**Cost Optimisation in AWS**

Cost Optimisation cannot be done only by moving towards the cloud, but some practices need to be followed, for example when we create an EC2 instance, by default volume is created and for the backup snapshots are also stored, when we delete the EC2 instance and forgot to delete the volume and snapshots and AWS keeps charging for volumes and snapshots. If the DevOps Engineer has found any such resources, they can send notifications or they can delete the instance. They will use Lambda function to write python code which will talk to API of AWS.

**Steps:**

1. Find all the EBS snapshots
2. To filter out the snapshots that are stale.

**Steps:**

1. **Creating an EC2 instance**

Volume is created along with EC2 instance

A screenshot of a computer

AI-generated content may be incorrect.

1. **Create a snapshot, where it is like storing the image.**

A screenshot of a computer

AI-generated content may be incorrect.

But the person forgot to delete the snapshot and deleted the volume and instance.

1. **Create a lambda function.**

A screenshot of a computer

AI-generated content may be incorrect.

In the code source update the code.

|  |
| --- |
| 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:  # 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.") |

Execution time for the lambda function needs to be increased, in this case it can be increased up to 10 seconds.

AWS considers execution time for billing hence try to keep it as less as possible.

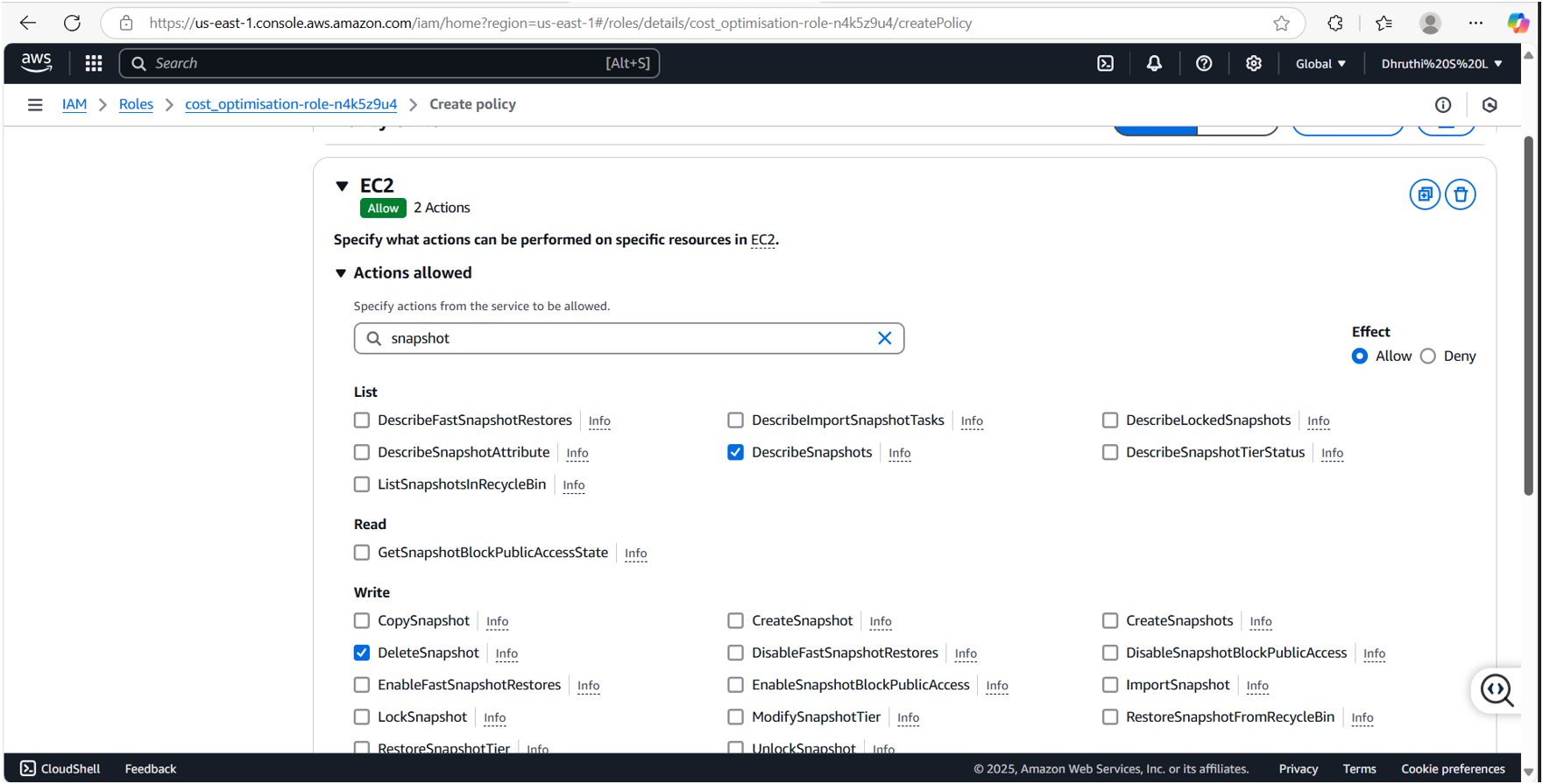
When one service tries to talk to other service it happens with the IAM roles.

1. **Attach policies to IAM.**

Some policies were not available; hence we need to create it.

Select service as EC2 for creating policy.

Attach the created policy



A computer screen shot of a computer screen

AI-generated content may be incorrect.

1. **We must give permission to EC2; hence we are forming policy again for it.**

**Describe Volume, Describe instances.**

If the instance is active the snapshots are not deleted.

A screenshot of a computer

AI-generated content may be incorrect.

1. **Deleted the EC2 instance and run the code then we can see that the snapshot is deleted.**

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.