

## Deploy a Lambda Function to start EC2 on Morning 10 AM and stop at 9 PM.

To start an EC2 instance at 10 AM, stop it at 9 PM, and send an email notification at both times, you can use AWS Lambda, CloudWatch Events, and Amazon Simple Notification Service (SNS).

### **Step 1: Set Up CloudWatch Event Rules**

Create a rule to start the instance at 10 AM:

- Go to the CloudWatch console.
- Choose "Rules" from the navigation pane.
- Choose "Create rule".
- For the event source, select "Event Source" and then "Schedule".
- Enter cron(30 4 \* \* ? \*) for 10 AM UTC.
- Add the Lambda function as the target.

**Event schedule** [Info](#)

**Cron expression**  
30 4 \* \* ? \*

**Next 10 trigger date(s)**  
Wed, Jul 10, 2024, 10:00 AM GMT+5:30  
Thu, Jul 11, 2024, 10:00 AM GMT+5:30  
Fri, Jul 12, 2024, 10:00 AM GMT+5:30  
Sat, Jul 13, 2024, 10:00 AM GMT+5:30  
Sun, Jul 14, 2024, 10:00 AM GMT+5:30  
Mon, Jul 15, 2024, 10:00 AM GMT+5:30  
Tue, Jul 16, 2024, 10:00 AM GMT+5:30  
Wed, Jul 17, 2024, 10:00 AM GMT+5:30  
Thu, Jul 18, 2024, 10:00 AM GMT+5:30  
Fri, Jul 19, 2024, 10:00 AM GMT+5:30

Local time zone ▼

Create a rule to stop the instance at 9 PM:

- Follow the same steps as above, but enter cron(30 15 \* \* ? \*) for 9 PM UTC.

- Add the same Lambda function as the target.

**Event schedule** [Info](#)

Cron expression

30 15 \* \* ? \*

Next 10 trigger date(s)

UTC ▼

Tue, 09 Jul 2024 15:30:00 UTC  
Wed, 10 Jul 2024 15:30:00 UTC  
Thu, 11 Jul 2024 15:30:00 UTC  
Fri, 12 Jul 2024 15:30:00 UTC  
Sat, 13 Jul 2024 15:30:00 UTC  
Sun, 14 Jul 2024 15:30:00 UTC  
Mon, 15 Jul 2024 15:30:00 UTC  
Tue, 16 Jul 2024 15:30:00 UTC  
Wed, 17 Jul 2024 15:30:00 UTC  
Thu, 18 Jul 2024 15:30:00 UTC

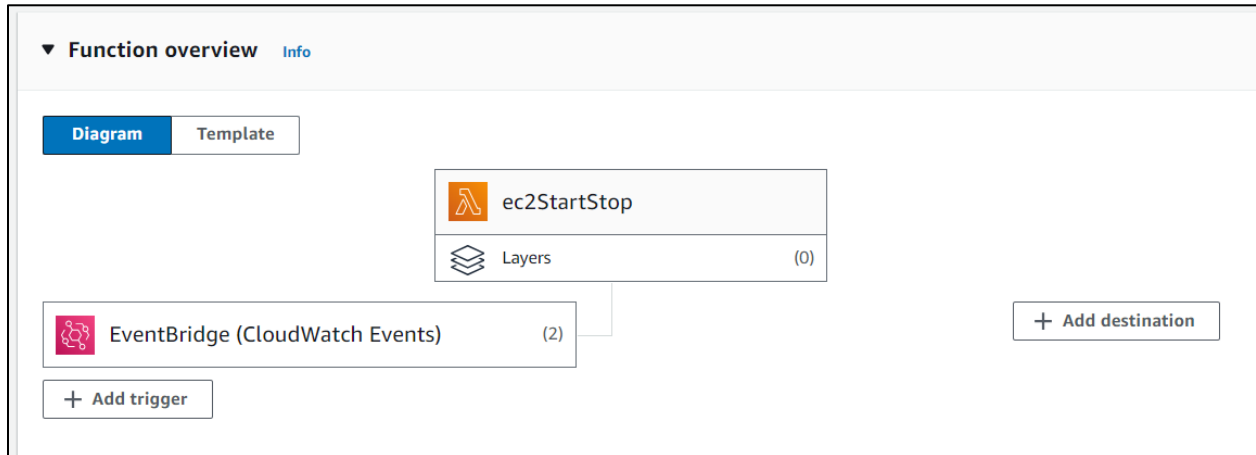
Make sure you name your CloudWatch Event rules *StartInstanceRule* and *StopInstanceRule* to match the code logic.

## Step 2: Set Up SNS Topic for Email Notifications

1. Create an SNS topic:
  - Go to the SNS console.
  - Choose "Create topic".
  - Enter a name and display name for the topic.
  - Choose "Create topic".
2. Subscribe to the SNS topic:
  - Go to the SNS console.
  - Choose the topic you created.
  - Choose "Create subscription".
  - Set the protocol to "Email".
  - Enter the email address to receive the notifications.
  - Choose "Create subscription".
  - Confirm the subscription from the email you receive.

Step 3: Write Lambda code and set the following environment variables for the Lambda function:

- INSTANCE\_ID to the ID of the instance we want to start and stop.
- SNS\_TOPIC\_ARN to the ARN of the SNS topic we created.



Add Lambda function as target in both of the CloudWatch Event rules StartInstanceRule and StopInstanceRule.

Code:

```
import json
import boto3
import os

def lambda_handler(event, context):
    ec2 = boto3.client('ec2')
    sns = boto3.client('sns')
    instance_id = os.environ['INSTANCE_ID']
    sns_topic_arn = os.environ['SNS_TOPIC_ARN']

    # Determine the action based on the rule name
    if 'rule_name' not in event:
        return {
            'statusCode': 400,
            'body': json.dumps('Missing rule_name in event')
        }

    rule_name = event['rule_name']

    if rule_name == 'StartInstanceRule':
        response = ec2.start_instances(InstanceIds=[instance_id])
```

```

    action = 'started'
    subject = 'EC2 Instance Started'
    message = f'Your EC2 instance {instance_id} has been started.'
elif rule_name == 'StopInstanceRule':
    response = ec2.stop_instances(InstanceIds=[instance_id])
    action = 'stopped'
    subject = 'EC2 Instance Stopped'
    message = f'Your EC2 instance {instance_id} has been stopped.'
else:
    return {
        'statusCode': 400,
        'body': json.dumps('Invalid rule name')
    }

# Send email notification
sns.publish(
    TopicArn=sns_topic_arn,
    Subject=subject,
    Message=message
)
return {
    'statusCode': 200,
    'body': json.dumps('Hello from Lambda!')
}

```

IAM Role to be attached to Lambda:

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:StartInstances",
        "ec2:StopInstances"
      ],
      "Resource": "arn of EC2"
    },
    {
      "Effect": "Allow",
      "Action": "sns:Publish",

```

```
"Resource": "arn of SNS topic"
    }
  ]
}
```

## Output:

