Cloud Cost Optimization — *AWS*

Used the combination of AWS cloud watch and Lambda functions To decrease the cloud cost by 25 percent.

Implementation:

Created lambda function in python, used the boto3 module to interact With AWS service APIs and this lambda function is triggered by Cloud Watch events. This lambda function would watch for any unused EBS snapshots and Either delete them or send out notification to the snapshot owner.

AWS Cloud Cost Optimization - Identifying Stale Resources

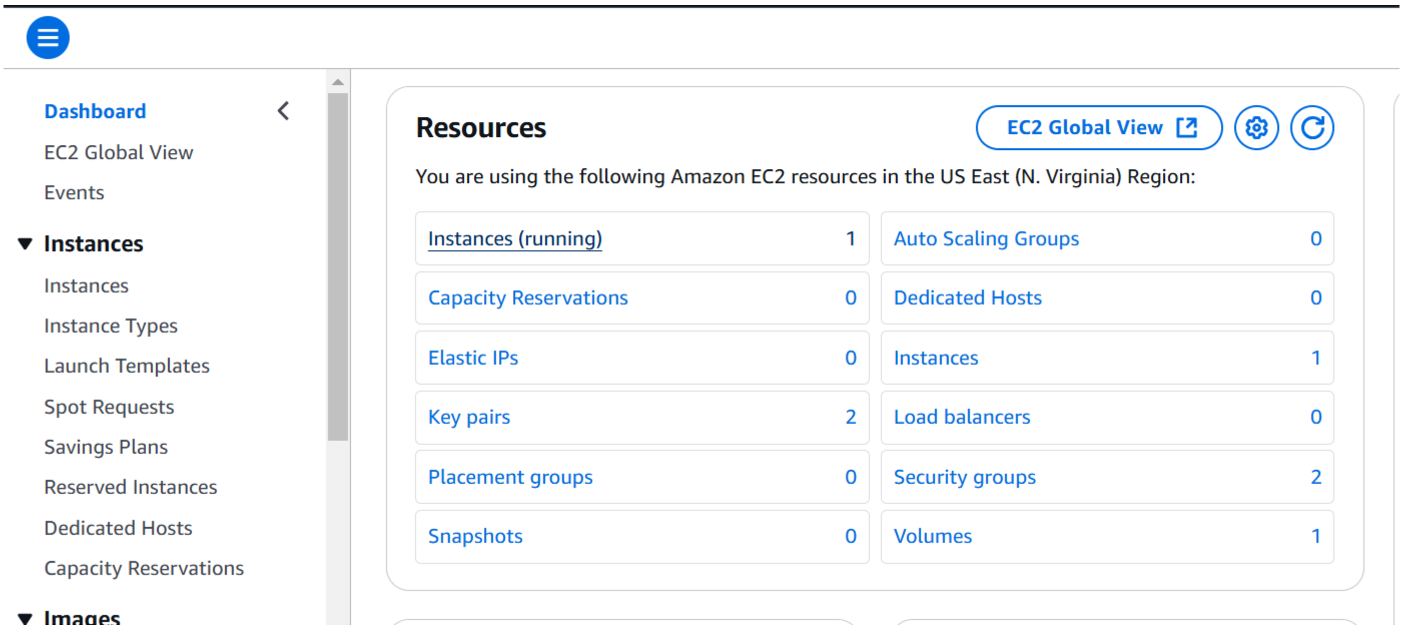
Identifying Stale EBS Snapshots

In this example, we'll create a Lambda function that identifies EBS snapshots that are no longer associated with any active EC2 instance and deletes them to save on storage costs.

Description:

The Lambda function fetches all EBS snapshots owned by the same account ('self') and also retrieves a list of active EC2 instances (running and stopped). For each snapshot, it checks if the associated volume (if exists) is not associated with any active instance. If it finds a stale snapshot, it deletes it, effectively optimizing storage costs.

