

Lab Assignment 6

A.6.1: Start/stop EC2 instances using AWS Lambda and Cloudwatch Events Scheduler

Task 6.1.1: Create one Windows/Linux instance.

Task 6.1.2: Open IAM service and create role to access lambda services.

Steps:

- Select AWS services
- Choose lambda use case.
- Click on permission and use two policies: AWSLambdaBasicExecutionRole (allow to use cloudwatch) and AmazonEC2Full Access.
Note: You can create your policies, or you can select existing policies
- Provide tag.
- Provide Role name and role description.

Task 6.1.3: Open Lambda services and create a new lambda function.

Steps:

- Provide function name.
- Use Python as runtime environment.
- Go to permissions and use the created role in Task 2.
- Now basic function code window will appear, and we need to write code for starting and stopping EC2 instances.

Code:

Start instance:

```
import boto3

region = 'us-east-2'

instances = ['i-0f019ff0664238dcb']

ec2 = boto3.client('ec2', region_name=region)

def lambda_handler(event, context):

    ec2.start_instances(InstanceIds=instances)

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

Stop Instance:

```
import boto3
```

```

region = 'us-east-2'

instances = ['i-0f019ff0664238dcb']

ec2 = boto3.client('ec2', region_name=region)

def lambda_handler(event, context):

    ec2.stop_instances(InstanceIds=instances)

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

```

Task 6.1.4: Save the code. Select a test event and configure test event.

Task 6.1.5: Test the EC2events by clicking on startEC2 event and stopEC2 event.

Task 6.1.6: Provide the execution details of the lambda function.

Task 6.1.7: Select the event bridge and click on create rule to trigger the created lambda function.

Steps:

- Provide name and description for the rule.
- Define pattern as event by providing Cron expression.
- Select targets as Lambda function, provide name of the function, copy the json text from the event window.
- Create the rule now.

Task 6.1.8: Take the snapshots of all performed tasks and merge with A.6.1 snapshots. Then create a doc/pdf of your enrolment number_lab05(Ex: E18CSE072_Lab06) and upload the file on LMS.

Web link:

<https://aws.amazon.com/premiumsupport/knowledge-center/start-stop-lambda-cloudwatch/>

<https://docs.aws.amazon.com/lambda/latest/dg/services-cloudwatchevents.html>

YouTube video link:

<https://www.youtube.com/watch?v=instSVC6gk0&feature=youtu.be>

<https://www.youtube.com/watch?v=oISuE16pGFQ>

<https://www.youtube.com/watch?v=bv-CKOMPcPI>

A.6.2. Create Alexa Skills with AWS Lambda using python.

Task 6.2.1: Go to alexa developer console and login to that console by providing your credential.

<https://developer.amazon.com/alexa/console/ask>

Task 6.2.2: Open alexa developer console and click on create skill.

Steps:

- Provide skill name.
- Choose language.
- Choose custom model to add your skill.
- Choose one language to host your skill's backend resources- choose python.
- Choose "start from scratch" template to add to your skill.

Task 6.2.3: Now, build your skill.

Steps:

- Go to Invocation and provide skill invocation name.
- Click on save model.
- Click on build model.

Task 6.2.4: Now, go to code and type your message in HelloWorldIntentHandler. Then, click on deploy button.

Task 6.2.5: Go to Test button. Initially, test is disabled. Make it enable. Then, type invocation name and after that you need to type hello. Your message will be displayed and converted into voice message.

Task 6.2.6: Take the snapshots of all performed tasks and merge with A.6.1 snapshots. Then create a doc/pdf of your enrolment number_lab05(Ex: E18CSE072_Lab06) and upload the file on LMS.

Web link:

<https://realpython.com/alexa-python-skill/>

YouTube video link:

<https://www.youtube.com/watch?v=mUgQKrJfDKw>

<https://www.youtube.com/watch?v=RzS69XCe9cw&feature=youtu.be>