

## 5. Create a lambda function to start and stop ec2 instance at regular intervals

### Step 1: Create an IAM Role for Lambda

- Log in to the AWS Management Console
- Navigate to the IAM dashboard
- Click "Roles" > "Create role"
- Select "AWS service" > "Lambda"
- Click "Next: Permissions"
- Attach the "json" policy by stopping and starting the ec2 instances or choose alternate way also
- Click "Next: Review"
- Name the role (e.g., "startstoppec21")
- Click "Create role"

### Step 2: Create a Lambda Function

- Navigate to the Lambda dashboard
- Click "Create function"
- Select "Author from scratch"
- Name the function (e.g., "startstopinstance")
- Choose "Python 3.9" as the runtime
- Click "Create function"

### Step 3: Configure Lambda Function Triggers

- Click "Add trigger"
  - Select "EventBridge (CloudWatch Events)"
  - Choose "Schedule" as the event source
  - Set the schedule to your desired interval (e.g., every 1 hour)
  - Click "Add"
  - here I created rate without using cron timing. Also create cron timing.
- When to use Rate:

1. Simple scheduling needs (e.g., every 5 minutes, daily).
2. Ease of understanding and maintenance is crucial.
3. You don't need precise control over scheduling.

When to use Cron:

1. Complex scheduling needs (e.g., specific times, intervals, and exclusions).
2. Precise control over scheduling is crucial.
3. You're comfortable with cron syntax and need the flexibility it provides.

### Step 4: Write Lambda Function Code from Aws documentation

Code: To stop the ec2 instances

```
import boto3
region = 'ap-south-1'
instances = ['i-02b6c6ec82ddf2265']
ec2 = boto3.client('ec2', region_name=region)
```

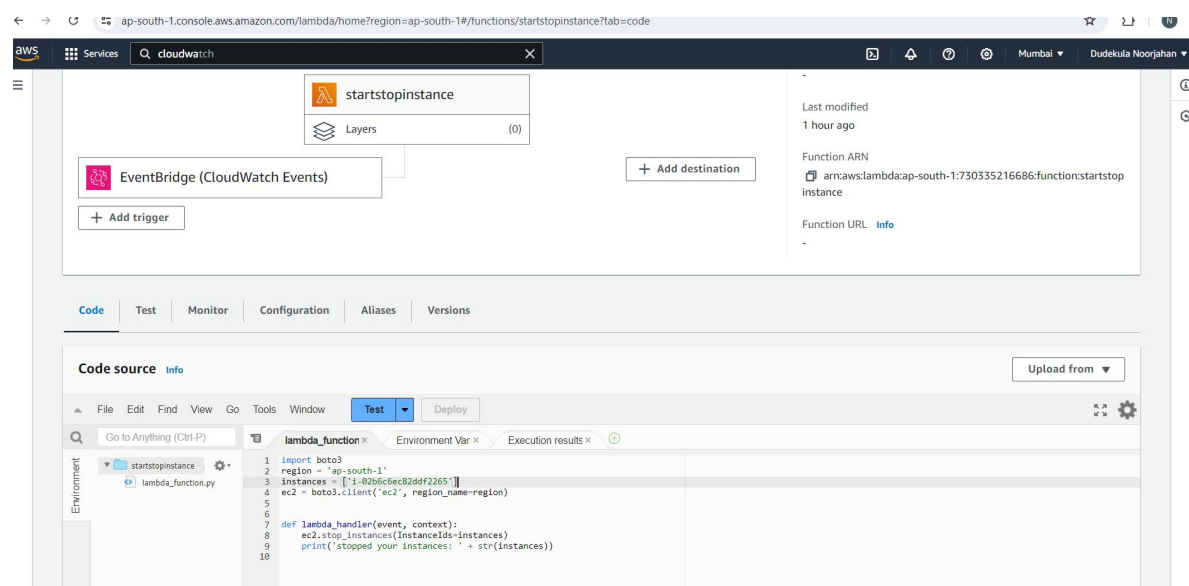
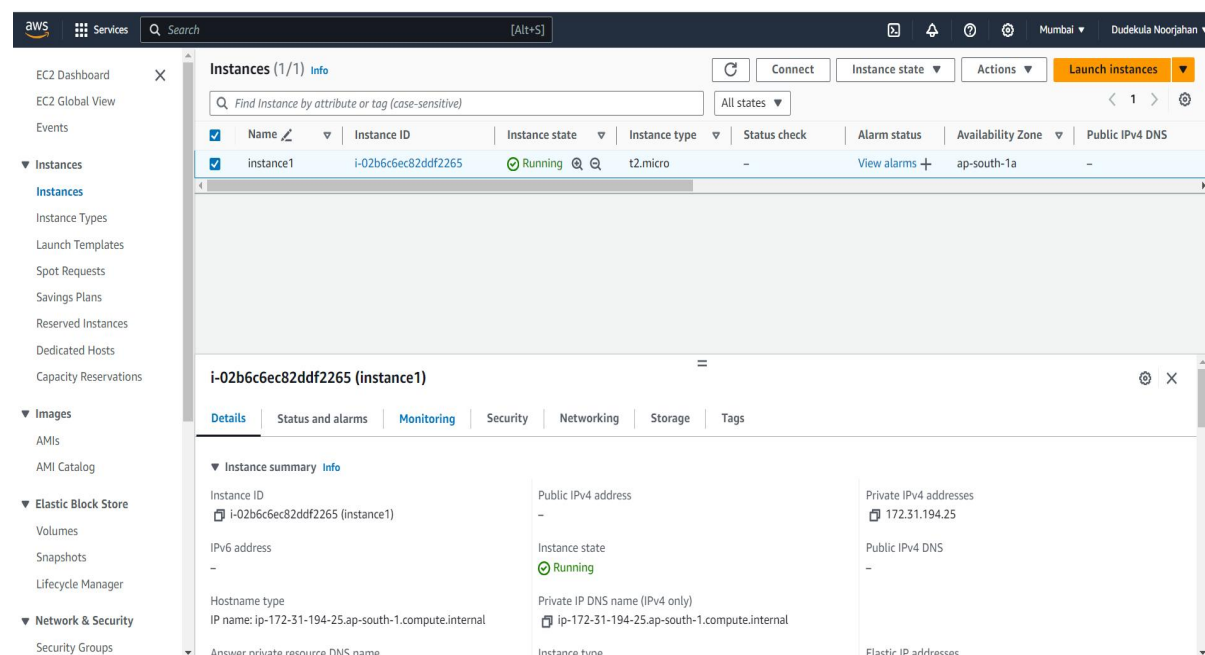
```
def lambda_handler(event, context):
    ec2.stop_instances(InstanceIds=instances)
```