1. Go to Aws console, check the resources (instances, volumes, snapshots...etc) in EC2 instances



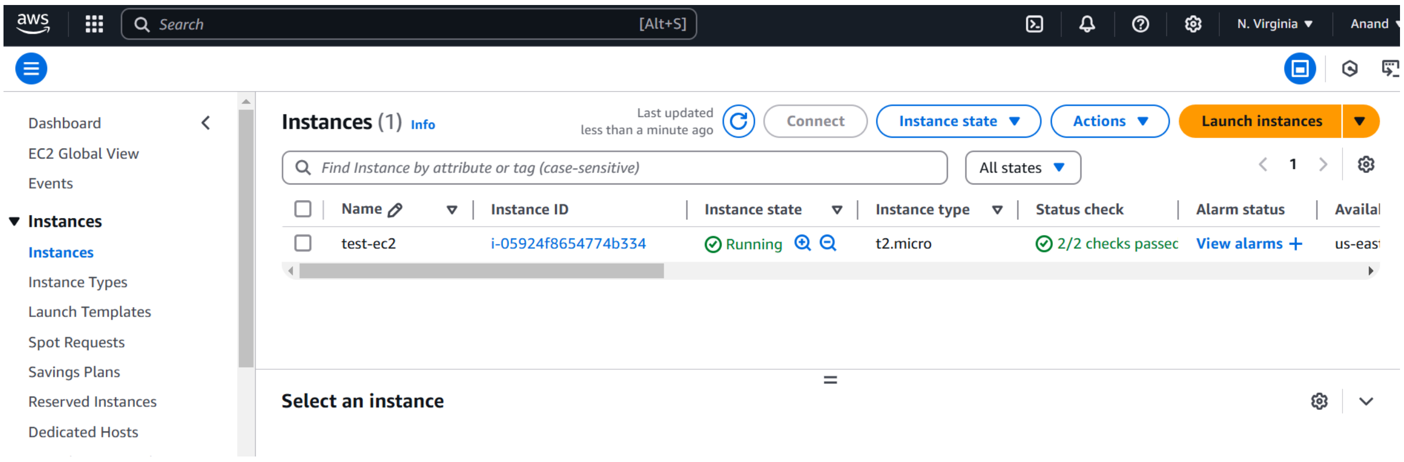
Resources

You are using the following Amazon EC2 resources in the US East (N. Virginia) Region:

Instances (running)	1	Auto Scaling Groups	0
Capacity Reservations	0	Dedicated Hosts	0
Elastic IPs	0	Instances	1
Key pairs	2	Load balancers	0
Placement groups	0	Security groups	2
Snapshots	0	Volumes	1

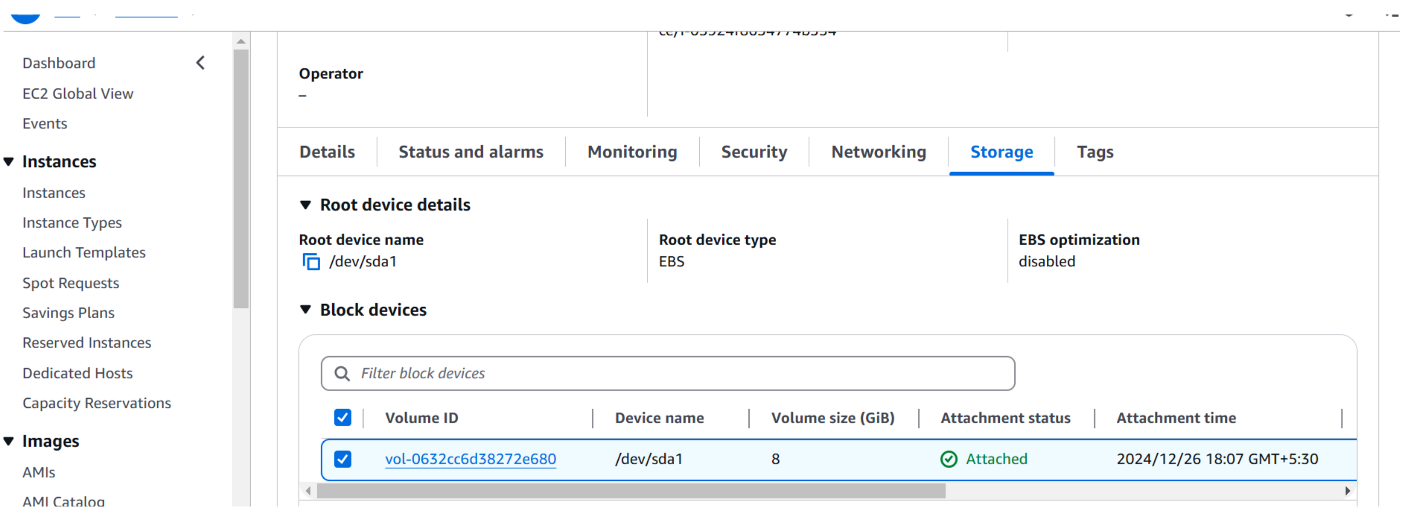
{Because I don’t have any Stale Resources in my AWS account for the purpose of this project demonstration, so I have created a Stale Resources i.e, from 2 - }

2. Created a EC2 instance by default it will add a volume and launch instance.



	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availal
☐	test-ec2	i-05924f8654774b334	Running	t2.micro	2/2 checks passec	View alarms +	us-eas

3. Click on instance id go to storage you find volume id of the created instance.



Operator
–

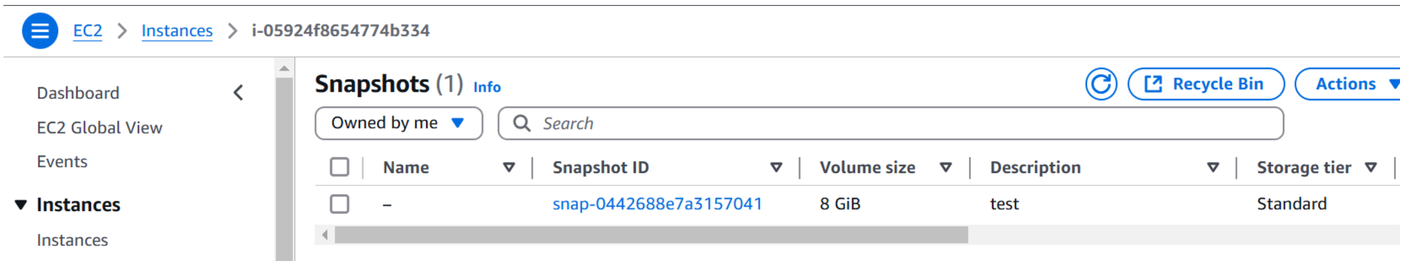
Root device details

Root device name: /dev/sda1
Root device type: EBS
EBS optimization: disabled

Block devices

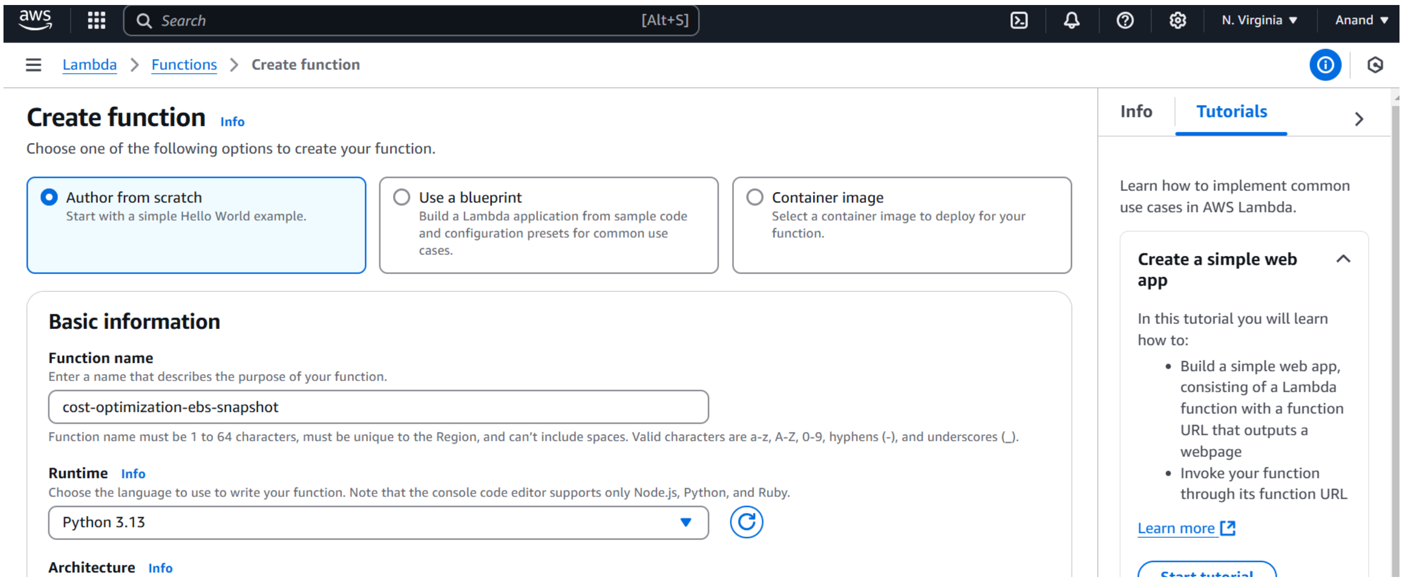
	Volume ID	Device name	Volume size (GiB)	Attachment status	Attachment time
☑	vol-0632cc6d38272e680	/dev/sda1	8	Attached	2024/12/26 18:07 GMT+5:30

4. Go back to the EC2 dashboard and click on snapshots and create snapshot.



	Name	Snapshot ID	Volume size	Description	Storage tier
☐	–	snap-0442688e7a3157041	8 GiB	test	Standard

5. Go to lambda function and create a lambda function.



Create function Info

Choose one of the following options to create your function.

- [✓ Author from scratch] Start with a simple Hello World example.
- Use a blueprint: Build a Lambda application from sample code and configuration presets for common use cases.
- Container image: Select a container image to deploy for your function.

Basic information

Function name
Enter a name that describes the purpose of your function.
cost-optimization-ebs-snapshot
Function name must be 1 to 64 characters, must be unique to the Region, and can't include spaces. Valid characters are a-z, A-Z, 0-9, hyphens (-), and underscores (_).

