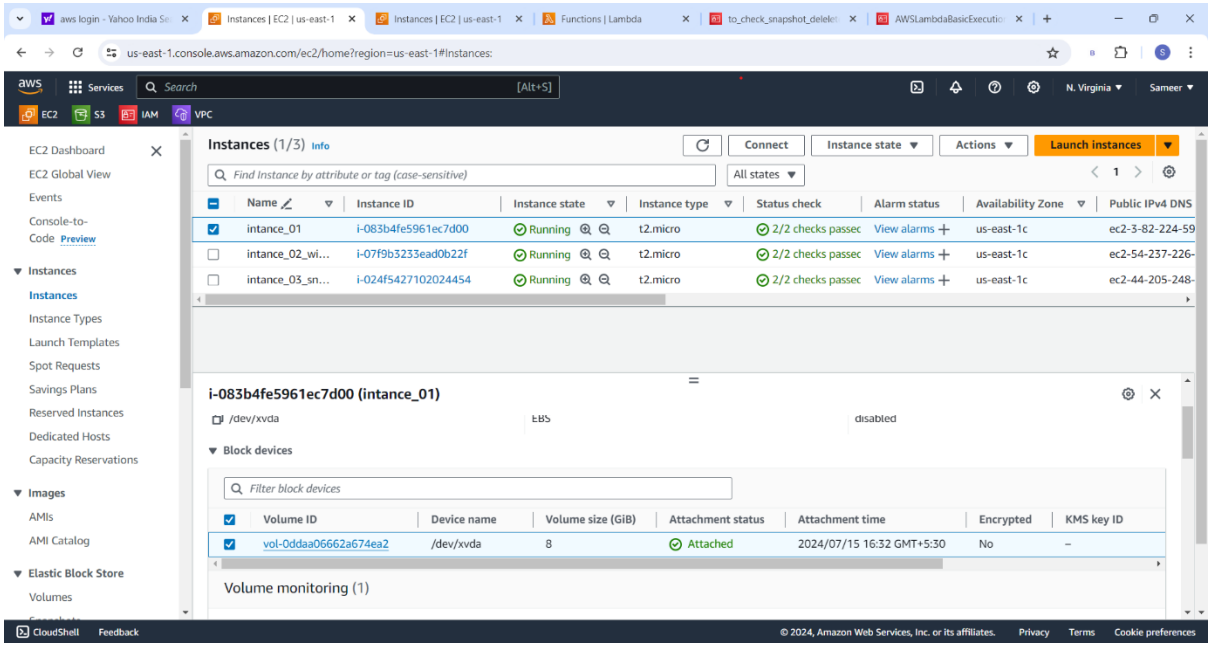
Cost Optimization

Python Script:

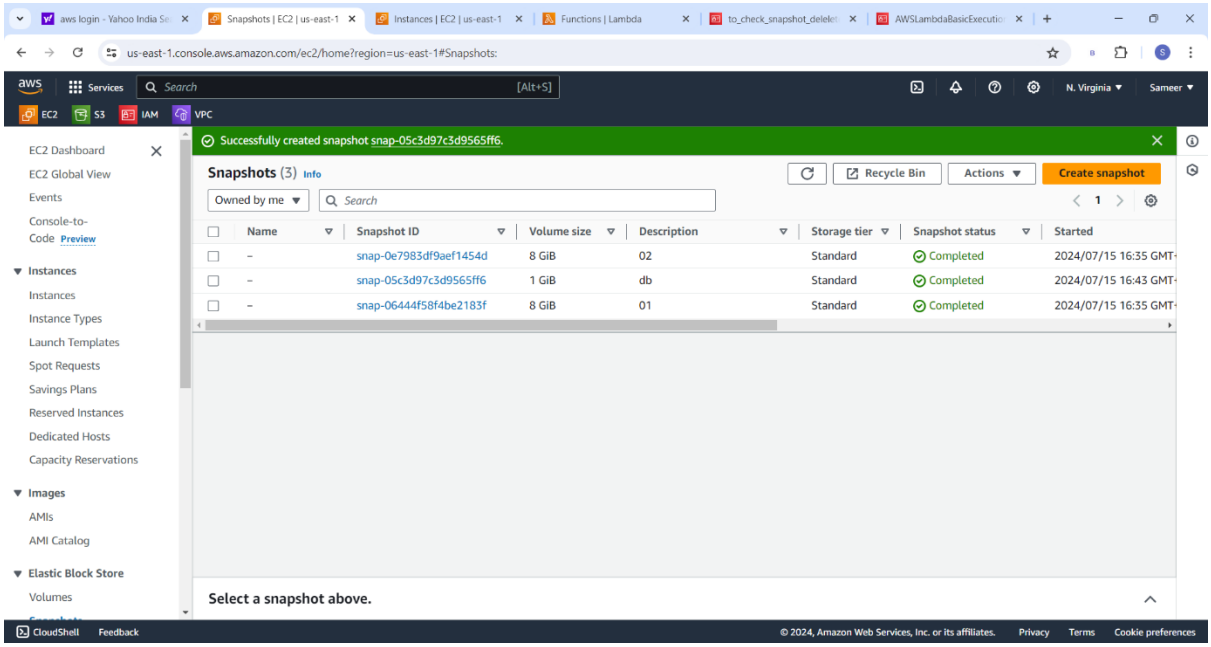
```
import boto3

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.")
```

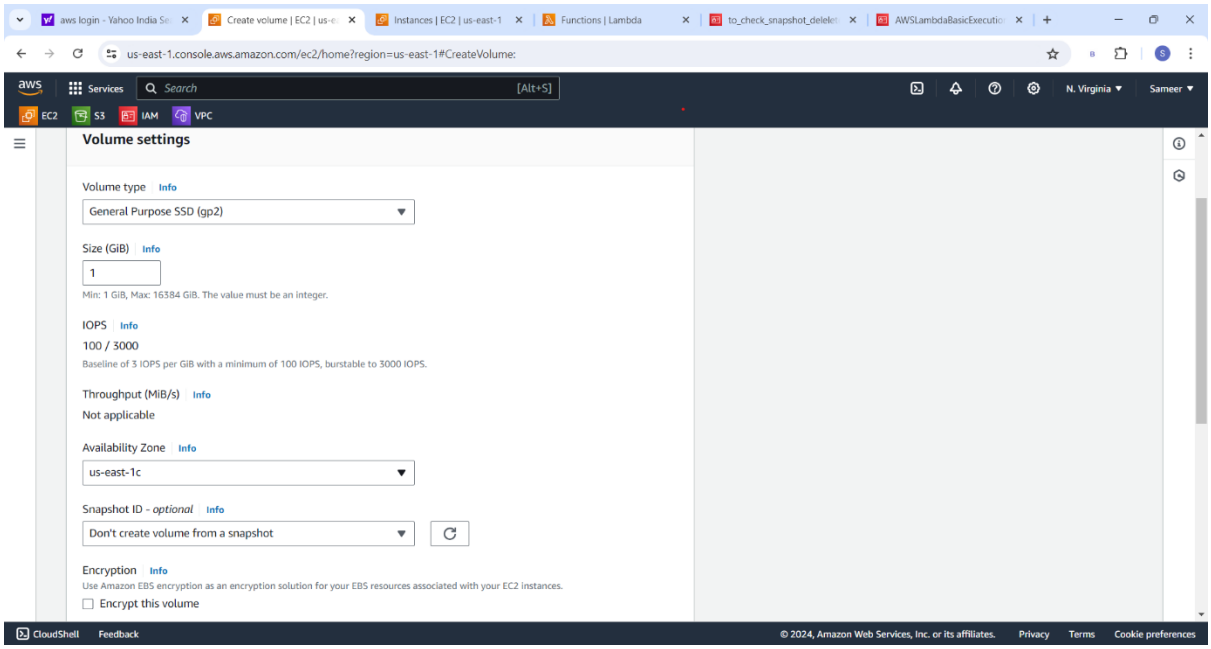
Create three instances (just for testing purposes).



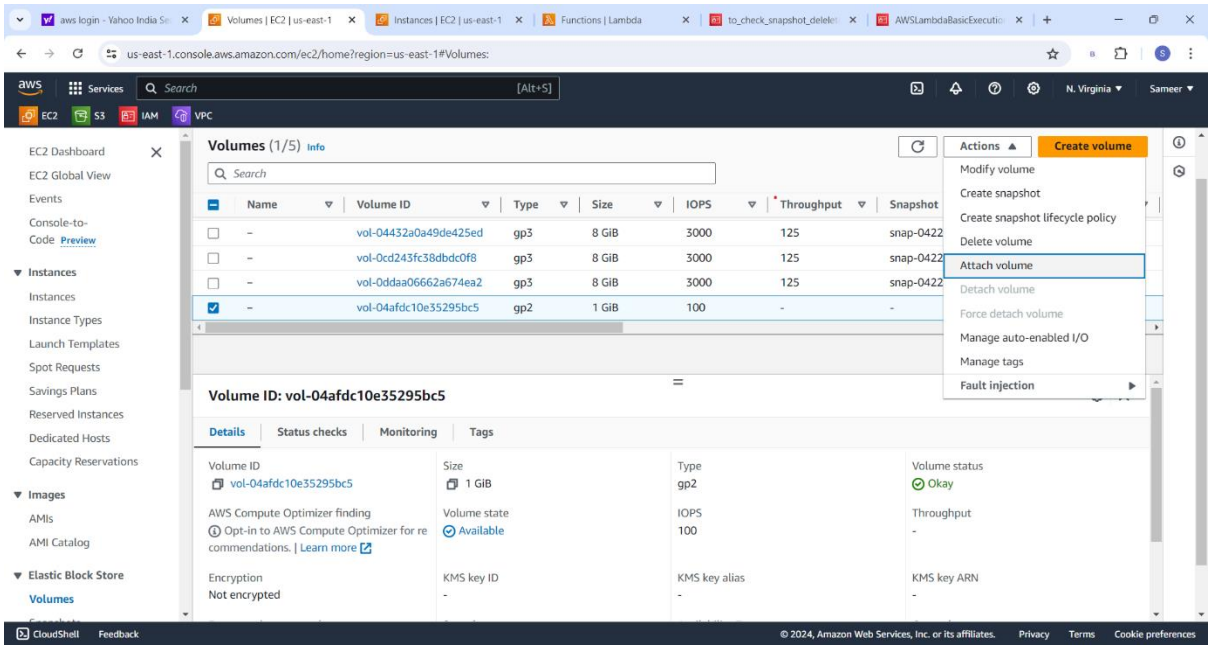
Create a snapshot of the existing volume.

