

Cost Optimization

Python Script:

```
import boto3

def lambda_handler(event, context):

    ec2 = boto3.client('ec2')

    # Get all EBS snapshots

    snap_response = ec2.describe_snapshots(OwnerIds=['self'])

    #Get all volumes

    volumes = ec2.describe_volumes(Filters=[{'Name': 'status', 'Values': ['available']}])

    available_volume = set()

    #delete volume if its not in use.

    if not volumes['Volumes']:

        print("No available volumes found.")

    else:

        print("Available volumes:")

        for volume in volumes['Volumes']:

            volume_id = volume['VolumeId']

            print(f"Volume ID: {volume_id}")

            ec2.delete_volume(VolumeId=volume_id)

            print(f"Volume ID: {volume_id} is deleted.")


    # 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 snap_response['Snapshots']:
```

```
    snapshot_id = snapshot['SnapshotId']
```

```
    volume_id = snapshot.get('VolumeId')
```

```
    if not volume_id:
```

```
        # Delete the snapshot if it's not attached to any volume
```

```
        ec2.delete_snapshot(SnapshotId=snapshot_id)
```

```
        print(f"Deleted EBS snapshot {snapshot_id} as it was not attached to any volume.")
```

```
    else:
```

```
        # Check if the volume still exists
```

```
        try:
```

```
            volume_response = ec2.describe_volumes(VolumeIds=[volume_id])
```

```
            if not volume_response['Volumes'][0]['Attachments']:
```

```
                ec2.delete_snapshot(SnapshotId=snapshot_id)
```

```
                print(f"Deleted EBS snapshot {snapshot_id} as it was taken from a volume not attached to any running instance.")
```

```
        except ec2.exceptions.ClientError as e:
```

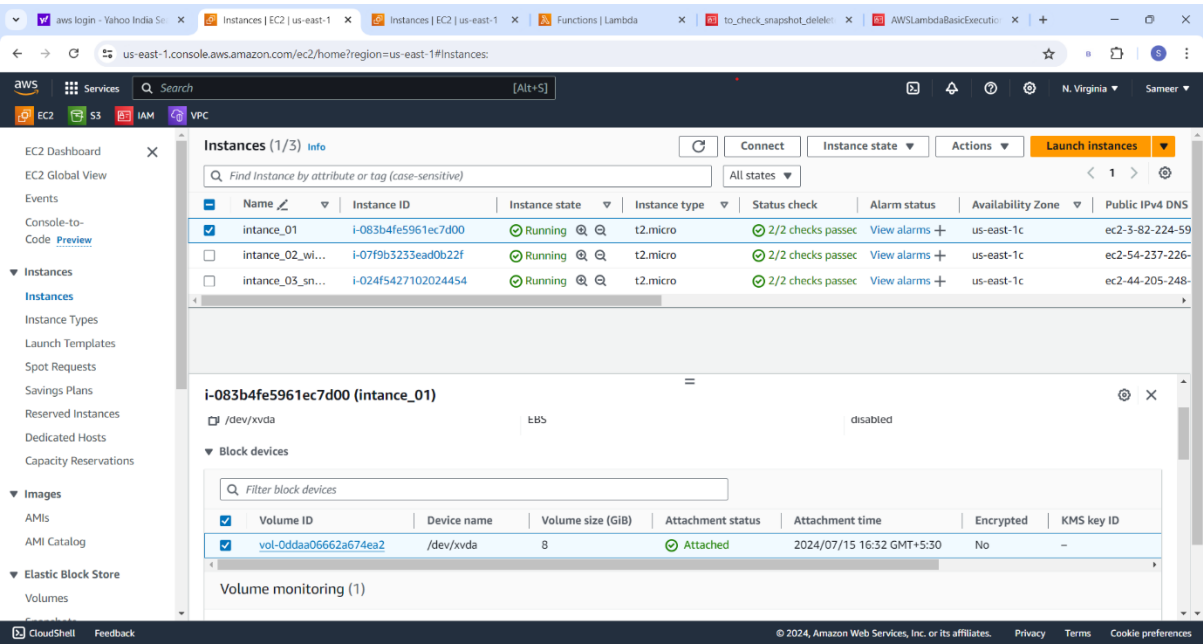
```
            if e.response['Error']['Code'] == 'InvalidVolume.NotFound':
```

```
                # The volume associated with the snapshot is not found (it might have been deleted)
```

