Automated EC2 Lifecycle and Snapshot Management using AWS Lambda and Boto3

Overview

This project automates essential EC2 instance lifecycle operations—creation, start, stop, reboot, and terminate—along with automated snapshot creation for EC2 volumes. The automation is implemented using AWS Lambda functions written in Python with Boto3, scheduled using Amazon EventBridge (cron expressions). Proper IAM roles and permissions are attached to ensure secure and controlled access.

Technologies Used

- AWS Lambda
- Amazon EC2
- Amazon EBS
- Amazon EventBridge (CloudWatch Events)
- IAM Roles and Policies
- Python (Boto3 SDK)

Features Implemented

1. EC2 Instance Management:
- Create, Start, Stop, Reboot, Terminate EC2 Instances
2. Snapshot Management:
- Periodic EBS volume snapshot creation
3. Automation with EventBridge:
- Scheduled triggers using cron expressions

Task	Frequency	Cron Expression
Start Instance	Daily at 7 AM	cron(30 1 * * ? *)
Stop Instance	Daily at 8 PM	cron(30 14 * * ? *)
Snapshot Creation	Every 6 hours	cron(0 */6 * * ? *)

Sample Cron Jobs

IAM Roles & Permissions

A custom IAM Role with permissions including:

- ec2:StartInstances
- ec2:StopInstances
- ec2:RebootInstances
- ec2:TerminateInstances
- ec2:RunInstances
- ec2:DescribeInstances
- ec2:CreateSnapshot

- ec2:DescribeVolumes
- ec2:CreateTags
- logs:*

Lambda Function Example (Start EC2 Instance)

```
import boto3

def lambda_handler(event, context):
ec2 = boto3.client('ec2', region_name='us-east-1')
instance_id = 'i-0abc123def456gh78'
ec2.start_instances(InstanceIds=[instance_id])
return {
'statusCode': 200,
'body': f'Started EC2 instance {instance_id}'
}
```

Outcome

- Fully automated EC2 lifecycle management
- No manual intervention required
- Enhanced cost efficiency
- Daily backups for recovery

Future Enhancements

- Add snapshot retention logic
- Email/SNS notifications
- Dashboard using CloudWatch metrics