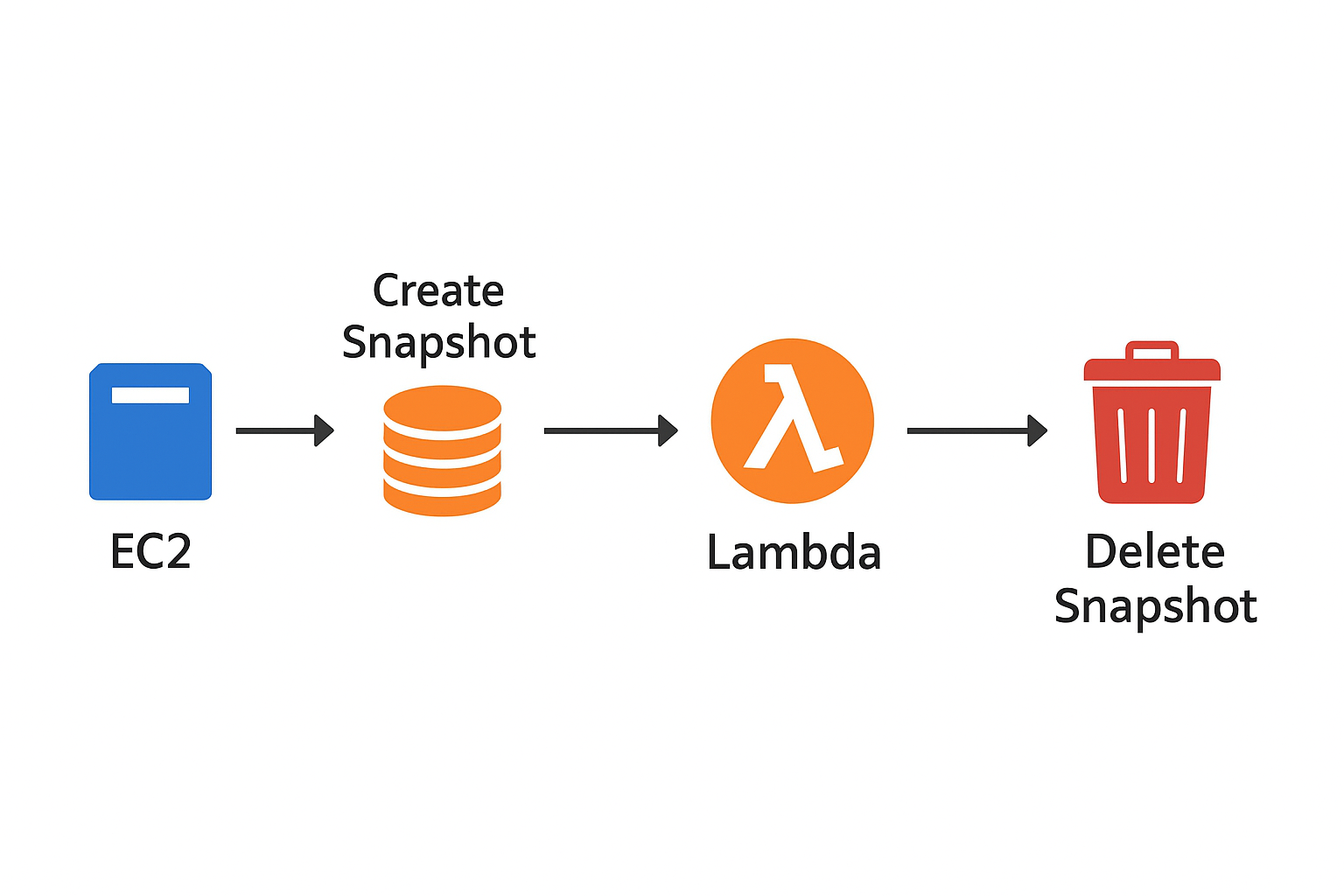
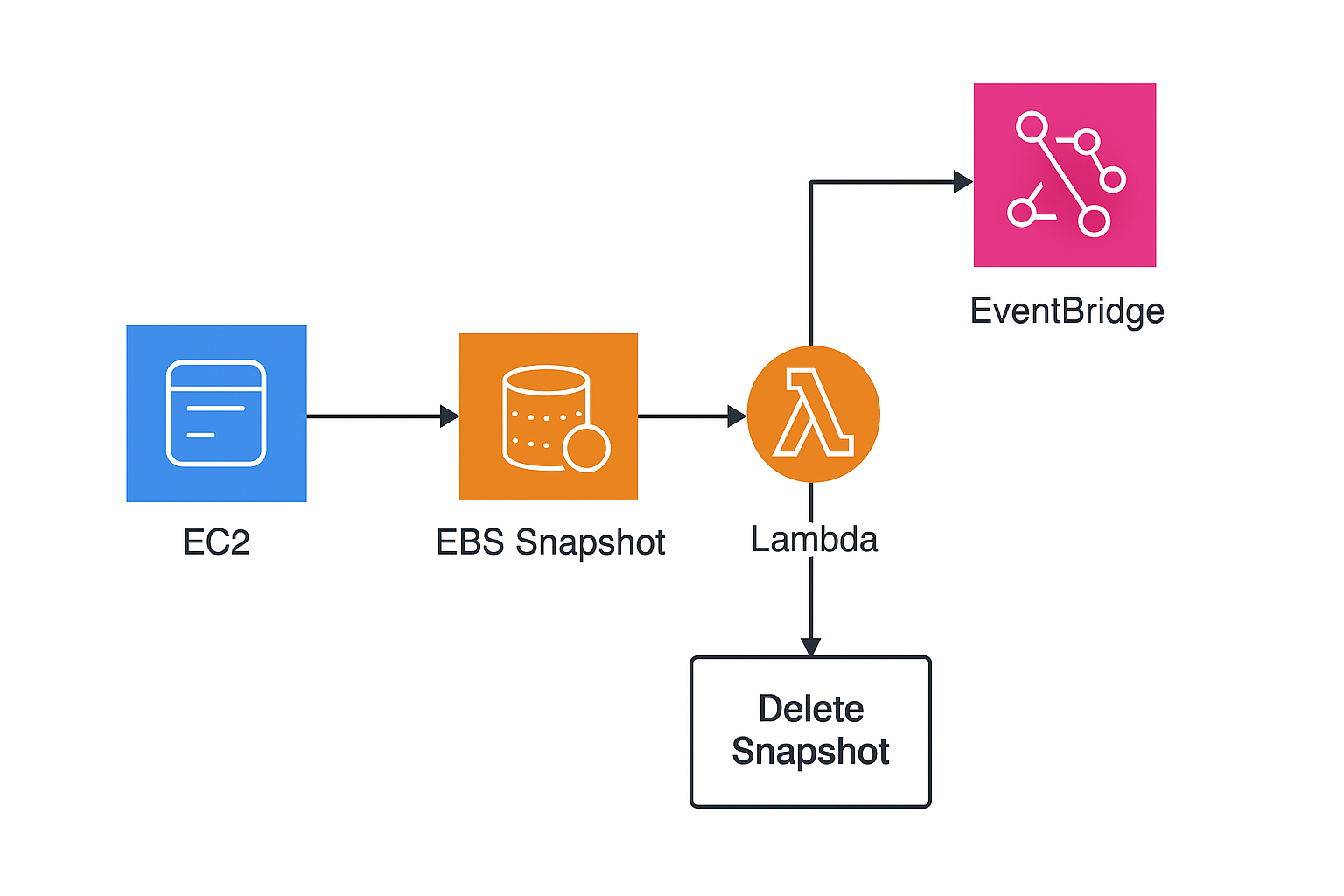
**TASK: Delete unused snapshots which is not associated with any EC2 , for optimize the cost.**