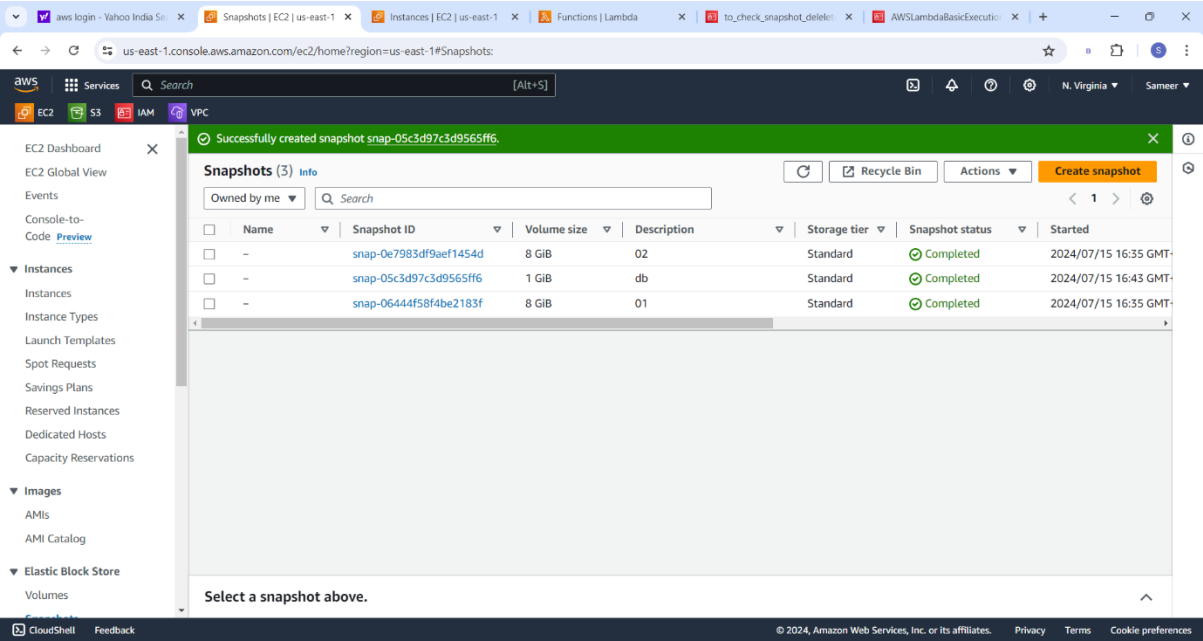
```
                ec2.delete_snapshot(SnapshotId=snapshot_id)
```

```
                print(f"Deleted EBS snapshot {snapshot_id} as its associated volume was not found.")
```

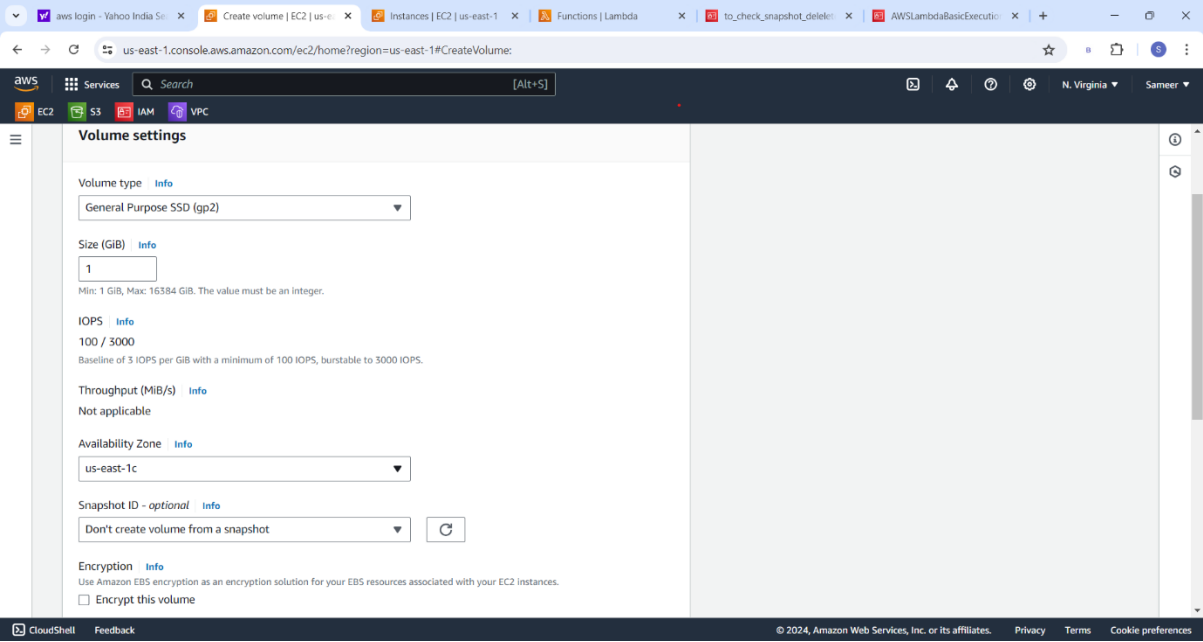
Create three instances (just for testing purposes).



