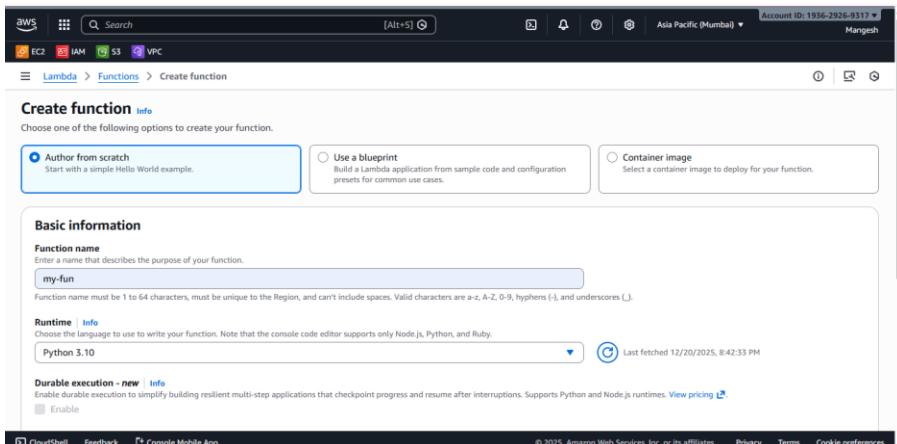


Auto Start Stop instance by using Lambda Function and Amazon EventBridge.

Steps:

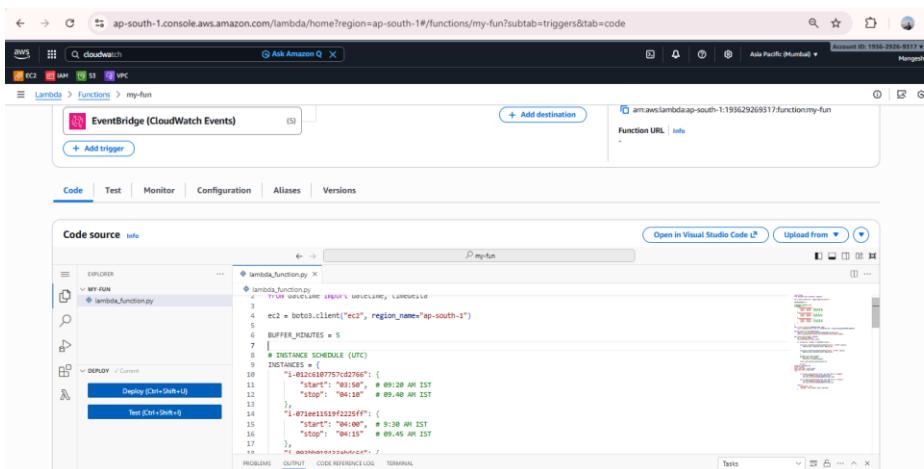
Create a Lambda Function. Select the language Python 3.10



Paste the code in code editor.

```
1 import boto3
2 from datetime import datetime, timedelta
3
4 ec2 = boto3.client("ec2", region_name="ap-south-1")
5
6 BUFFER_MINUTES = 5
7 # INSTANCE SCHEDULE (UTC)
8 INSTANCES = {
9     "i-012c6107757cd2766": {
10         "start": "03:50", # 09:20 AM IST
11         "stop": "04:10" # 09:40 AM IST
12     },
13     "i-071ee11519f2225ff": {
14         "start": "04:00", # 9:30 AM IST
15         "stop": "04:15" # 09:45 AM IST
16     },
17     "i-093bb018433abdc64": {
18         "start": "04:40", # 10:10 AM IST
19         "stop": "18:40" # 12:05 AM IST
20     }
21 }
22
23 def is_time_to_execute(scheduled_time, now):
24     return scheduled_time <= now <= (scheduled_time + timedelta(minutes=BUFFER_MINUTES))
25
26 def get_instance_state(instance_id):
27     response = ec2.describe_instances(InstanceIds=[instance_id])
28     return response["Reservations"][0]["Instances"][0]["State"]["Name"]
29
30 def lambda_handler(event, context):
31     now = datetime.utcnow()
32     print(f"Current UTC Time: {now}")
33
34     for instance_id, schedule in INSTANCES.items():
35
36         start_time = datetime.strptime(schedule["start"], "%H:%M").replace(
37             year=now.year, month=now.month, day=now.day
38         )
39
40         stop_time = datetime.strptime(schedule["stop"], "%H:%M").replace(
41             year=now.year, month=now.month, day=now.day
42         )
43
44         # Handle stop after midnight
45         if stop_time < start_time:
46             stop_time += timedelta(days=1)
47
48         state = get_instance_state(instance_id)
49
50         print(f"""
51 Instance: {instance_id}
52 State: {state}
53 Start Time (UTC): {start_time}
54 Stop Time (UTC): {stop_time}
55 """)
56
57         if is_time_to_execute(start_time, now) and state == "stopped":
58             print(f"STARTING instance {instance_id}")
59             ec2.start_instances(InstanceIds=[instance_id])
60
61         elif is_time_to_execute(stop_time, now) and state == "running":
62             print(f"STOPPING instance {instance_id}")
63             ec2.stop_instances(InstanceIds=[instance_id])
64
65     return {
66         "statusCode": 200,
67         "message": "EC2 schedule check completed"
68     }
69 }
```

Deploy the code.



Create a new test event.

Go into Configuration → General Configuration → Edit → Time-Out → increase upto 30 second.

The screenshot shows the 'Edit basic settings' page for a Lambda function named 'my-fun'. Key configuration details include:

- Memory:** 128 MB
- Ephemeral storage:** 512 MB
- Execution role:** Lambda-EC2-FullAccess
- Timeout:** 30 seconds
- Snapshot:** None
- Existing role:** Use an existing role (selected)
- Existing role dropdown:** Lambda-EC2-FullAccess

Go into Configuration → Environment Variables → Edit → Key = TZ (Time Zone) , Pair = UTC

The screenshot shows the 'Edit environment variables' page for the same Lambda function. A single environment variable is defined:

Key	Value
TZ	UTC

Go into another service Amazon EventBridge → Go into Rules → Create Scheduled Rule

The screenshot shows the 'Rules' page in the Amazon EventBridge console. It lists five rules on the 'default' event bus. The interface includes filters for Name, Status, Type, Event bus, ARN, and Description.

Define a rule → Put the details in step-1

The screenshot shows the 'Create rule' process at Step 1: 'Define rule detail'. The user has selected the 'Scheduled rule detail' option. The configuration includes:

- Name:** server-1-Start
- Description - optional:** 10am IST = 4:30 am UTC
- Event bus:** default (disabled)
- Enable the rule on the selected event bus:** Enabled
- EventBridge Scheduler - A new AWS scheduling capability!** (Info box)

Write a crontab expression in next step that trigger an lambda function on Specific time in UTC.

The screenshot shows the 'Define schedule' step of creating a new rule. The 'Schedule pattern' tab is selected, showing a cron expression 'cron(04 * * ? * 1)'. The 'Cron expression' tab shows the expanded cron values: Minutes: 04, Hours: 04, Day of month: *, Month: ?, Day of week: 1, Year: *. The interface also displays the next 10 trigger dates from Fri, 19 Dec 2025 04:30:00 UTC to Sun, 28 Dec 2025 04:30:00 UTC.

In Step 3. Select Target. In the execution role Select → Create new role for this specific resource.

The screenshot shows the 'Select target(s)' step. A new target is being created, named 'Amazon_EventBridge_Invoke_Lambda_224477970'. The target type is set to 'AWS service' and the function is 'my-lfn'. The 'Create a new role for this specific resource' option is selected. The 'Role name' field contains the same target name.

In step-4 . Tags

The screenshot shows the 'Configure tags - optional' step. It indicates that no tags are associated with the resource. An 'Add new tag' button is available to add up to 50 more tags.

In step-5. Review and create.

The screenshot shows the 'Review and create' step. It summarizes the rule details: Rule name 'server-1-start', Status 'Enabled', Description '10am IST - 4.30 am UTC', and Event bus 'default'. The 'Event schedule' section shows the cron expression '0 04 * * *'. The 'Step 3: Select target(s)' section shows the target 'Targets'.

Rule created.

Do same steps for creating 6 Rules in EventBridge.

Name	Status	Type	Event bus	ARN	Description
server-1-Start	Enabled	Scheduled Standard	default	arn:aws:event:sap-south-1:193629269317:rule/server-1-Start	9.40
server-1-Stop	Enabled	Scheduled Standard	default	arn:aws:event:sap-south-1:193629269317:rule/server-1-Stop	9.50
server-2-Start	Enabled	Scheduled Standard	default	arn:aws:event:sap-south-1:193629269317:rule/server-2-Start	9.50
server-2-Stop	Enabled	Scheduled Standard	default	arn:aws:event:sap-south-1:193629269317:rule/server-2-Stop	10.00
server-3-Start	Enabled	Scheduled Standard	default	arn:aws:event:sap-south-1:193629269317:rule/server-3-Start	10.00
server-3-Stop	Enabled	Scheduled Standard	default	arn:aws:event:sap-south-1:193629269317:rule/server-3-Stop	10.00

Go into Configuration → Trigger → Select EventBridge (CloudWatch Events) → Add Rules to trigger the functions.

Add trigger here . Select the eventbridge rule here for triggering and add.

Trigger added.

Add next trigger do same process. Here we add 5 trigger means the rules we created in Amazon EventBridge service.

The screenshot shows the AWS Lambda console with the path `Lambda > Functions > my-ec2-function`. The left sidebar is collapsed. The main area is titled "Triggers (6) Info". It lists four triggers under the "Trigger" section:

- EventBridge (CloudWatch Events): server-1-Start**: Rule state: **ENABLED**
- EventBridge (CloudWatch Events): server-1-Stop**: Rule state: **ENABLED**
- EventBridge (CloudWatch Events): server-2-Start**: Rule state: **ENABLED**
- EventBridge (CloudWatch Events): server-2-Stop**: Rule state: **ENABLED**

Here We done all the process.

Lets check.

The screenshot shows the AWS EC2 Instances page with the path `EC2 > Instances`. The left sidebar shows "Instances" expanded, with "Instances" selected. The main table displays three instances:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability
server-1	i-026625c3d39fbfcff	Stopping	t3.micro	-	View alarms +	ap-south-1t
server-2	i-0e47b3f05c7110c0e	Running	t3.micro	Initializing	View alarms +	ap-south-1t
server-3	i-07aca339ab12dfae5	Stopped	t3.micro	-	View alarms +	ap-south-1t