Runtime Info
Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.
Python 3.13

Architecture Info

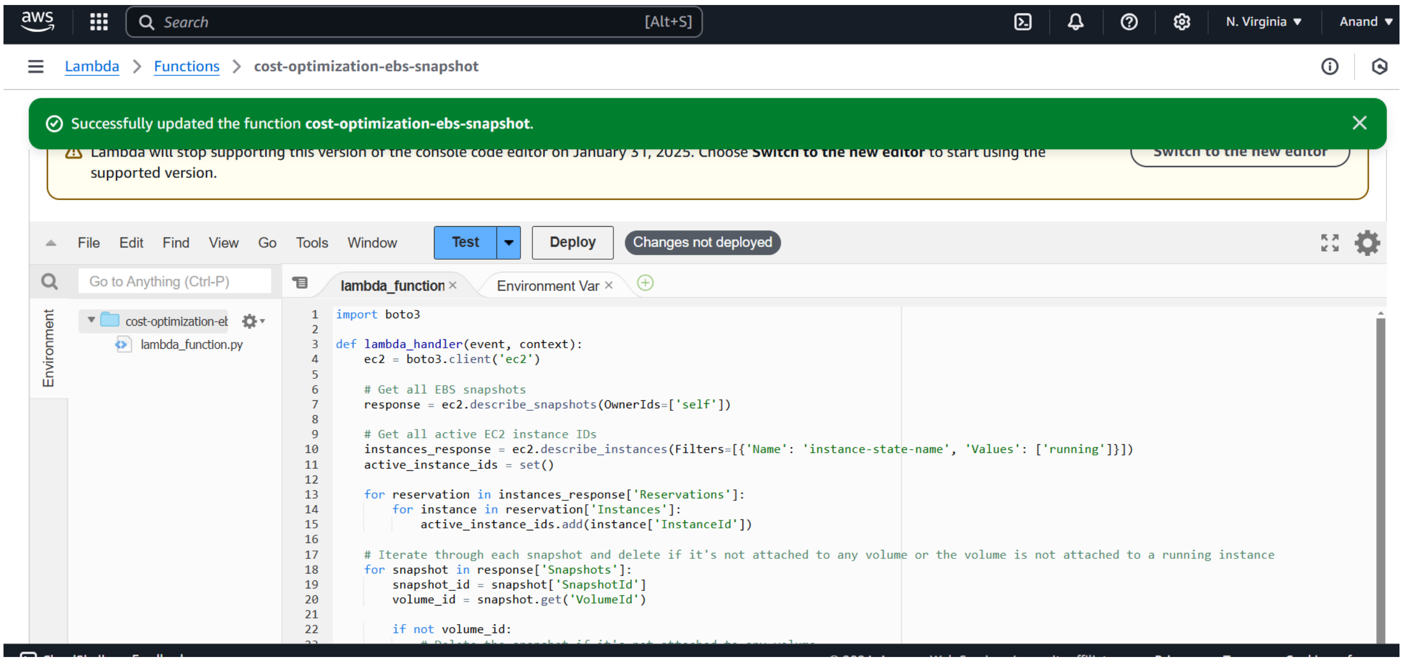
Learn how to implement common use cases in AWS Lambda.

Create a simple web app

In this tutorial you will learn how to:
- Build a simple web app, consisting of a Lambda function with a function URL that outputs a webpage
- Invoke your function through its function URL

Learn more

6. You can write your own lambda function code if you know, else go to this gitHub .
7. Remove the existing code and past the code, save this, give a Test name and click on Deploy.

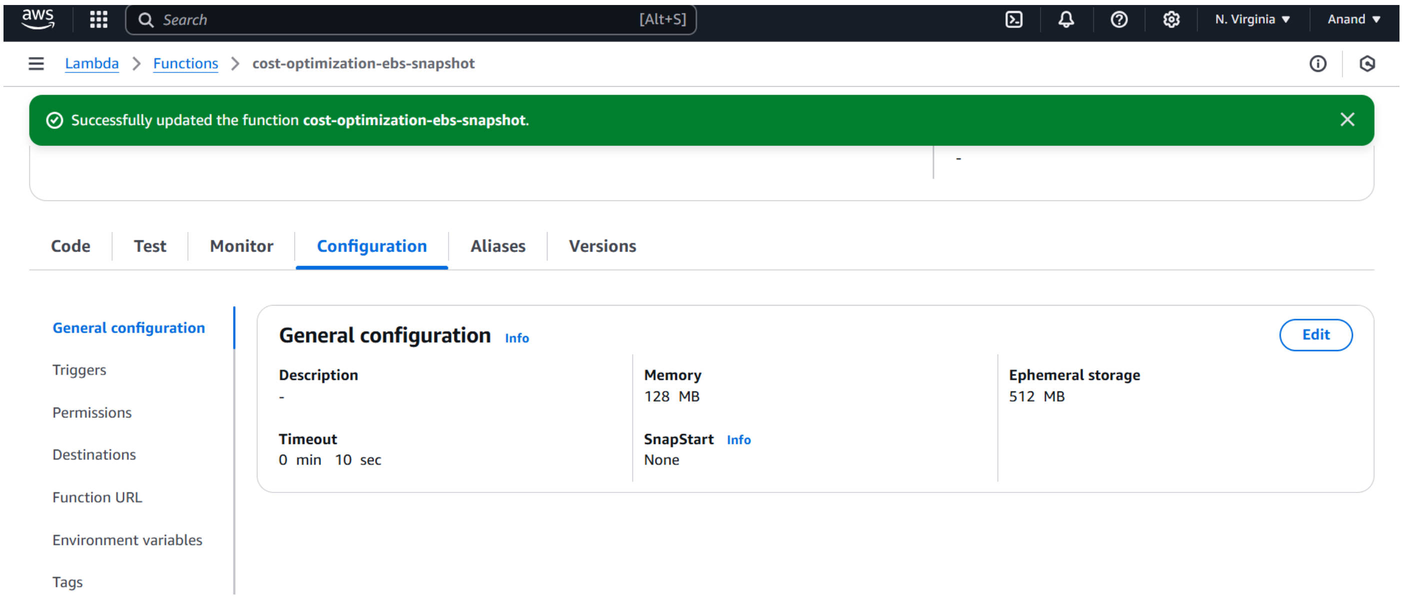


Successfully updated the function **cost-optimization-ebs-snapshot**.

Lambda will stop supporting this version of the console code editor on January 31, 2025. Choose **Switch to the new editor** to start using the supported version.

```python
import boto3

def lambda_handler(event, context):
    ec2 = boto3.client('ec2')

    # Get all EBS snapshots
    response = ec2.describe_snapshots(OwnerIds=['self'])

    # Get all active EC2 instance IDs
    instances_response = ec2.describe_instances(Filters=[{'Name': 'instance-state-name', 'Values': ['running']}])
    active_instance_ids = set()

    for reservation in instances_response['Reservations']:
        for instance in reservation['Instances']:
            active_instance_ids.add(instance['InstanceId'])

    # Iterate through each snapshot and delete if it's not attached to any volume or the volume is not attached to a running instance
    for snapshot in response['Snapshots']:
        snapshot_id = snapshot['SnapshotId']
        volume_id = snapshot.get('VolumeId')

        if not volume_id:
```

8. Now configure the timeout to 10sec


Lambda > Functions > cost-optimization-ebs-snapshot

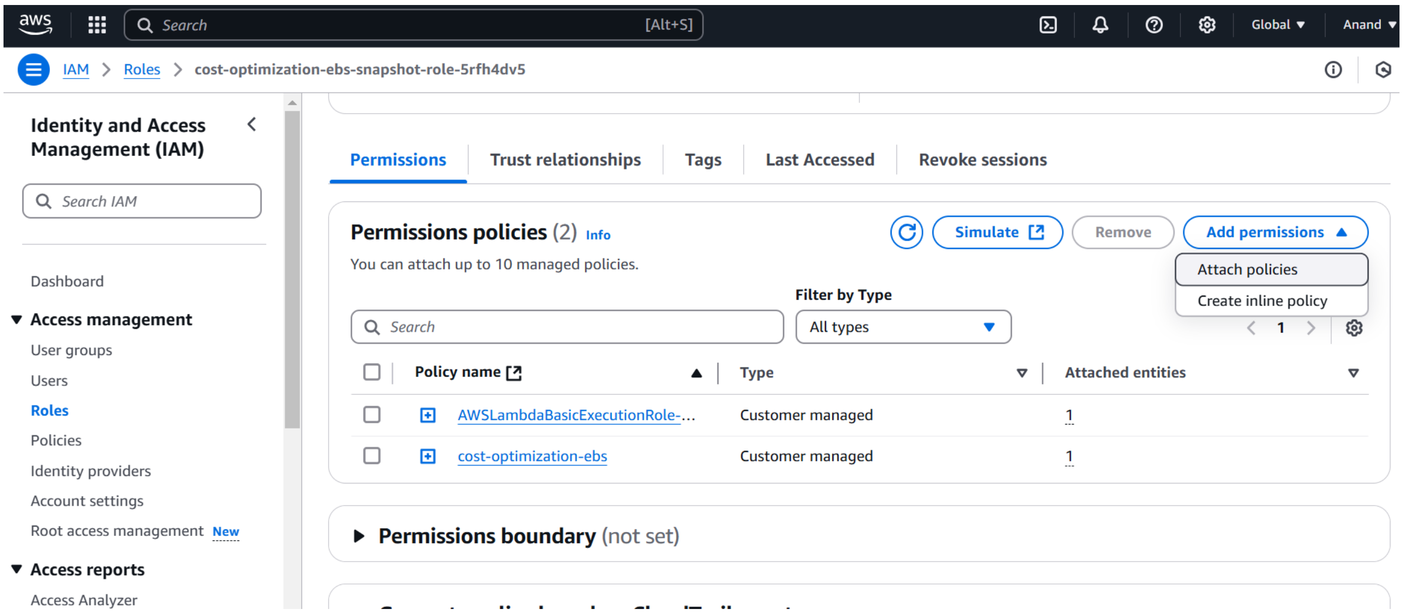
Successfully updated the function **cost-optimization-ebs-snapshot**.

General configuration Info

Description	Memory	Ephemeral storage
-	128 MB	512 MB
Timeout	**SnapStart**	
0 min 10 sec	None	

9. Go to permissions and click on Role name and add the permissions if you want you can create policies or add existing policies. Here I am creating my policies suitable for my project(snapshot permissions, EC2, Volume) and select resources all and name to the policy, create it.