****

****

**EXECUTION:**

## **📝 Task: Delete unused EBS snapshots to optimize AWS cost**

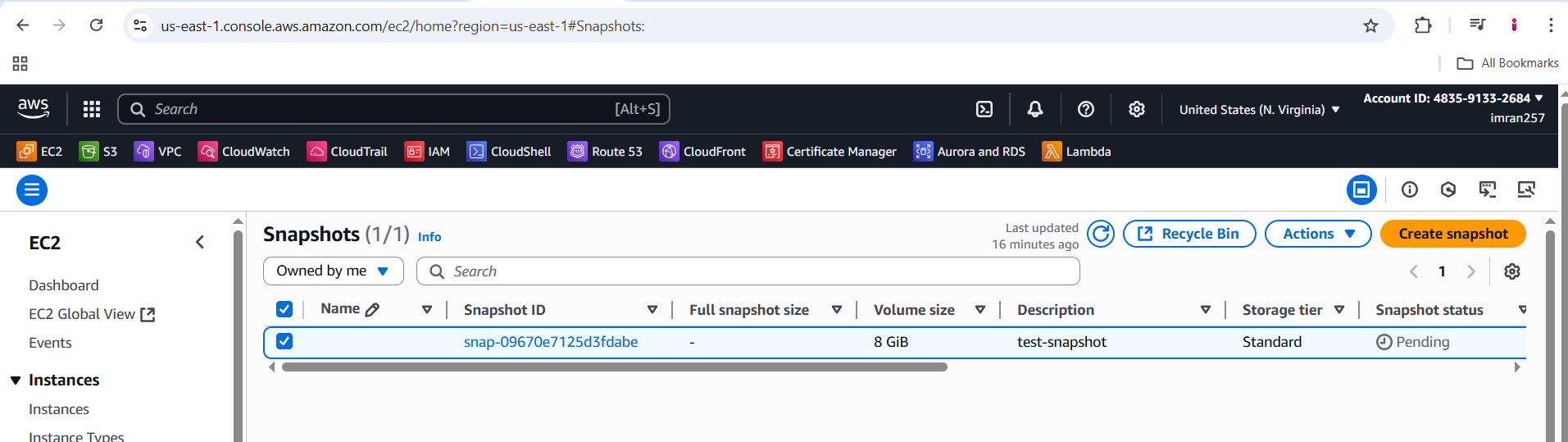
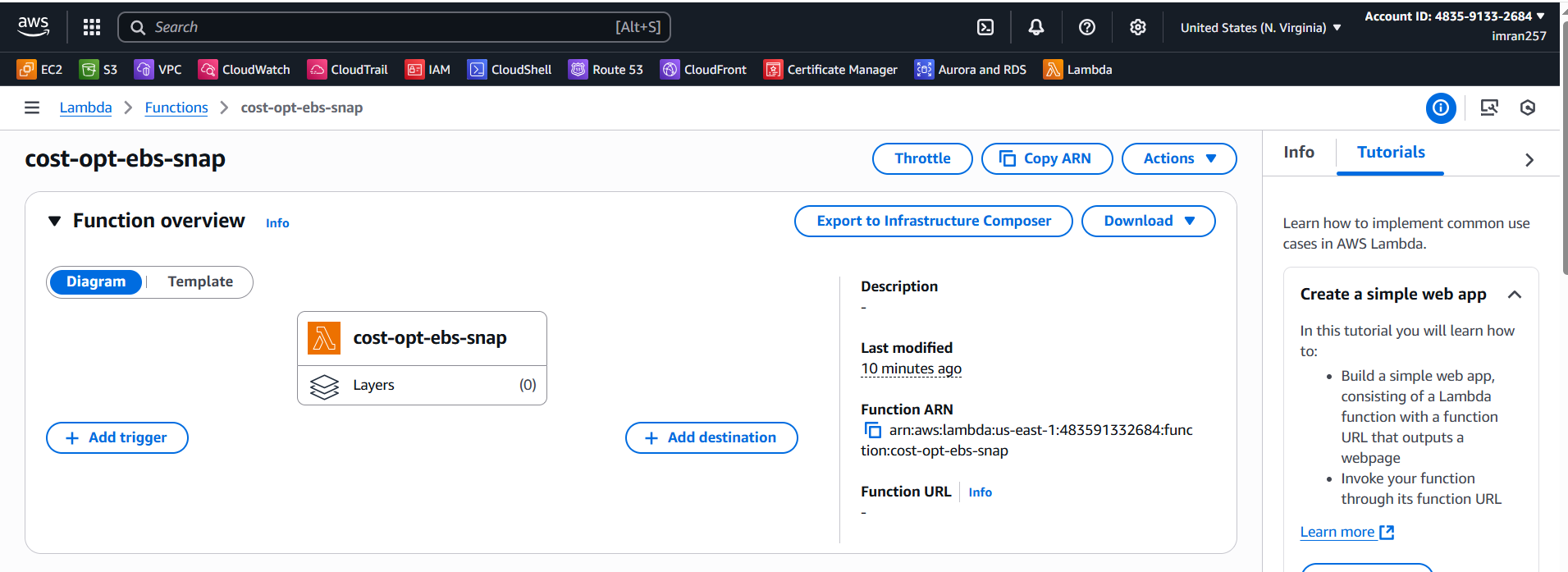
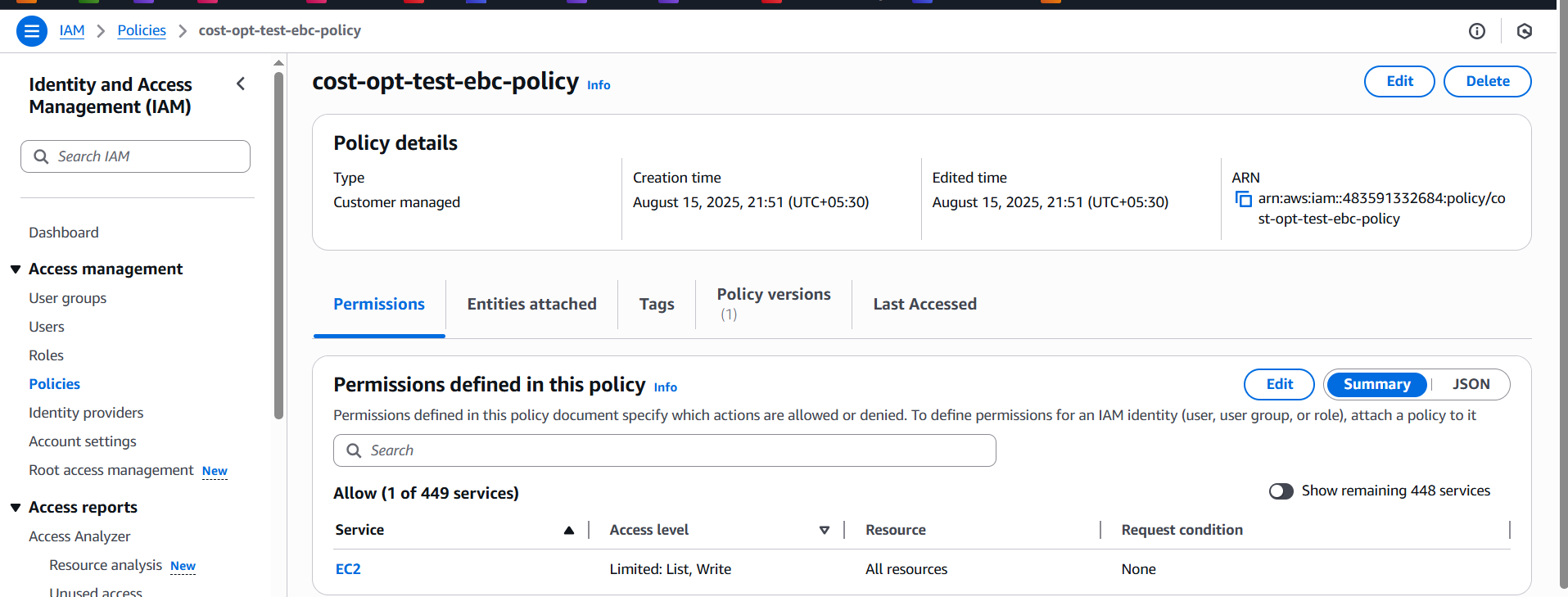
