption	None
Metadata	1
Tags	0 Tags

go to simple email service to verify the email



	Email Address Identities	Verification Status
■	anandpuranik136@gmail.com	verified



CloudWatch > Log Groups > Streams for /aws/lambda/S3_ses_event

Log Streams	Last Event Time
2020/04/09/[$LATEST]0042415fa0bd423b8f358ca605fa5ed9	2020-04-09 20:51 UTC+5:30
2020/04/09/[$LATEST]fe4e85b849e6425c884a12c8a24fba46	2020-04-09 20:49 UTC+5:30
2020/04/09/[$LATEST]0506f08a86e2496aba70ded3902ede52	2020-04-09 20:45 UTC+5:30
2020/04/09/[$LATEST]83719bc4cd3e4a5d815f96ab828368df	2020-04-09 20:23 UTC+5:30

then go to cloudwatch and check whether deleting notification is accessing or not