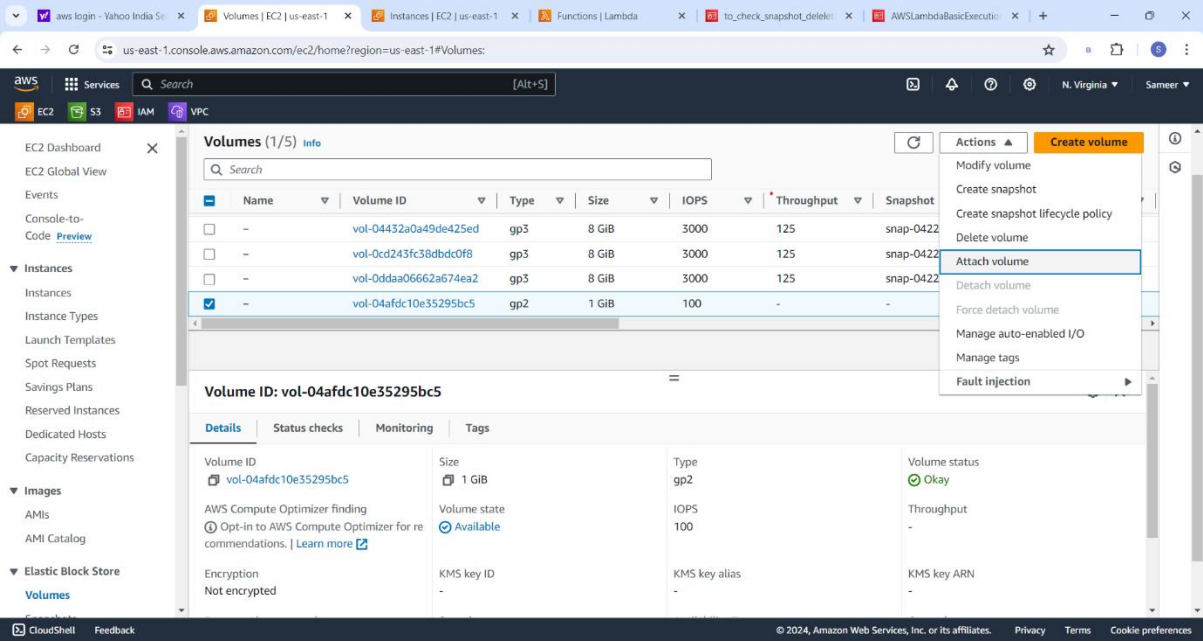
Create a snapshot of the existing volume.

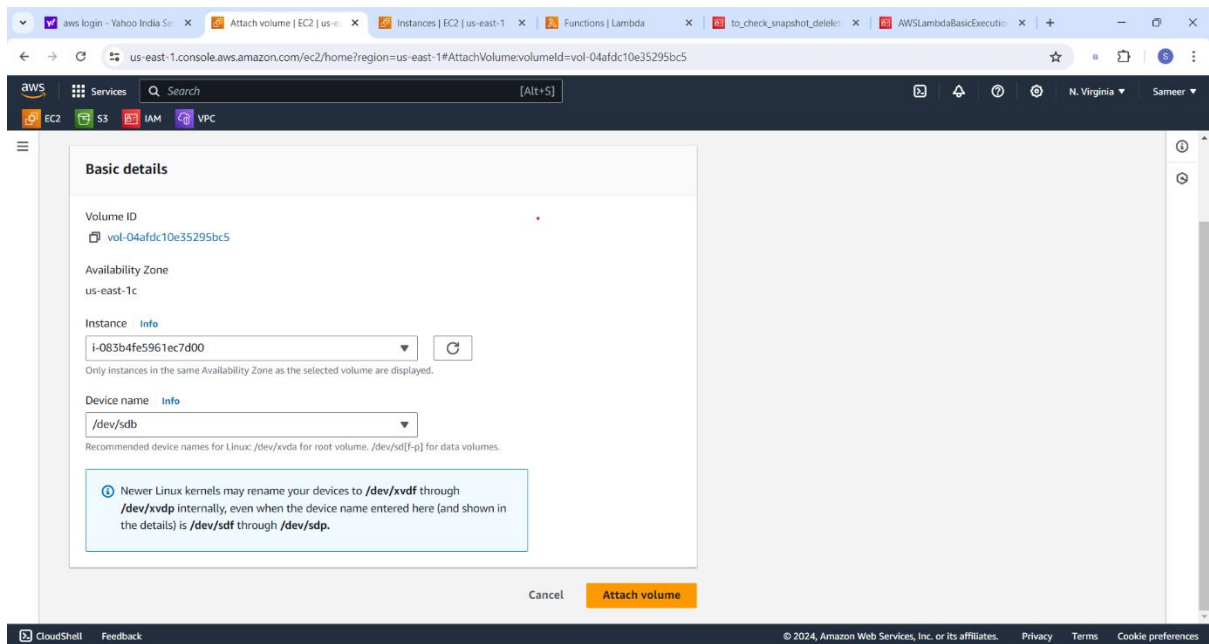


Create a volume.

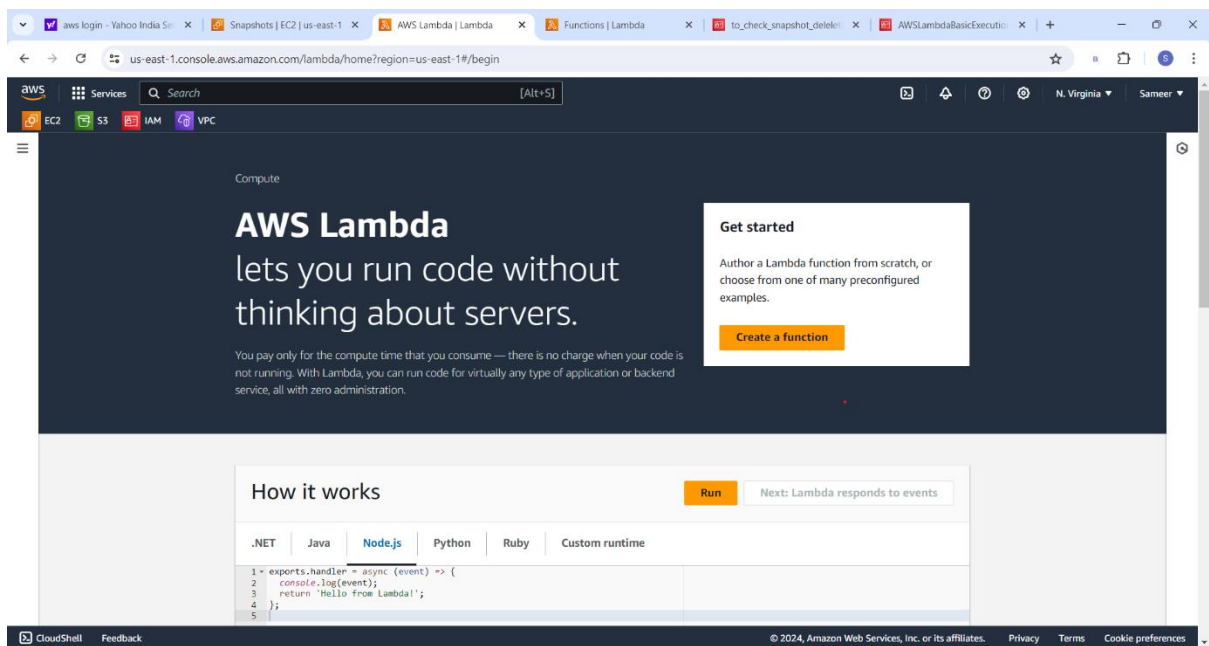


Attach the volume to the instance.





Create a Lambda function



Choose Python as the programming language

us-east-1.console.aws.amazon.com/lambda/home?region=us-east-1#/create/function?firstrun=true

Services Search [Alt+S]

EC2 S3 IAM VPC

☒ 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.

Use only letters, numbers, hyphens, or underscores with no spaces.

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

Architecture [Info](#)
Choose the instruction set architecture you want for your function code.
☒ x86_64
☐ arm64

Permissions [Info](#)
By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Here is the dashboard of my function.

us-east-1.console.aws.amazon.com/lambda/home?region=us-east-1#/functions/check_snapshot_volume?newfunction=true&tab=code

Services Search [Alt+S]

EC2 S3 IAM VPC

Successfully created the function **check_snapshot_volume**. You can now change its code and configuration. To invoke your function with a test event, choose "Test".

Lambda > Functions > check_snapshot_volume

check_snapshot_volume Throttle Copy ARN Actions

Function overview [Info](#) Export to Application Composer Download

Diagram Template

check_snapshot_volume

Layers (0)

+ Add trigger + Add destination

Description -

Last modified 5 seconds ago

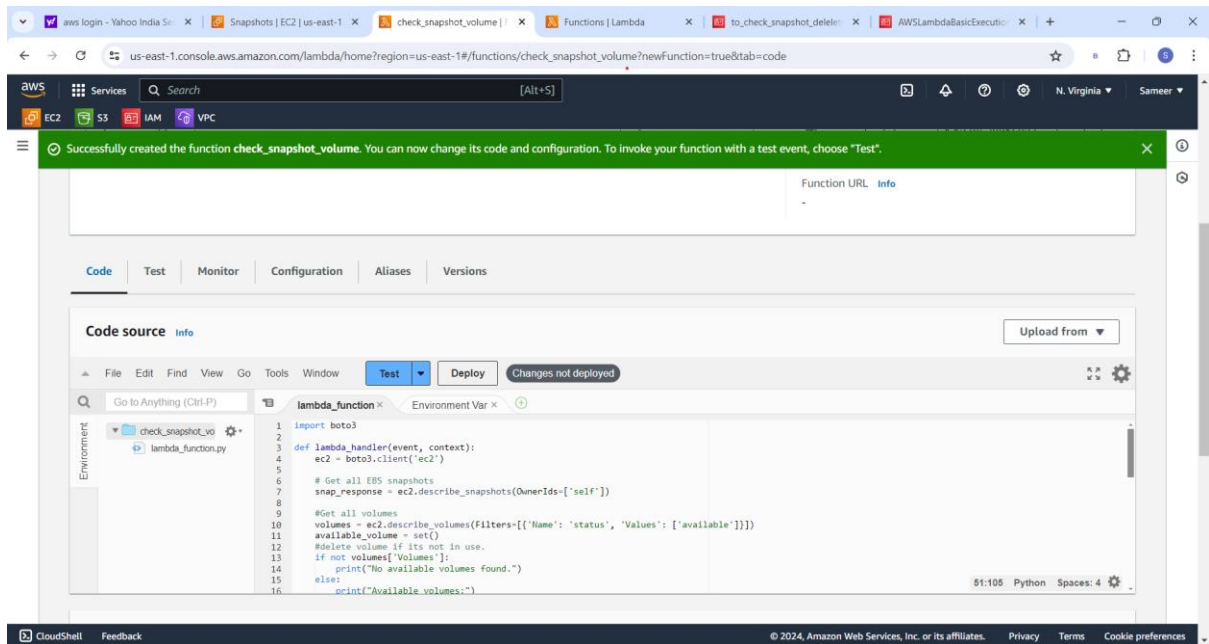
Function ARN arn:aws:lambda:us-east-1:123057001602:function:check_snapshot_volume

Function URL [Info](#)

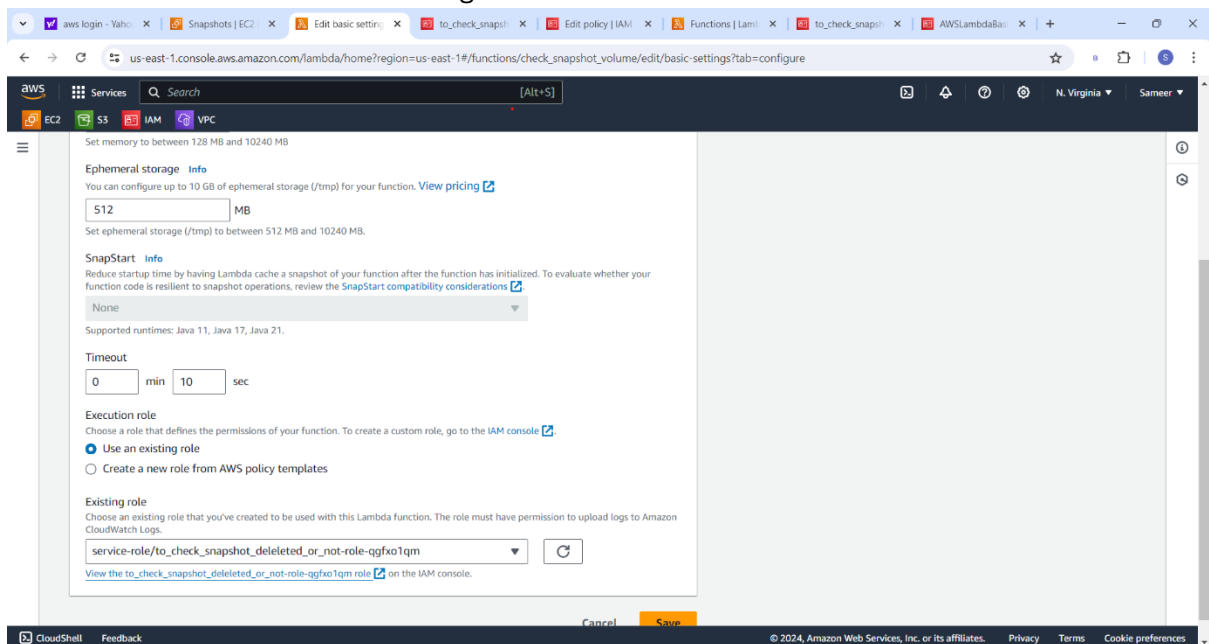
Code Test Monitor Configuration Aliases Versions

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

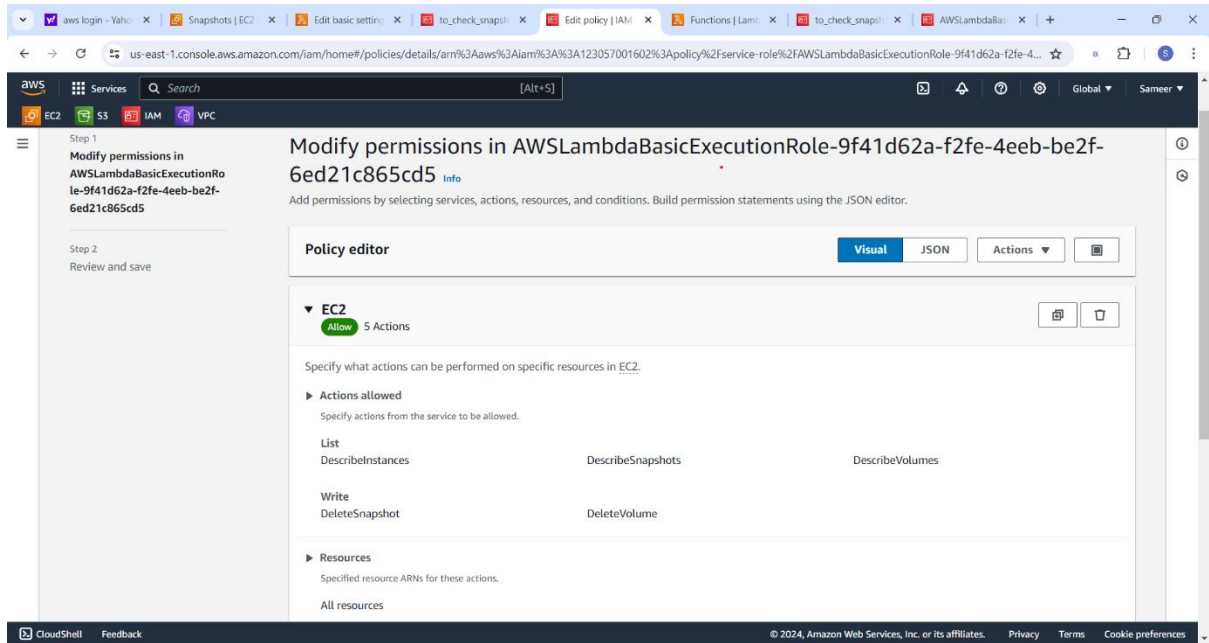
Scroll down, and in the code section, write a function to delete unused volumes and snapshots.



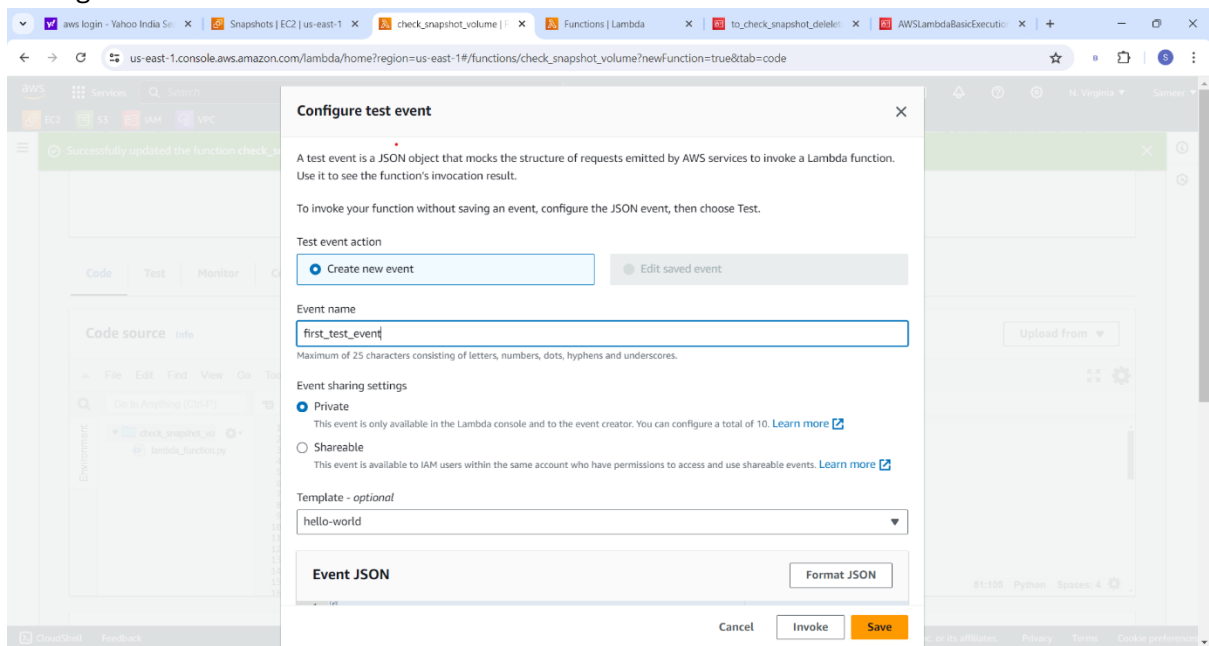
Increase the timeout to allow for longer execution time.



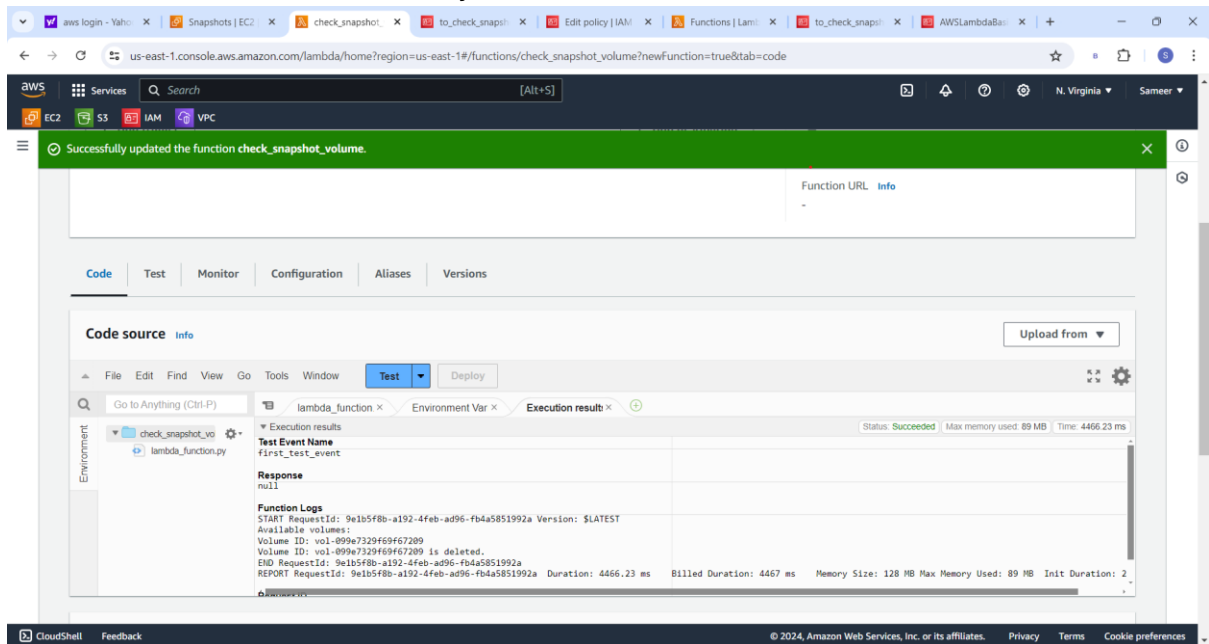
Modify the role to give permission to delete and describe resources.