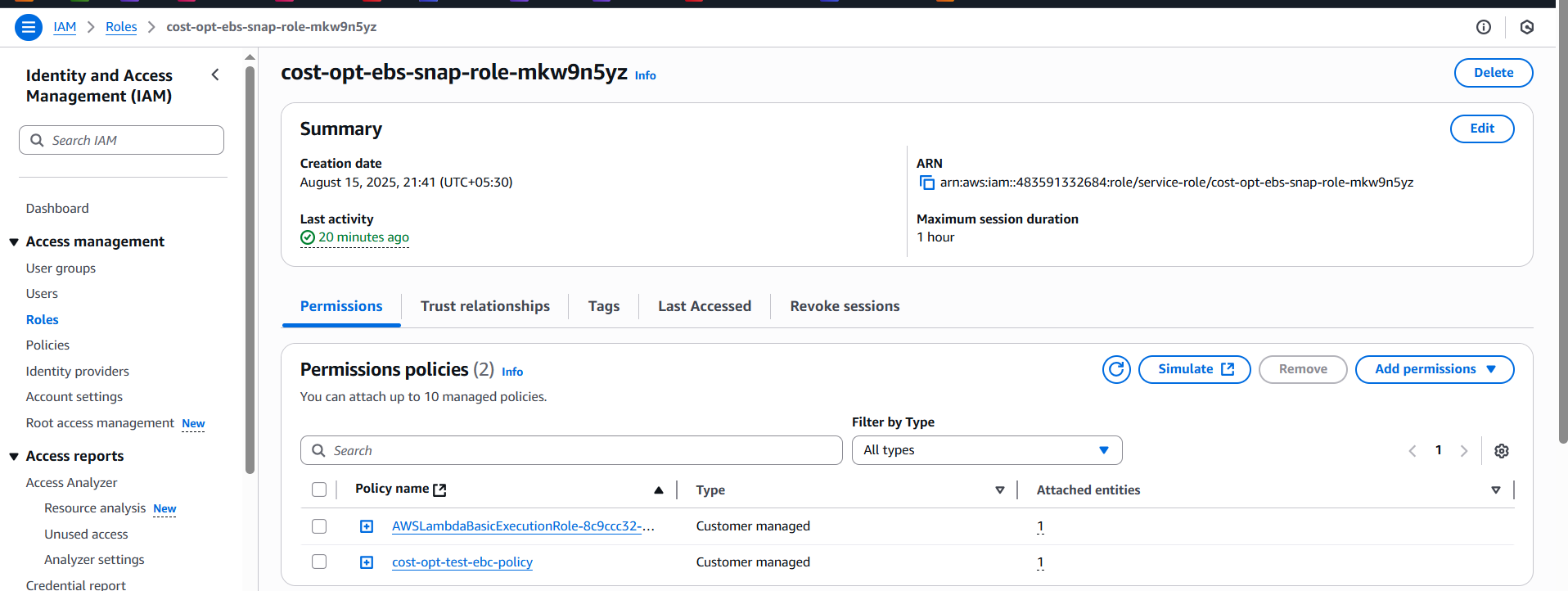
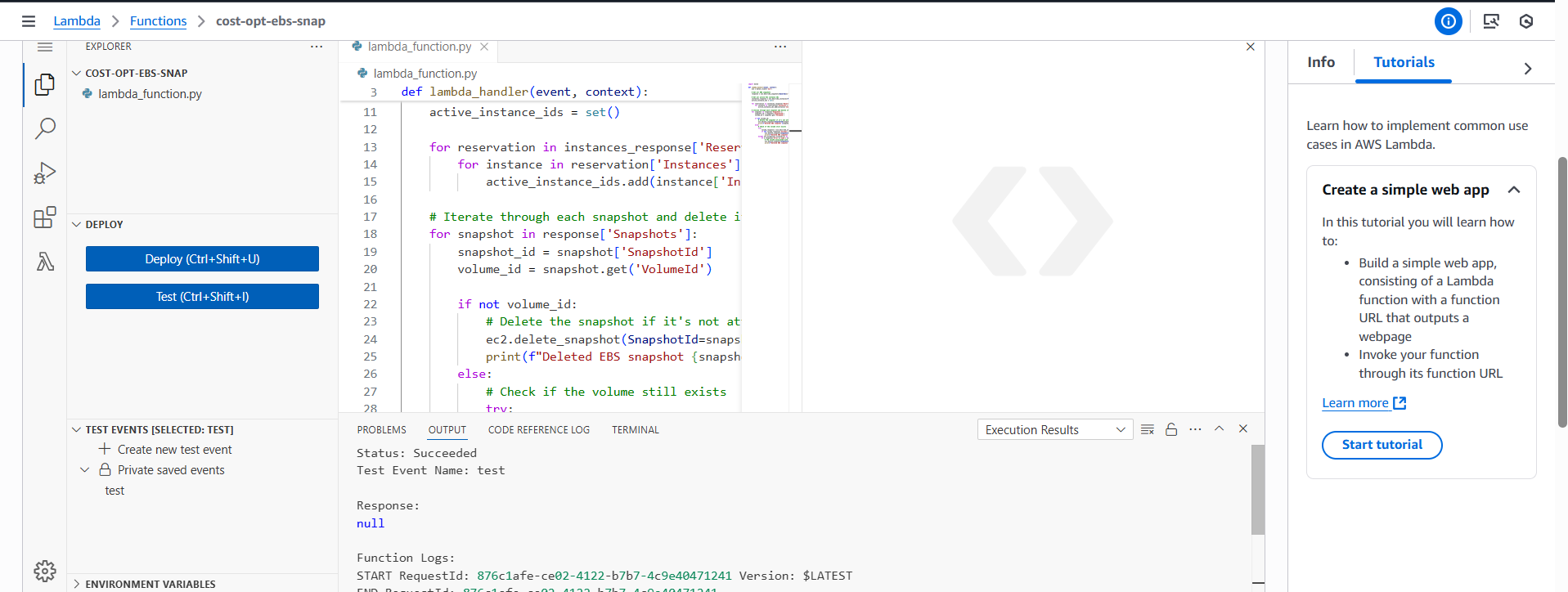