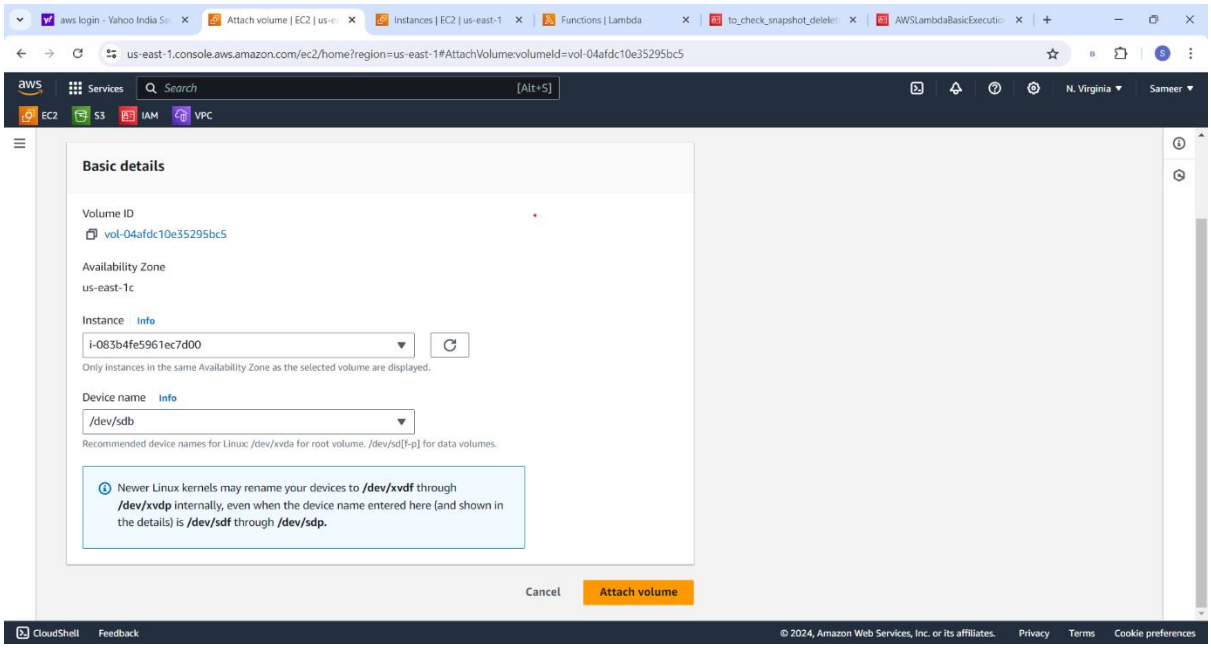


Create a volume.

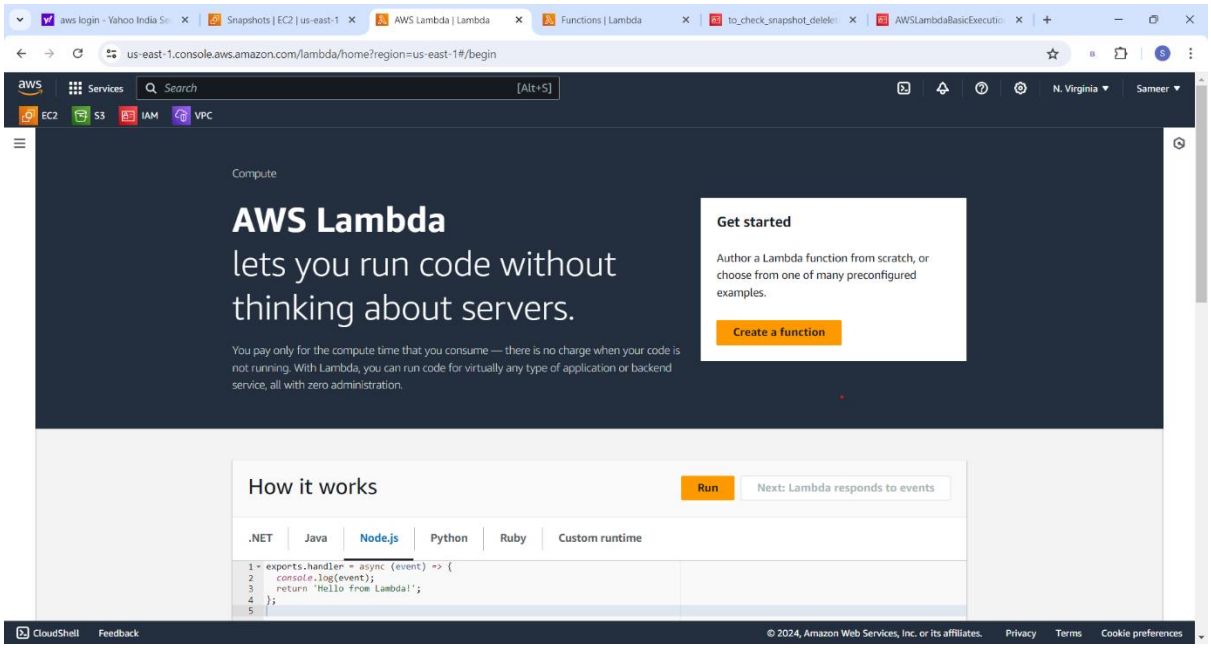


Attach the volume to the instance.

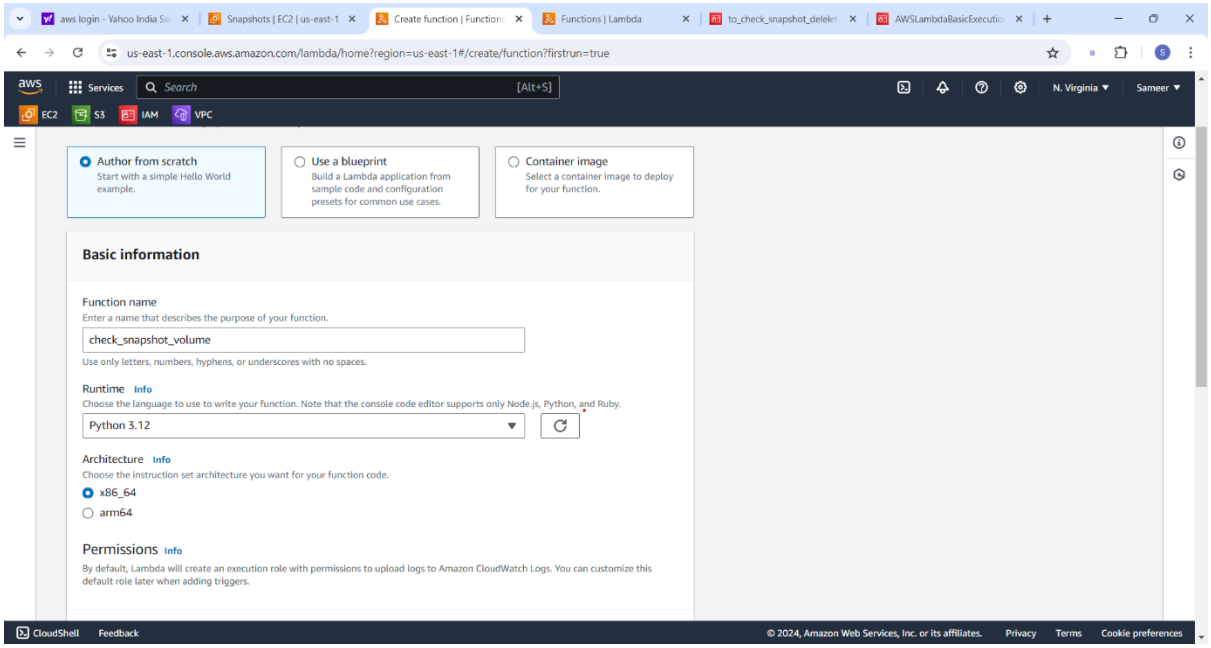




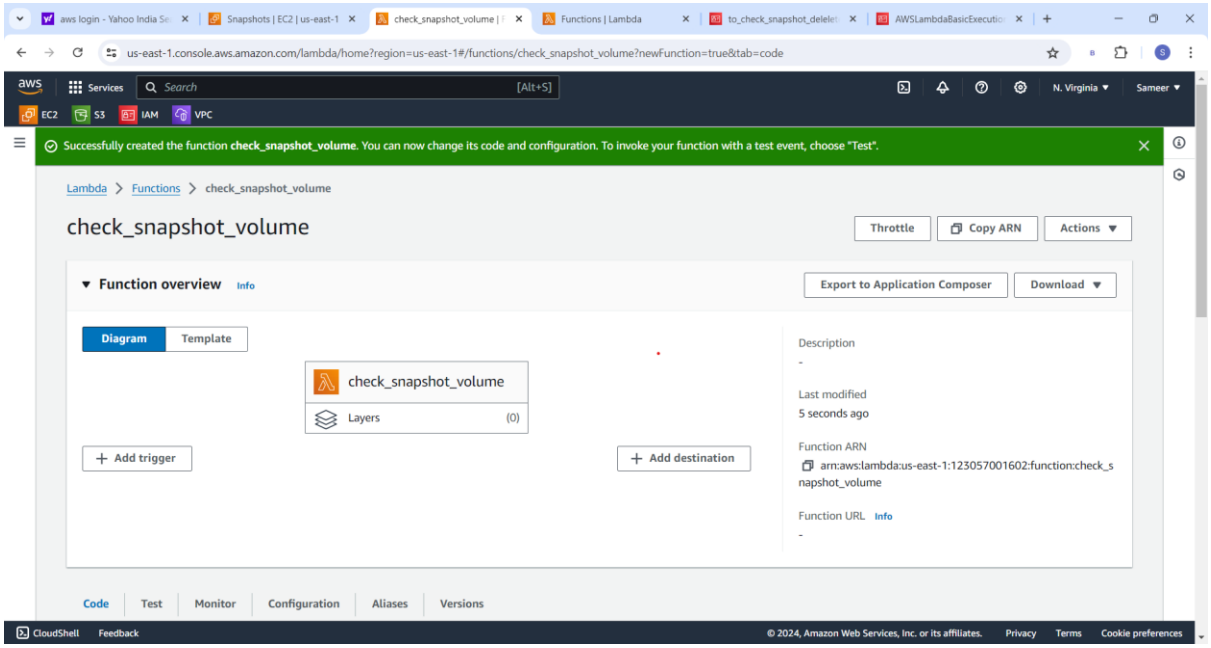
Create a Lambda function



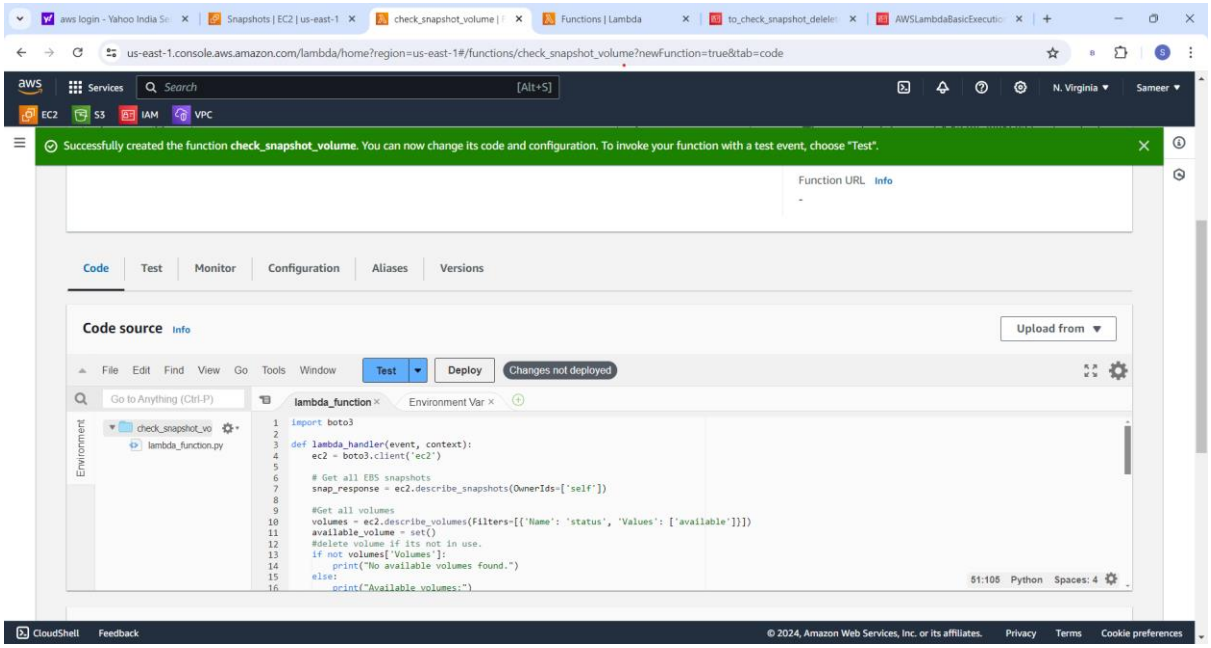
Choose Python as the programming language



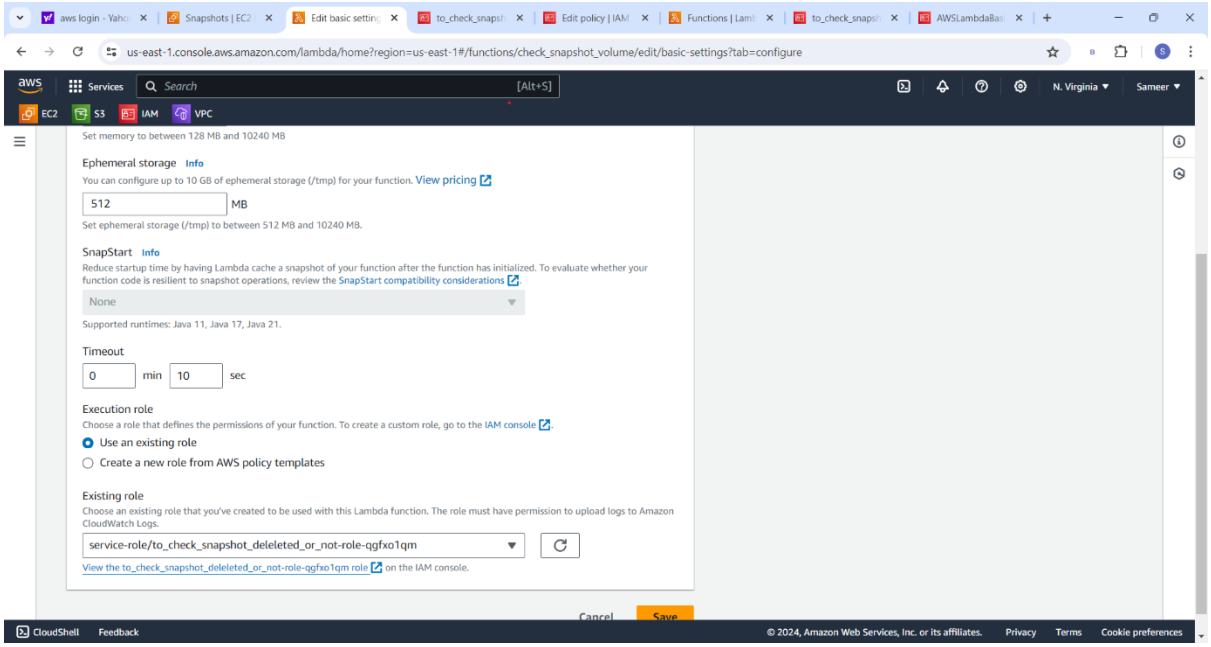
Here is the dashboard of my function.