10. Now attached the created policies and add the permission.

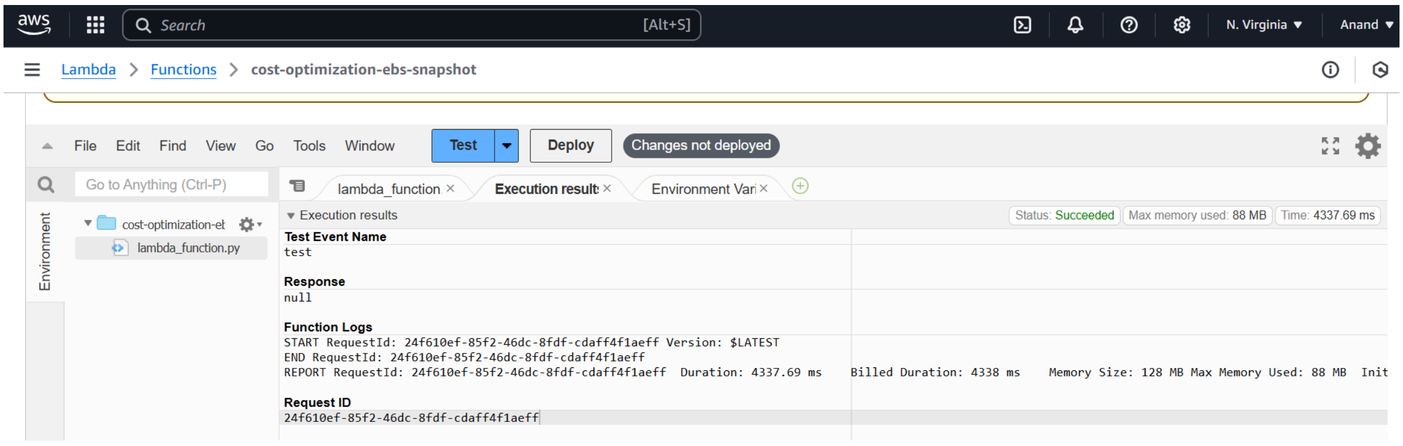

IAM > Roles > cost-optimization-ebs-snapshot-role-5rfh4dv5

Permissions policies (2) Info

You can attach up to 10 managed policies.

Policy name	Type	Attached entities
AWSLambdaBasicExecutionRole-...	Customer managed	1
cost-optimization-ebs	Customer managed	1

11. Click on the Test you will see like this below if you don't have stale resources.


Lambda > Functions > cost-optimization-ebs-snapshot
File Edit Find View Go Tools Window Test Deploy Changes not deployed
Go to Anything (Ctrl-P)
lambda_function
Execution result
Environment Vari
Environment
cost-optimization-et
lambda_function.py

Execution results

Status: Succeeded | Max memory used: 88 MB | Time: 4337.69 ms

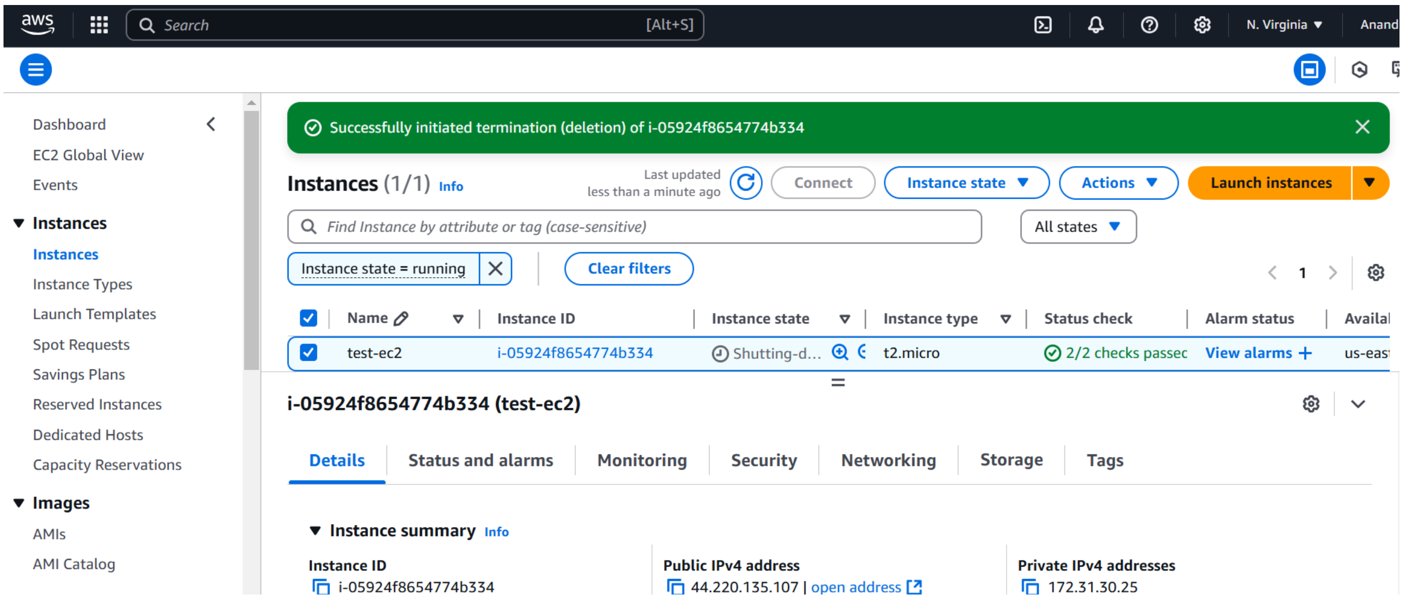
Test Event Name
```
test
```

Response
```
null
```

Function Logs
```
START RequestId: 24f610ef-85f2-46dc-8fdf-cdaff4f1aeff Version: $LATEST
END RequestId: 24f610ef-85f2-46dc-8fdf-cdaff4f1aeff
REPORT RequestId: 24f610ef-85f2-46dc-8fdf-cdaff4f1aeff  Duration: 4337.69 ms    Billed Duration: 4338 ms    Memory Size: 128 MB Max Memory Used: 88 MB  Init
```

Request ID
```
24f610ef-85f2-46dc-8fdf-cdaff4f1aeff
```

12. For project purpose I will delete the instance, but the snapshot is still present. Now I have stale resource.



Successfully initiated termination (deletion) of i-05924f8654774b334

Instances (1/1) Info — Last updated less than a minute ago

Find Instance by attribute or tag (case-sensitive)