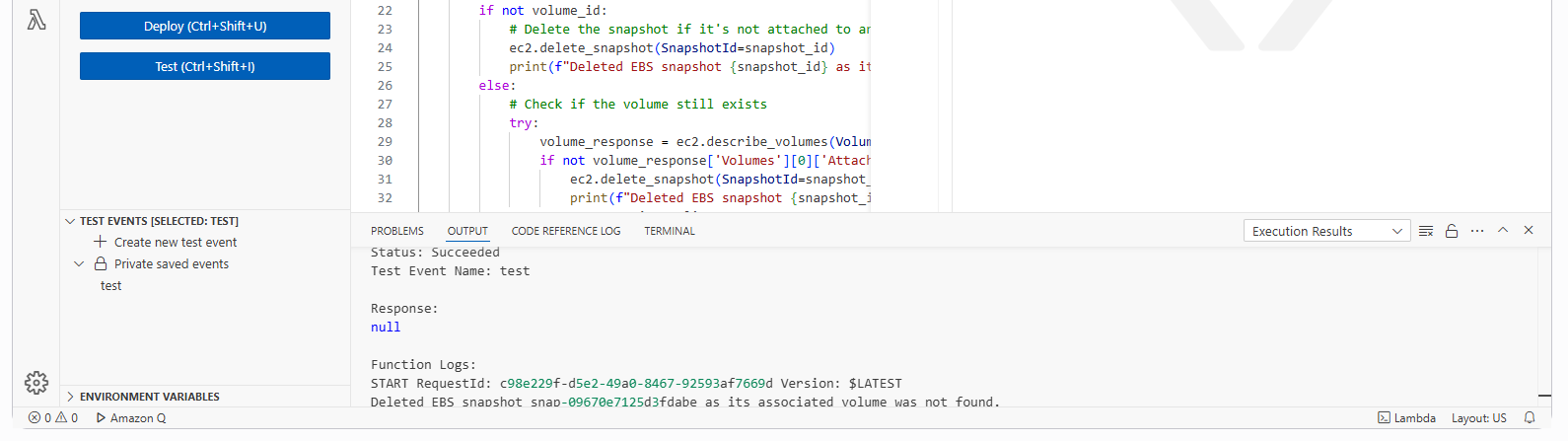
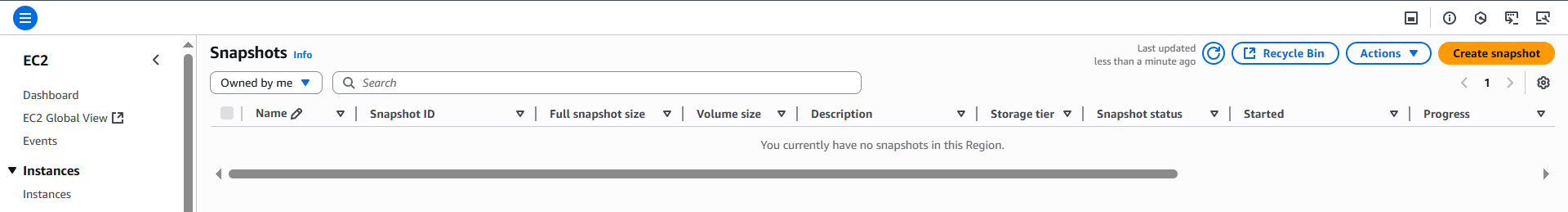
### Steps You Performed