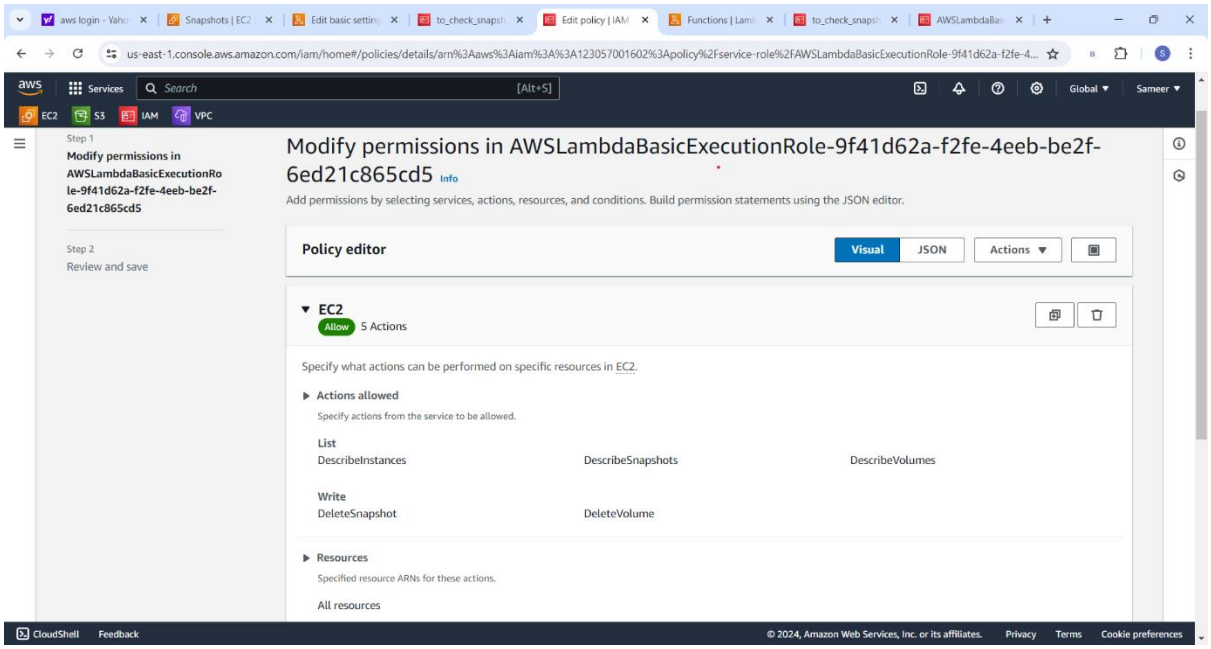
Scroll down, and in the code section, write a function to delete unused volumes and snapshots.



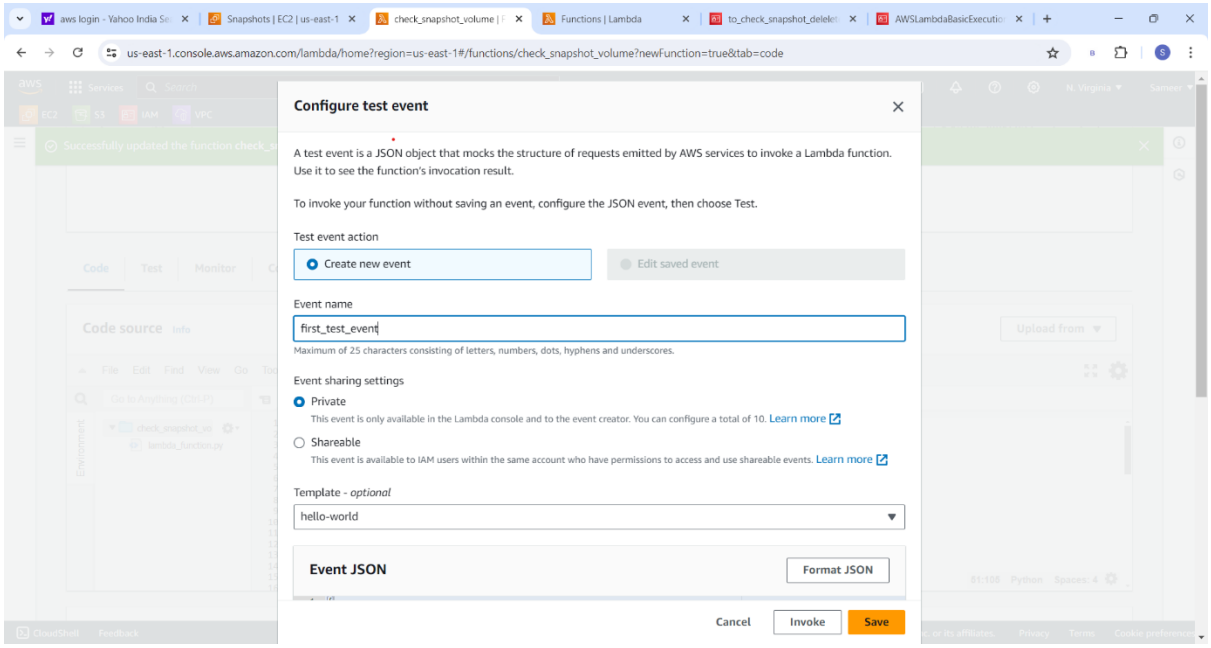
Increase the timeout to allow for longer execution time.