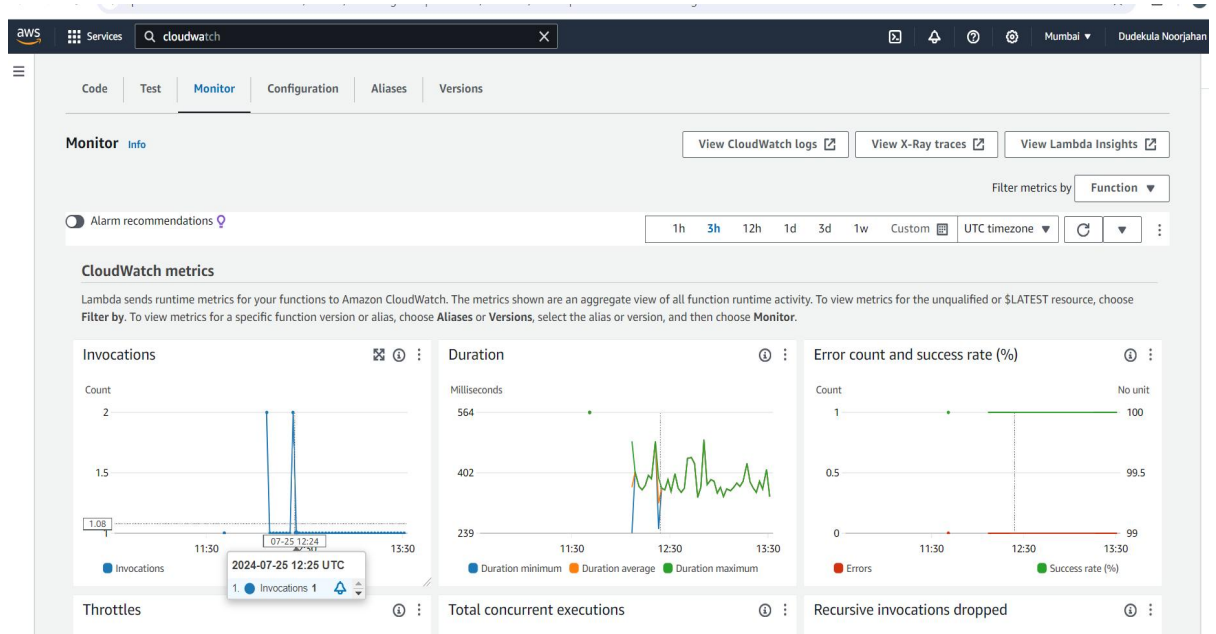
```
print('stopped your instances: ' + str(instances))
```

Code: To start the ec2 instances

```
import boto3
region = 'ap-south-1'
instances = ['i-02b6c6ec82ddf2265']
ec2 = boto3.client('ec2', region_name=region)
```

```
def lambda_handler(event, context):
    ec2.start_instances(InstanceIds=instances)
    Print('started your instances:' + str(instances))
```





The screenshot displays the AWS CloudWatch Code source interface for a Lambda function. The page shows the code source, environment variables, and execution results. The execution results section shows a successful test event named 'startstop' with a response of 'null'. The function logs show the following details: START RequestId: 73d0c313-a7d6-416c-9758-c7a8509aed09 Version: \$LATEST, stopped your instances: ['i-02b6c6c3d6f22d51'], END RequestId: 73d0c313-a7d6-416c-9758-c7a8509aed09, REPORT RequestId: 73d0c313-a7d6-416c-9758-c7a8509aed09 Duration: 247.81 ms Billed Duration: 248 ms Memory Size: 128 MB Max Memory Used: 86 MB. The code properties section shows the package size as 309 bytes, the SHA256 hash as aXqrvhpFMh5kdTJxh6hl+Kb3KGpYN/WXglANzeciXo=, and the last modified date as July 25, 2024 at 05:37 PM GMT+5:30.

## Step 6: create for the Opposite Action to start or stop our ec2instances

- Create another CloudWatch event with the opposite schedule and action

That's it! Your Lambda function should now start and stop your EC2 instance at regular intervals.