1. Create an EC2 Instance  
   * You launched an EC2 instance.
   * This instance is needed to later create snapshots.
2. Take a Snapshot of EC2  
   * You created an EBS snapshot from the EC2 instance’s volume.
   * Snapshots are used for backup and recovery, but if left unused, they increase storage cost.
3. Create a Lambda Function  
   * You created a Lambda function named cost-opt-ebs-snap.
   * Purpose: Automatically detect and delete unused snapshots.
4. Write the Lambda Code (using boto3 – AWS SDK for Python)  
   * The code first lists all snapshots owned by your account.
   * It also lists all running EC2 instances and their associated volumes.
   * Logic:  
     + If a snapshot is not linked to any volume → delete it.
     + If the snapshot is linked to a volume, but the volume is not attached to a running instance → delete it.
     + If the volume itself is already deleted → delete the snapshot.
5. Create an IAM Policy for Lambda  
   * You created a custom IAM policy with permissions:  
     + ec2:DeleteSnapshot (to delete snapshots)
     + ec2:DescribeSnapshots, ec2:DescribeVolumes, ec2:DescribeInstances (to check which snapshots/volumes/instances exist).
6. Attach the Policy to Lambda Execution Role  
   * In the Lambda function’s configuration, you attached the IAM role with the above policy.
   * This allowed your Lambda function to actually perform snapshot deletions.
7. Test the Lambda Function  
   * First test: Snapshots were not deleted because they were still associated with the EC2 instance.
   * Then you terminated the EC2 instance → now snapshots were unused.
   * Second test: Lambda executed → unused snapshot was deleted successfully.