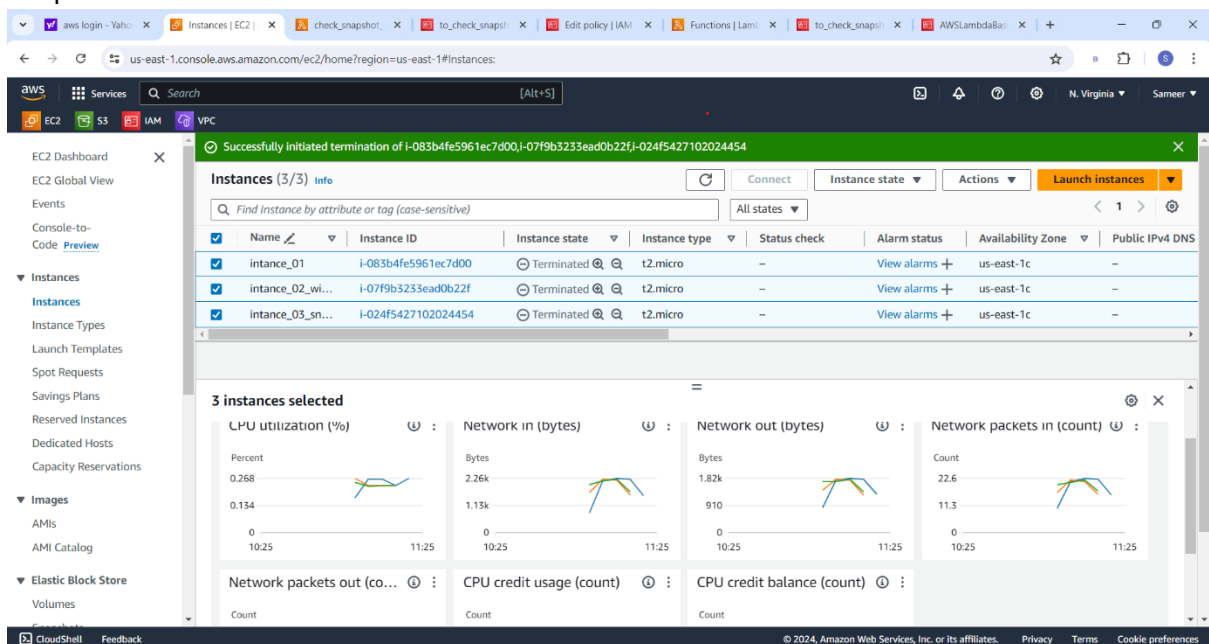
Configure a test event.



Test the event. Here, we see that only unused volumes are deleted.



But when we terminate all instances, then when we run the code, all unused volumes and snapshots are deleted.



here we can see that all unused snapshots and volumes are deleted successfully.

The screenshot shows the AWS Management Console for the 'us-east-1' region, specifically the 'Volumes' page. The left sidebar contains navigation links for various AWS services. The main content area shows a table of volumes. A single volume is listed with the following details: IOPS 100, Throughput -, Snapshot -, Created 2024/07/15 16:39 GMT+5:30, Availability Zone us-east-1c, Volume state Available, Alarm status No alarms, and Attached resources +. Below the table, there is a section for 'Fault tolerance for all volumes in this Region' and a 'Snapshot summary' showing '0 / 3' recently backed up volumes out of a total of 3 volumes. The summary also mentions the Data Lifecycle Manager default policy for EBS Snapshots status.

Then when we run code all unused volume and snapshot deleted

The screenshot shows the AWS Lambda console for the 'us-east-1' region, specifically the 'Execution results' page for a function named 'check_snapshot_volume'. The function was successfully executed, and the logs show that all unused volumes and snapshots were deleted. The logs include the function ID, request ID, and details about the deleted volumes and snapshots.

```
Function Logs
START RequestId: 90e43d43-492b-4d24-9cb7-5e00fc7fe642 Version: $LATEST
Available volumes:
Volume ID: vol-04afdc10e35295bc5
Volume ID: vol-04afdc10e35295bc5 is deleted.
Deleted EBS snapshot snap-0e7983df9aef1454d as its associated volume was not found.
Deleted EBS snapshot snap-05c3d97c3d9565ff6 as its associated volume was not found.
Deleted EBS snapshot snap-06444f58f40e21133f as its associated volume was not found.
END RequestId: 90e43d43-492b-4d24-9cb7-5e00fc7fe642
```

Here we can see that all unused snapshot and volumes deleted successfully

