## 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**

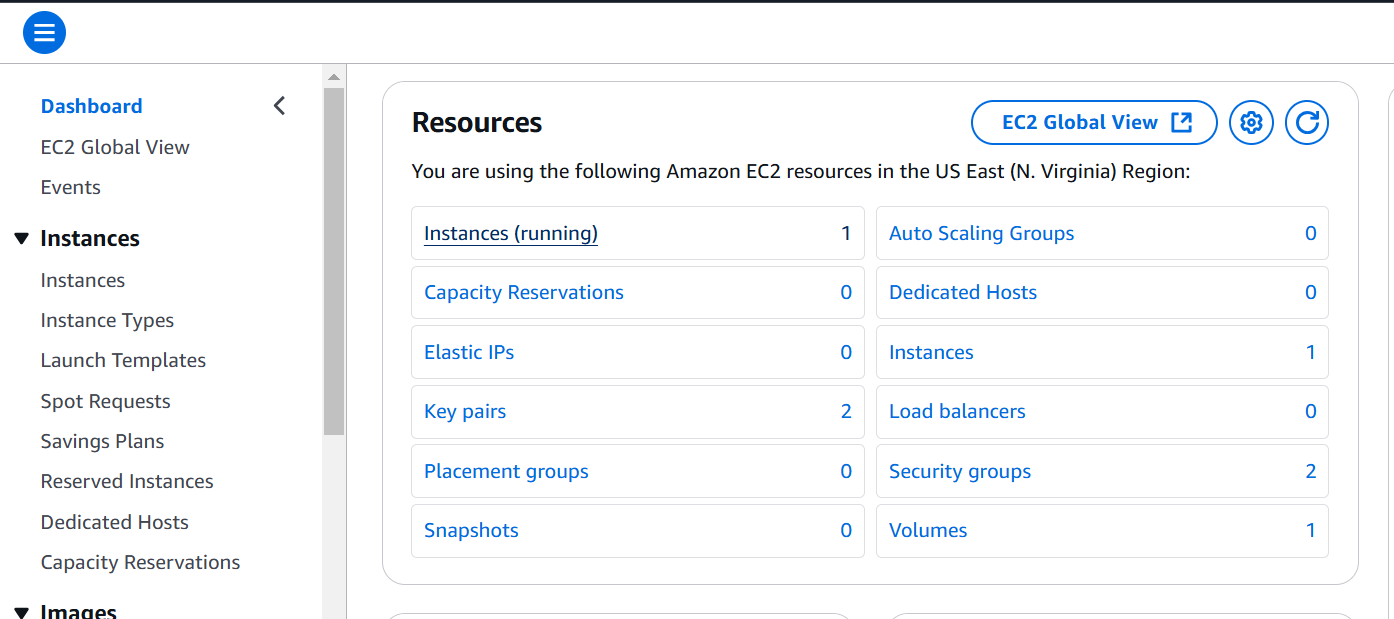
**Identifying Stale EBS Snapshots**

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:**

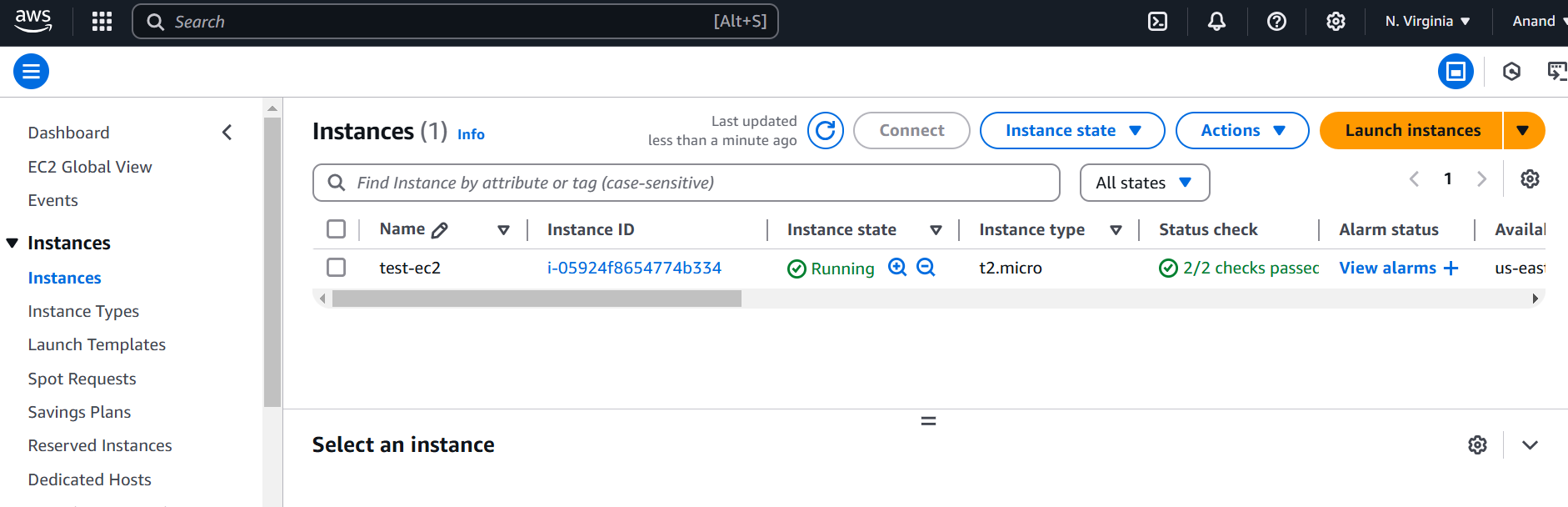
The Lambda function fetches all EBS snapshots owned by the same account ('self') and also retrieves a list of active EC2 instances (running and stopped). For each snapshot, it checks if the associated volume (if exists) is not associated with any active instance. If it finds a stale snapshot, it deletes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