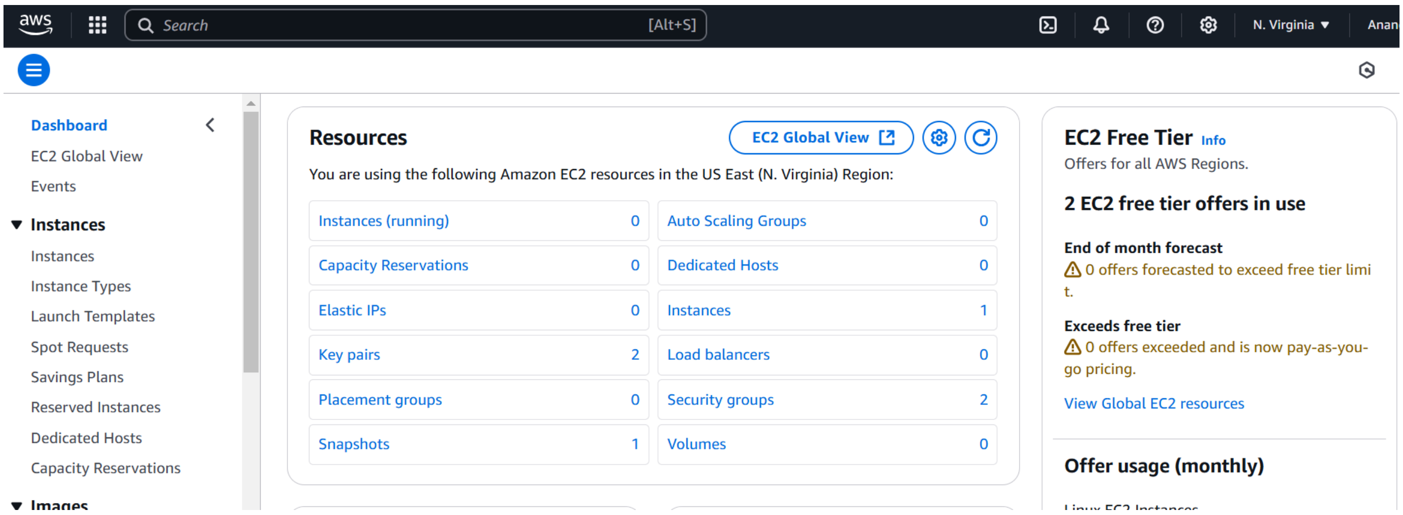
Instance state = running

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availa
test-ec2	i-05924f8654774b334	Shutting-d...	t2.micro	2/2 checks passec	View alarms +	us-eas

i-05924f8654774b334 (test-ec2)

Instance summary Info

Instance ID	Public IPv4 address	Private IPv4 addresses
i-05924f8654774b334	44.220.135.107 | open address	172.31.30.25



Resources

You are using the following Amazon EC2 resources in the US East (N. Virginia) Region:

Instances (running)	0	Auto Scaling Groups	0
Capacity Reservations	0	Dedicated Hosts	0
Elastic IPs	0	Instances	1
Key pairs	2	Load balancers	0
Placement groups	0	Security groups	2
Snapshots	1	Volumes	0

EC2 Free Tier Info

Offers for all AWS Regions.

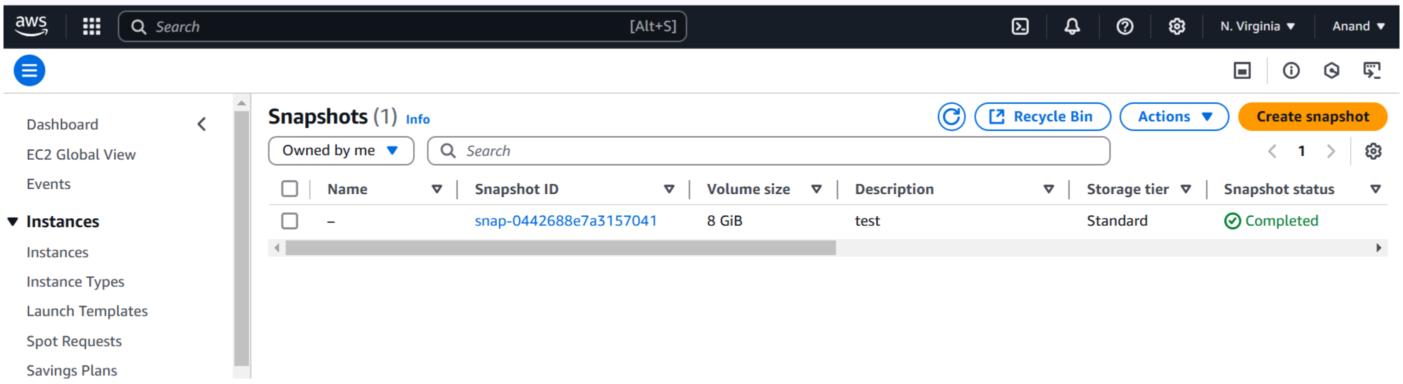
2 EC2 free tier offers in use

End of month forecast
0 offers forecasted to exceed free tier limi t.