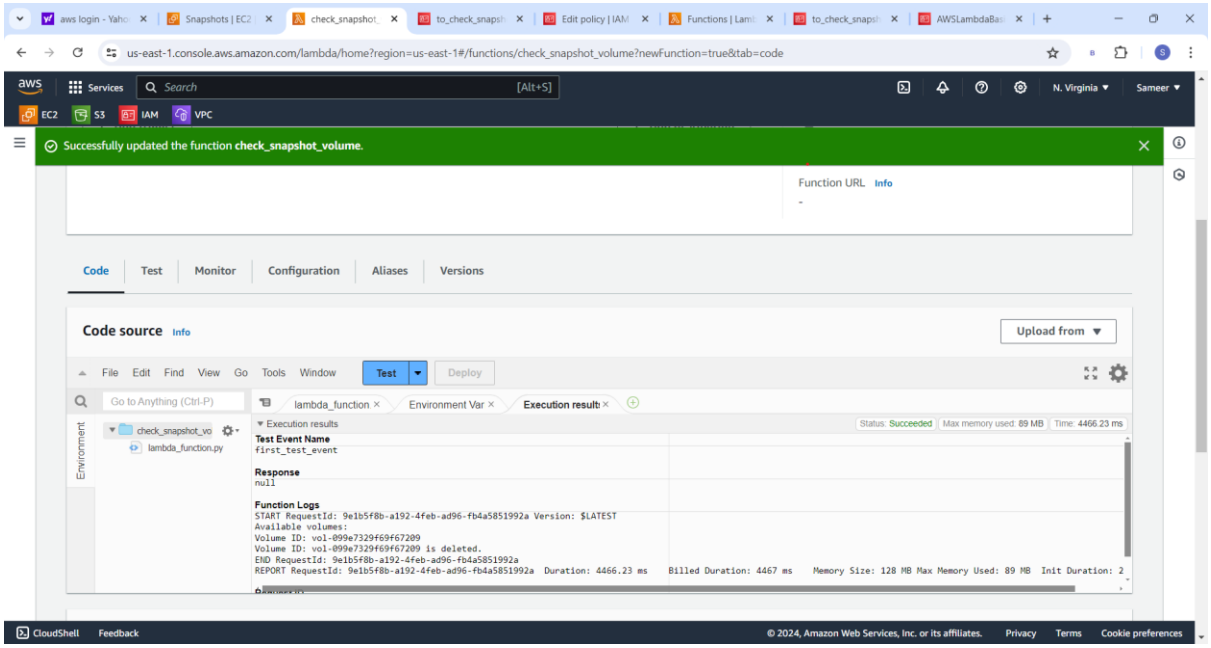
Modify the role to give permission to delete and describe resources.



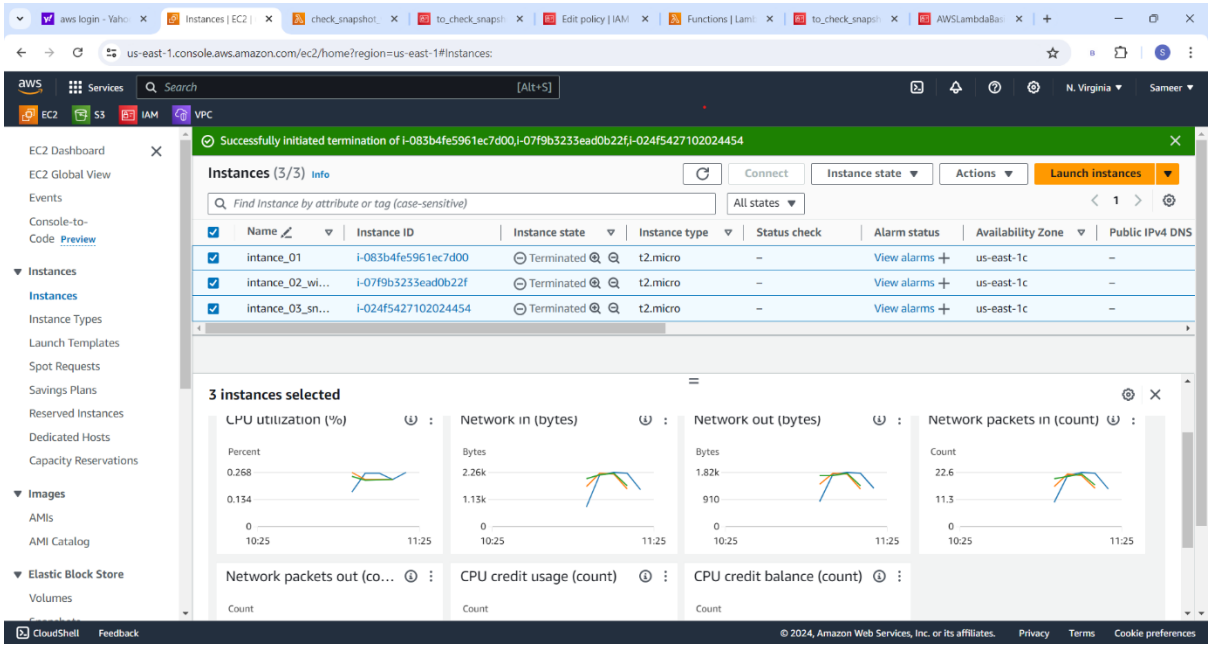
Configure a test event.



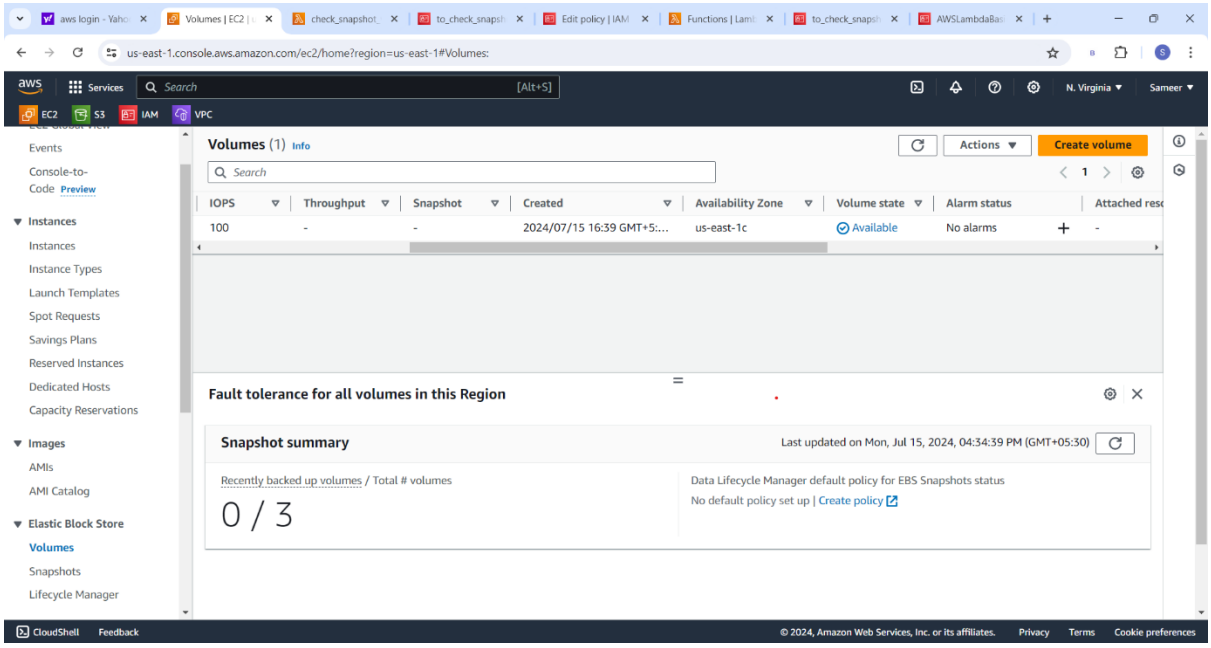
Test the event. Here, we see that only unused volumes are deleted.



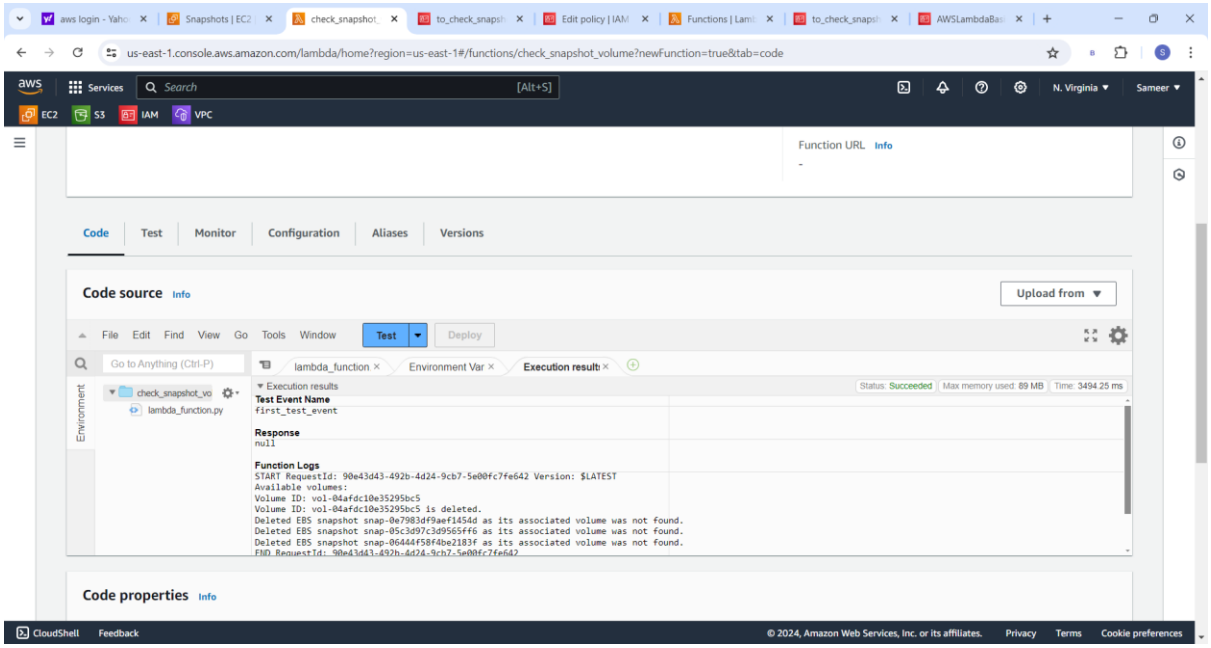
But when we terminate all instances, then when we run the code, all unused volumes and snapshots are deleted.



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



Then when we run code all unused voume and snapshot deleted



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