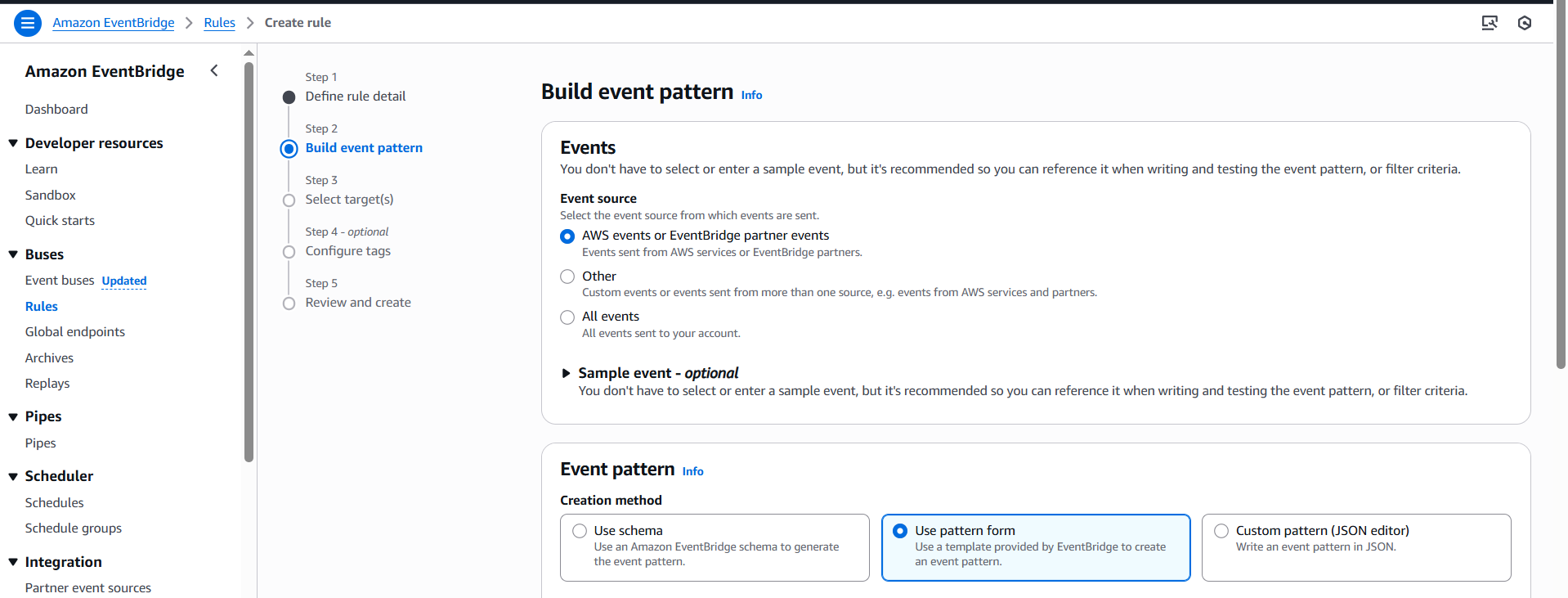
## ✅ Outcome

* You automated EBS snapshot cleanup using AWS Lambda.
* This helps in cost optimization by ensuring unused snapshots don’t keep consuming storage.

IMPLEMENTATION:

1. Create EC2. (if you have SNAPSHOTS and they are not associated to any EC2 then directly implement the lambda function)
2. Take snapshot of EC2
3. 
4. Create a lambda function “cost-opt-ebs-snap”
5. 
6. Write the Code to list out snapshots and delete.
7. 
8. Create a policy with EC2 and select write permission for delete snapshot and select list describe snapshot, describe volume, describe instances.
9. 
10. Then your lambda function → configuration → Permissions → click the role and attach policy which you created.
11. 
12. Now lambda → code → test →
13. 
14. The snapshot not deleted because its associated with EC2
15. Now delete EC2 so snapshot is available now , then lambda → code → test.
16. 
17. Now the snapshot is deleted.
18. 

**As you know lambda function is an event driven here i am showing manual implement on other hand we can do this by creating a event.**



### **We can trigger the Lambda using EventBridge:**

### **EventBridge (Scheduled Rule) → Trigger Lambda**

* Since snapshot cleanup is a **recurring task**, you can run Lambda **on a schedule** (e.g., daily or weekly).

**Steps:**

1. Go to **Amazon EventBridge → Rules → Create rule**.
2. Rule type = **Schedule**.  
   * Example: rate(1 day) → runs every 24 hours.
   * Or cron(0 0 \* \* ? \*) → runs at midnight UTC daily.
3. Target = **Your Lambda function (cost-opt-ebs-snap)**.
4. Save → Now Lambda runs automatically each day to clean unused snapshots.