Exceeds free tier
0 offers exceeded and is now pay-as-you-go pricing.

View Global EC2 resources

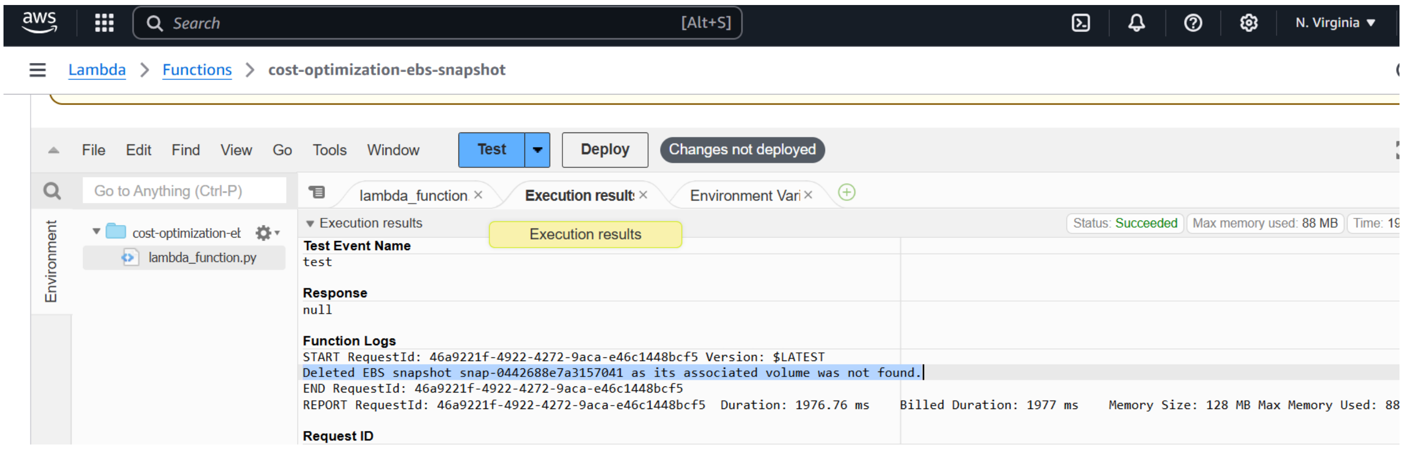
Offer usage (monthly)



Snapshots (1) Info

Name	Snapshot ID	Volume size	Description	Storage tier	Snapshot status
–	snap-0442688e7a3157041	8 GiB	test	Standard	Completed

13. Now again Test the code, the stale resource is deleted.



Execution results

Status: Succeeded | Max memory used: 88 MB

Test Event Name

```
test
```

Response

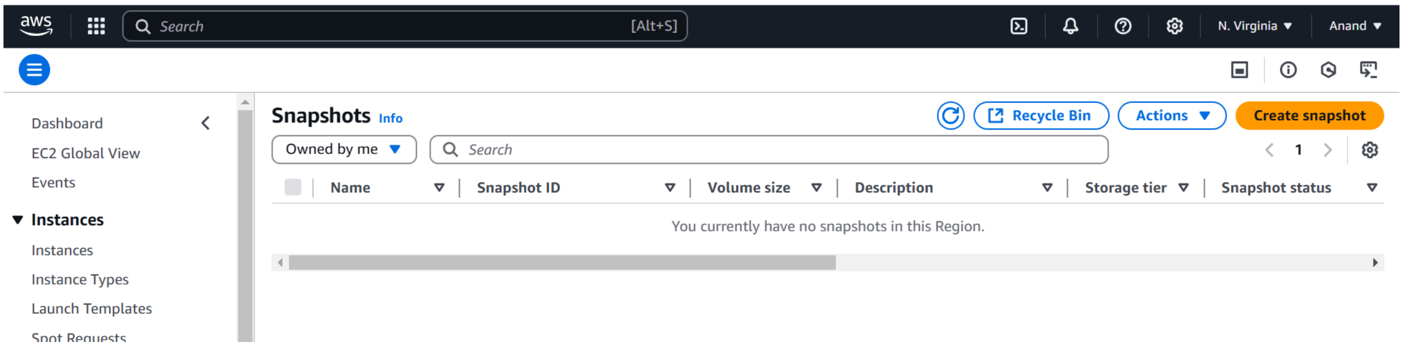
```
null
```

Function Logs

```
START RequestId: 46a9221f-4922-4272-9aca-e46c1448bcf5 Version: $LATEST
Deleted EBS snapshot snap-0442688e7a3157041 as its associated volume was not found.
END RequestId: 46a9221f-4922-4272-9aca-e46c1448bcf5
REPORT RequestId: 46a9221f-4922-4272-9aca-e46c1448bcf5  Duration: 1976.76 ms  Billed Duration: 1977 ms   Memory Size: 128 MB Max Memory Used: 88
```

Request ID

14. The stale snapshot is deleted here and project is completed.



Snapshots Info

Name	Snapshot ID	Volume size	Description	Storage tier	Snapshot status

You currently have no snapshots in this Region.

NOTE: For Hand on experience I used my AWS free tier account to do this while learning, if you also using Free tier account, make sure delete all the things you performed.