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

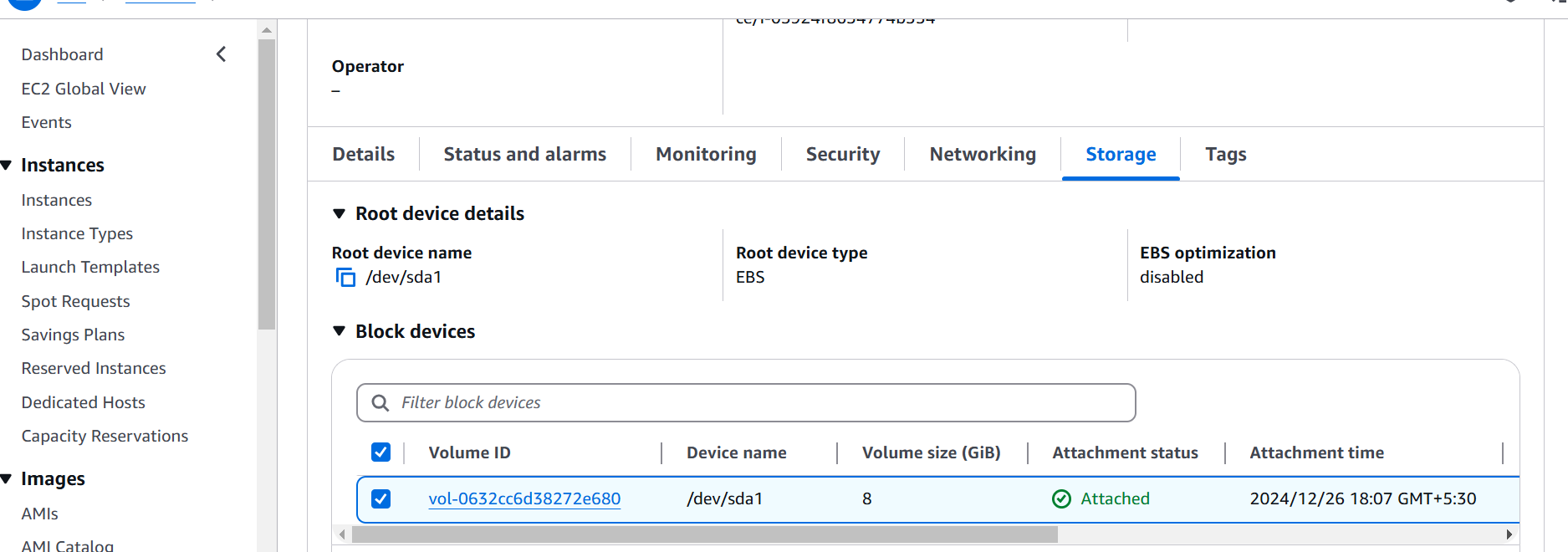


{Because I don’t have any Stale Resources in my AWS account for the purpose of this project demonstration, so I have created a Stale Resources i.e, from 2 - }

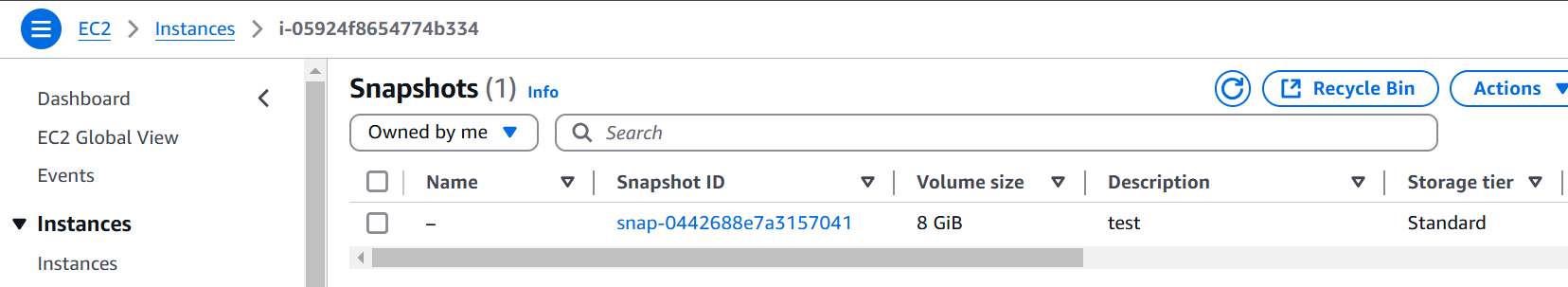
1. Created a EC2 instance by default it will add a volume and launch instance.



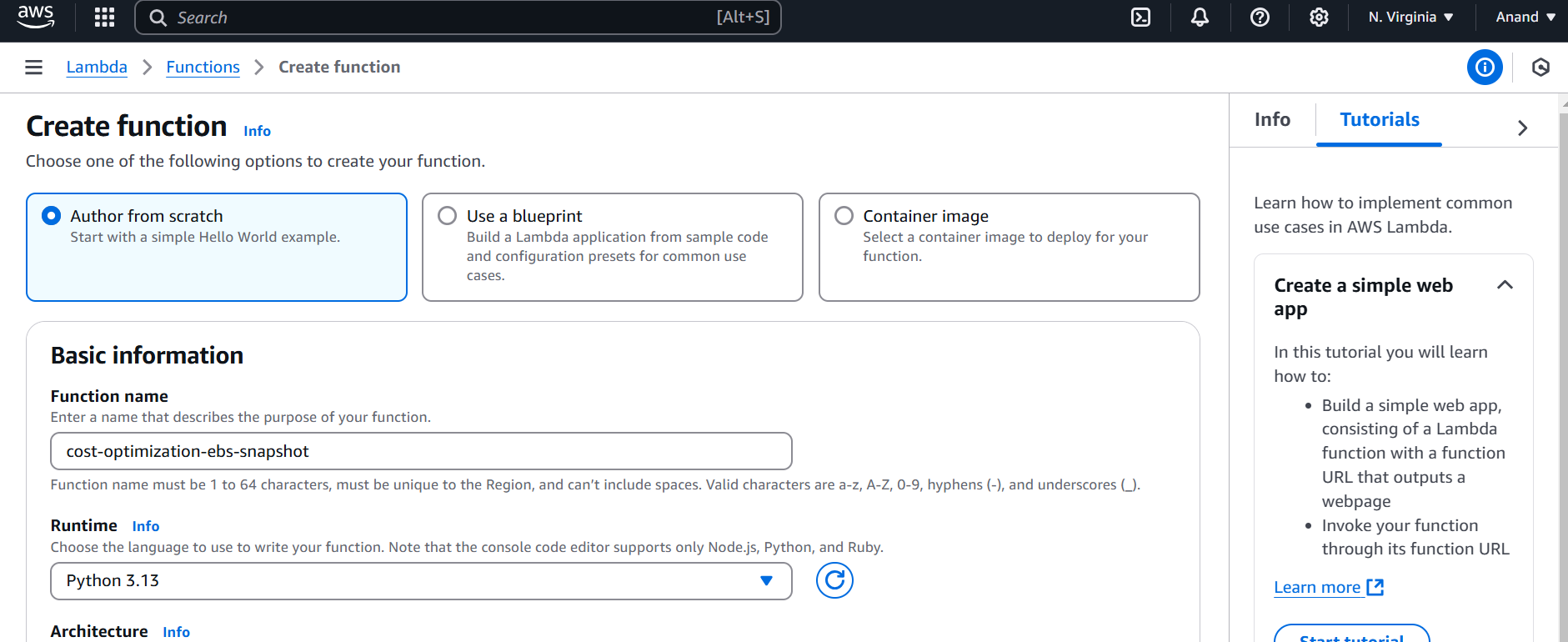
1. Click on instance id go to storage you find volume id of the created instance.



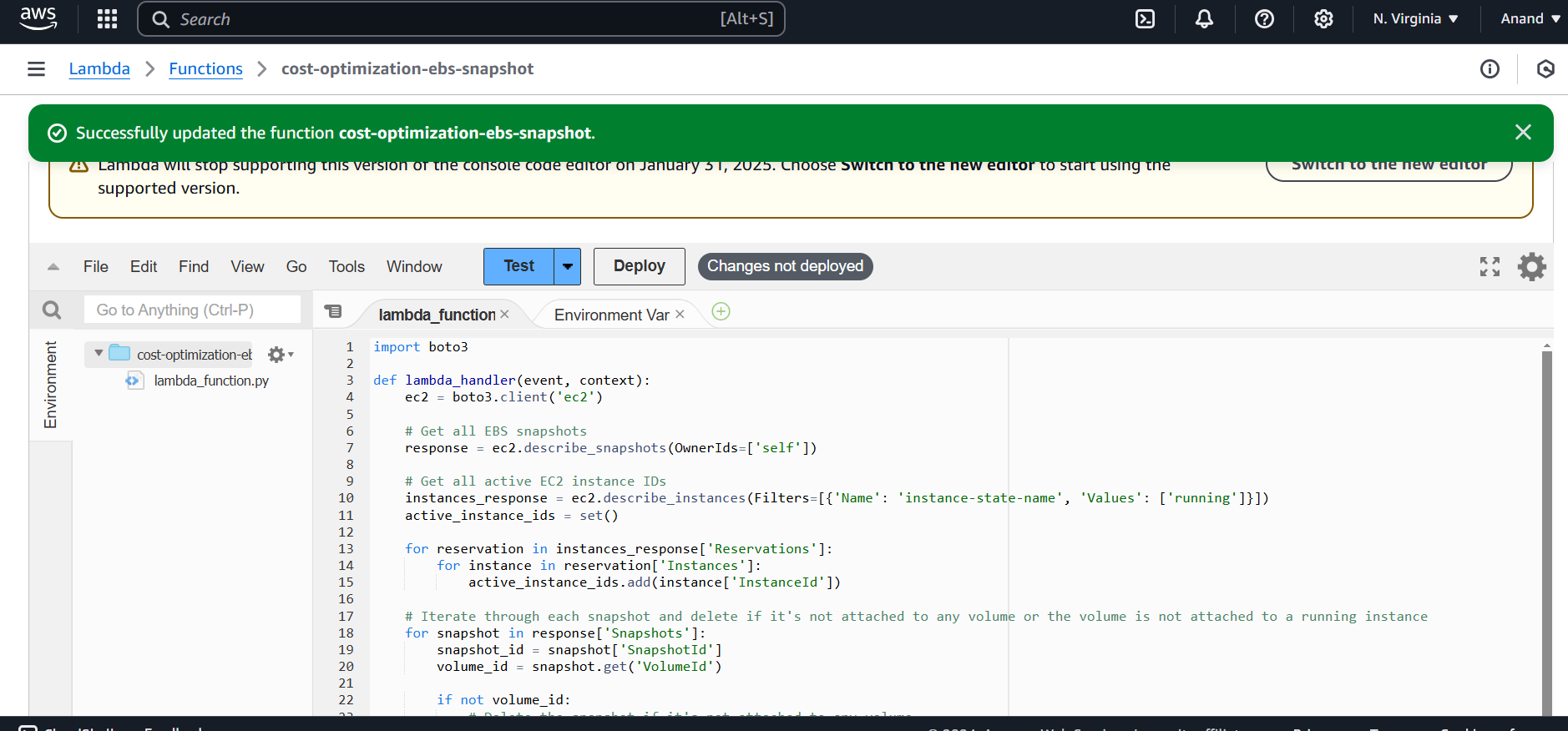
1. Go back to the EC2 dashboard and click on snapshots and create snapshot.



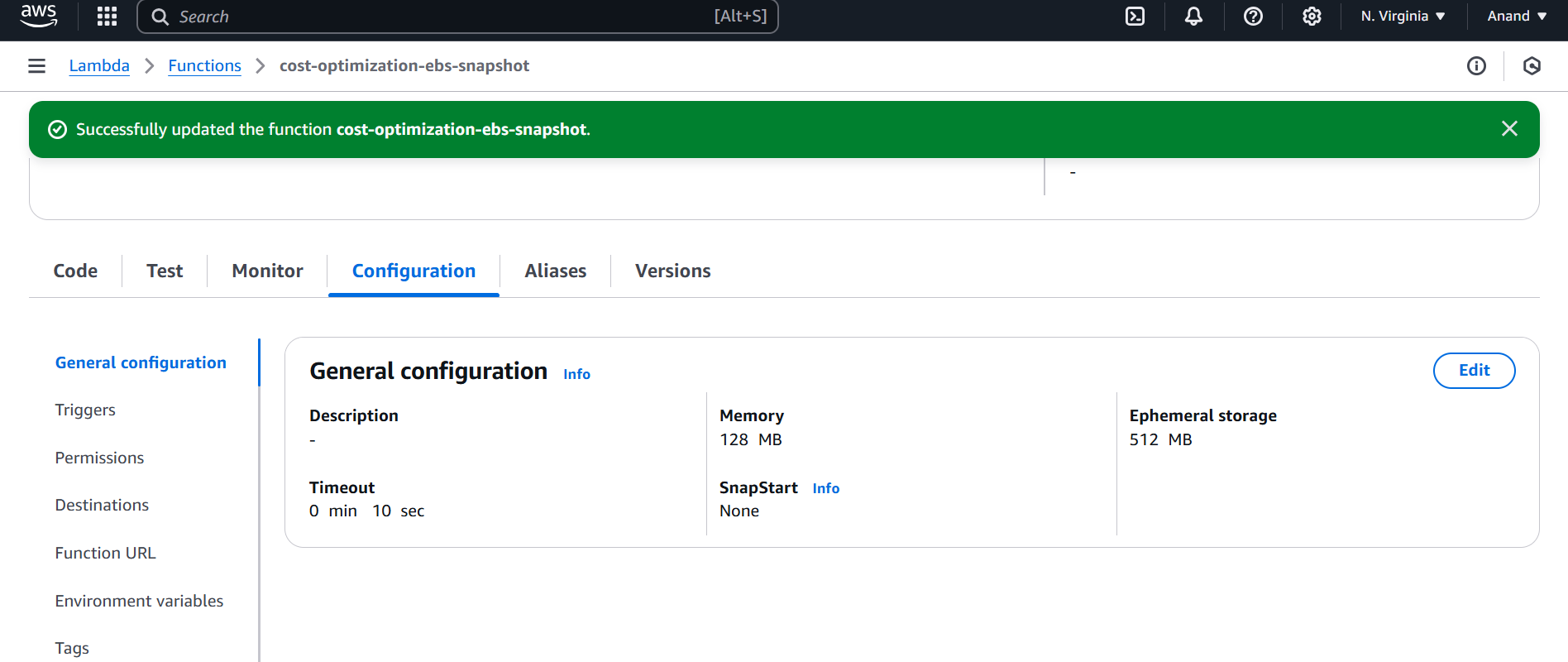
1. Go to lambda function and create a lambda function.



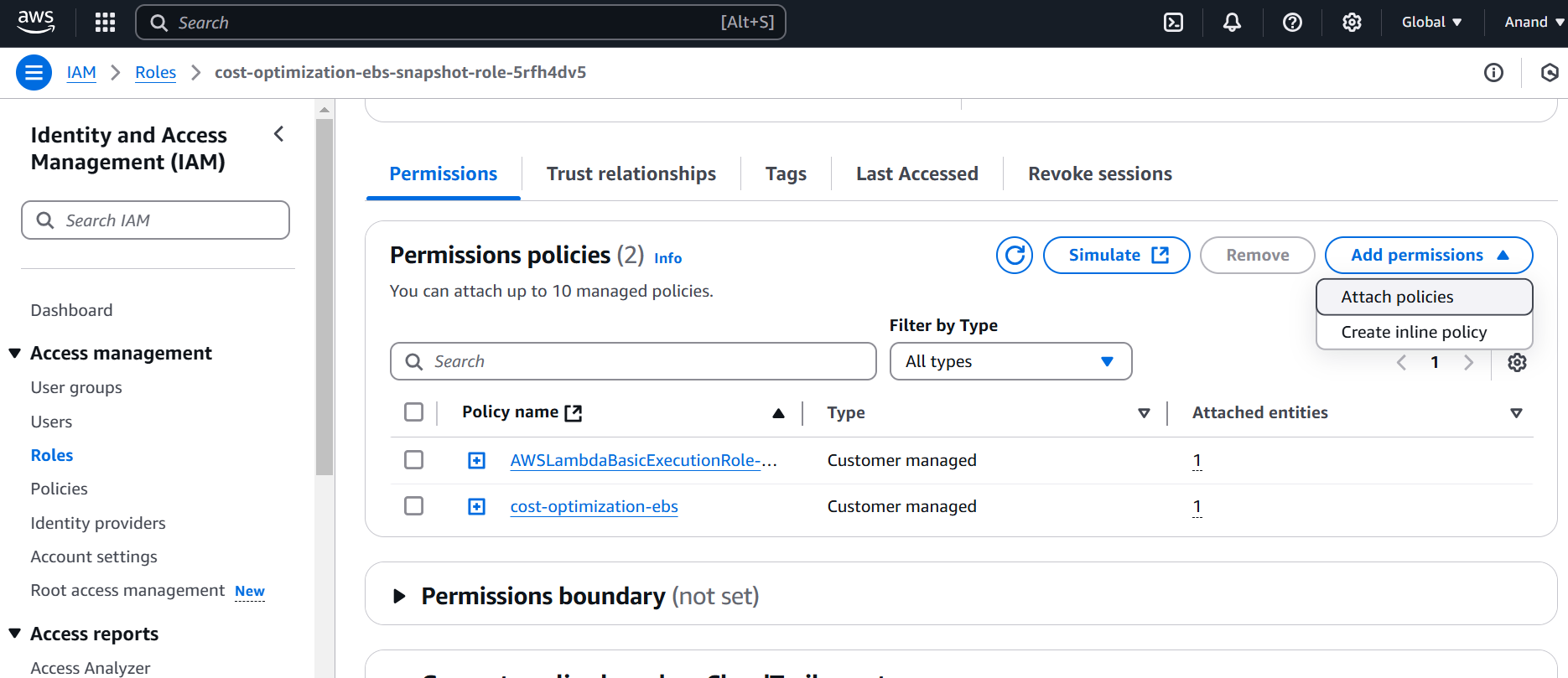
1. You can write your own lambda function code if you know, else go to this [gitHub](https://github.com/Anandbabu-patnam/AWS-Projects/blob/main/ebs_stale_snapshosts.py) .
2. Remove the existing code and past the code, save this, give a Test name and click on Deploy.



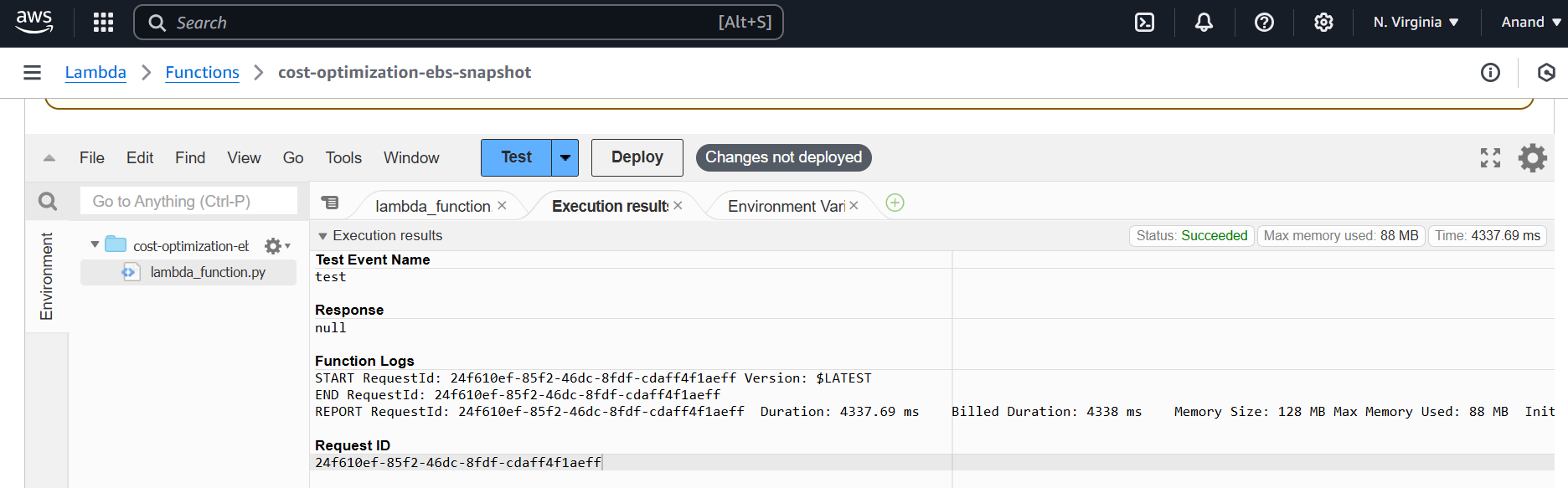
1. Now configure the timeout to 10sec



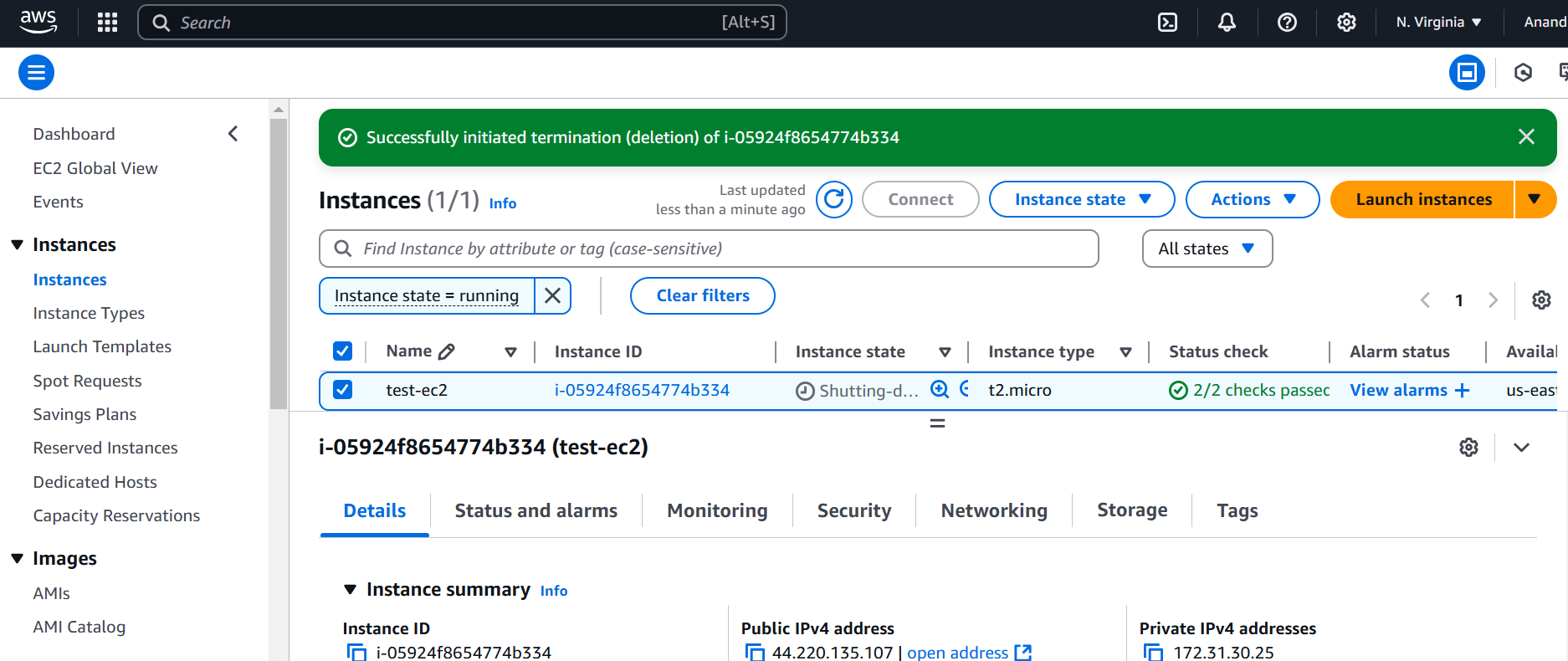
1. Go to permissions and click on Role name and add the permissions if you want you can create policies or add existing policies. Here I am creating my policies suitable for my project(snapshot permissions, EC2, Volume) and select resources all and name to the policy, create it.
2. Now attached the created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

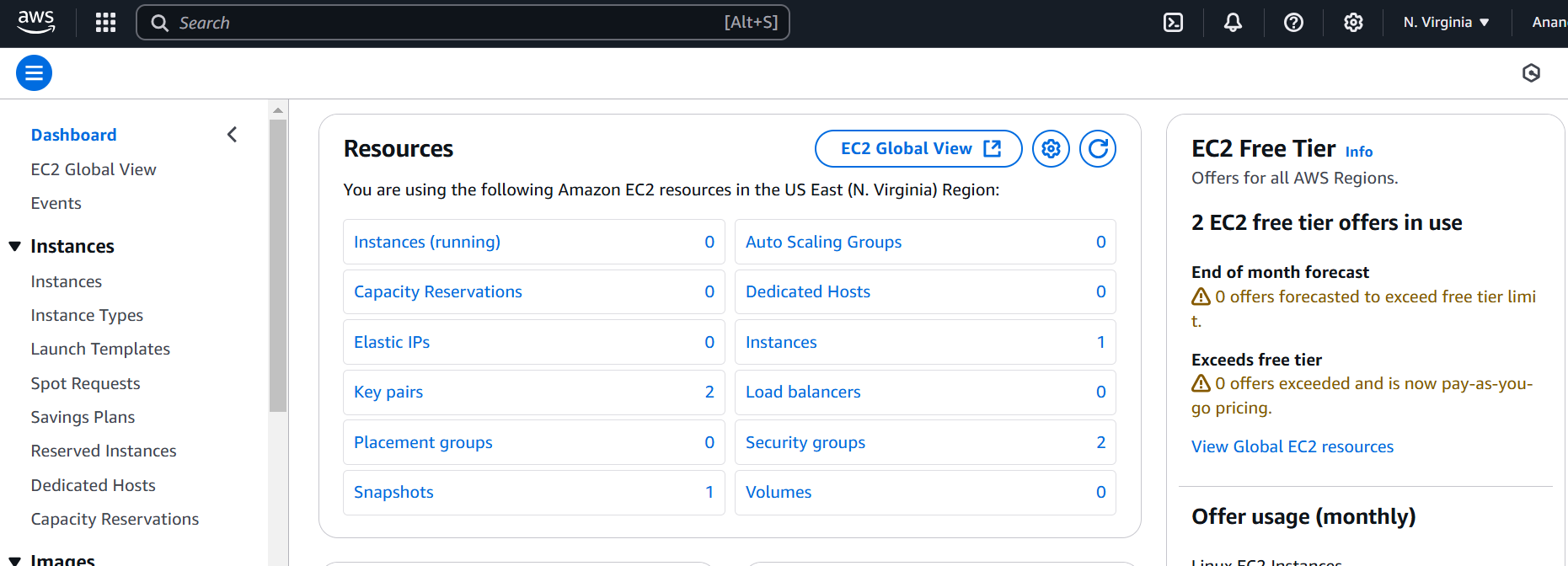


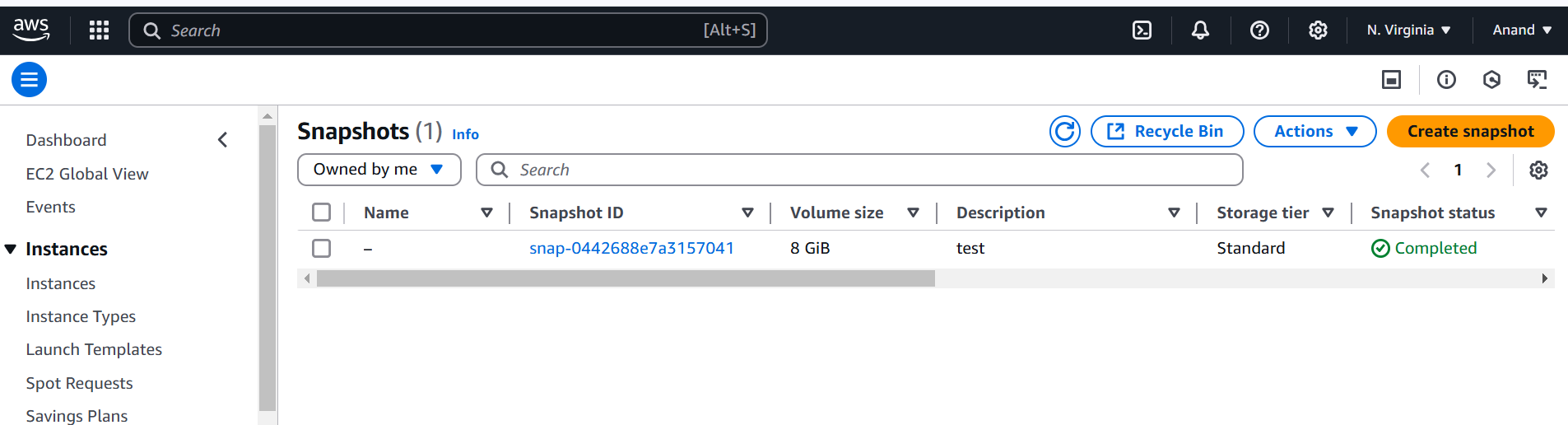
1. Click on the Test you will see like this below if you don’t have stale resources.



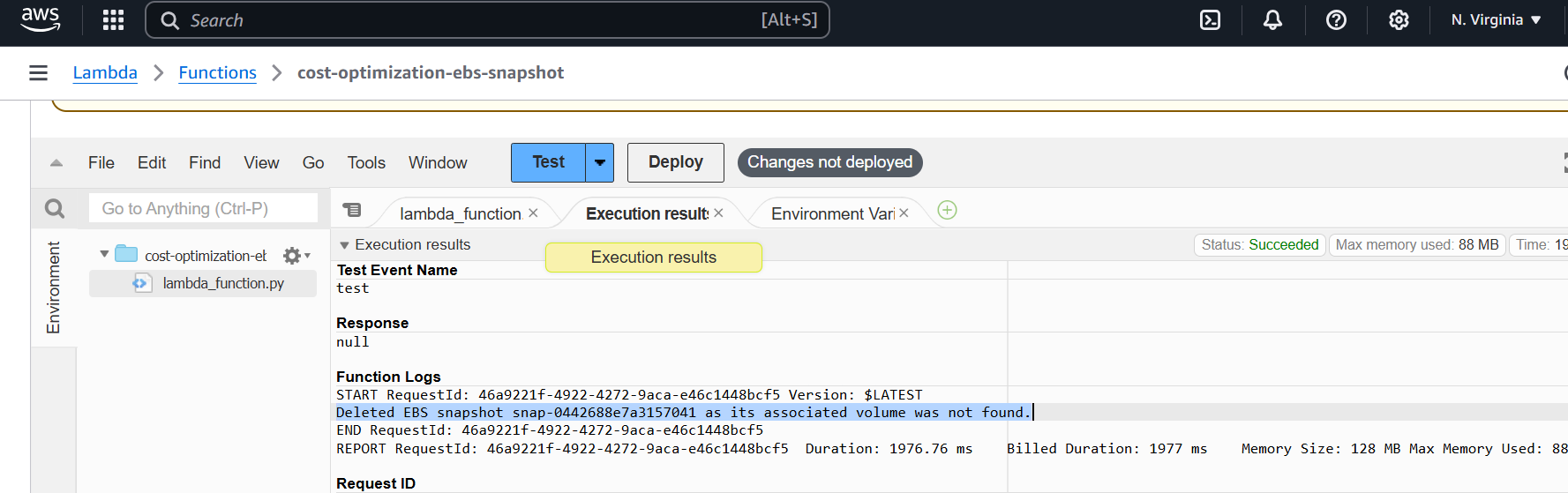
1. For project purpose I will delete the instance, but the snapshot is still present. Now I have stale resource.



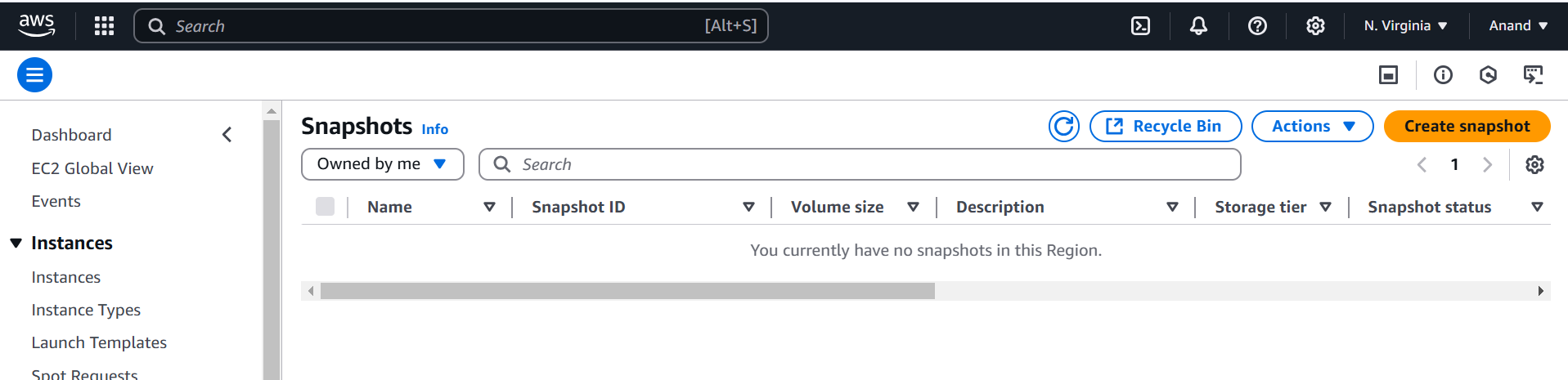




1. Now again Test the code, the stale resource is deleted.



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



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.