Advance DevOps
Experiment No. 6B

Name: Shariya Ansari

Roll No: 03

Class: IT SEM V

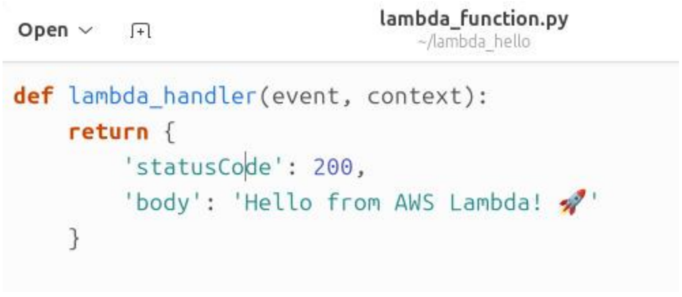
Date: 5/10/25

Aim: A tiny sample serverless computing application on AWS Lambda and using AWS CLI.

Procedure:

1. Create a simple Lambda handler function

Write a Python function named lambda_handler that returns a sample JSON response. Save this in a file named lambda_function.py.

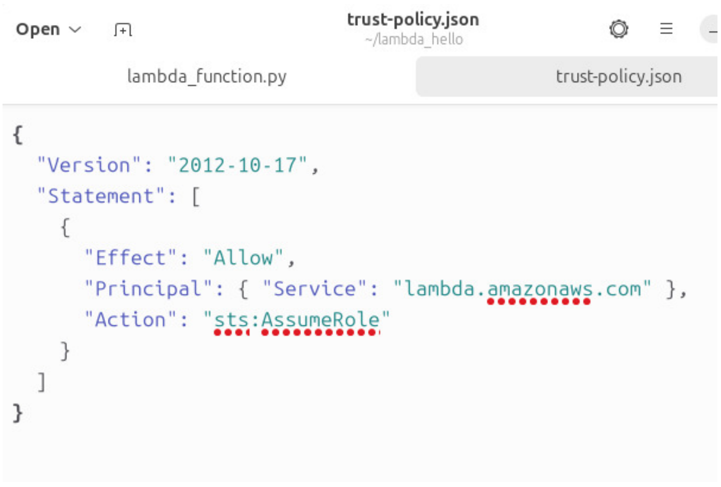


```python
def lambda_handler(event, context):
    return {
        'statusCode': 200,
        'body': 'Hello from AWS Lambda! 🚀'
    }
```

2. Prepare the trust policy JSON

Create a trust-policy.json file. This policy defines permissions for AWS Lambda to assume a role.

Example:



```json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": { "Service": "lambda.amazonaws.com" },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

3. Package your Lambda function

(If your function has dependencies, zip all files. For this simple function, just zip the Python file.)

```
zip function.zip lambda_function.py
```

4. Create an IAM role for Lambda

Use AWS CLI to create a role that Lambda can assume.

bash

```
aws iam create-role --role-name lambda-ex --assume-role-policy-document file://trust-policy.json
```

5. Attach the basic execution policy

bash

```
aws iam attach-role-policy --role-name lambda-ex --policy-arn arn:aws:iam::aws:policy/service-role/AWSLambdaBasicExecutionRole
```

6. Create the Lambda function

Use the role's ARN obtained in step 4.

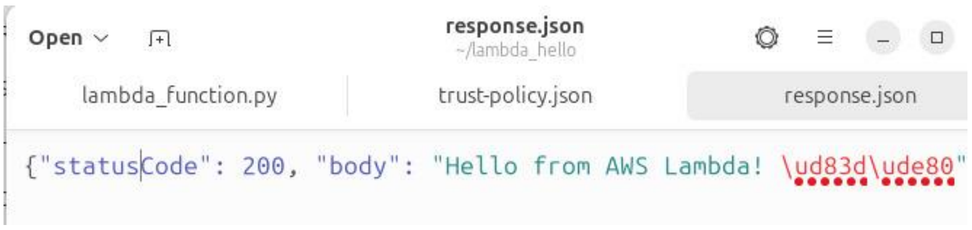
```
aws lambda create-function \
  --function-name hello-lambda \
  --runtime python3.9 \
  --role arn:aws:iam::537940551828:role/lambda-ex-role \
  --handler lambda_function.lambda_handler \
  --zip-file fileb://function.zip
```

7. Test the Lambda function

Invoke your Lambda and store the output in a file, e.g., response.json.

bash

```
aws lambda invoke --function-name hello-cli response.json
```



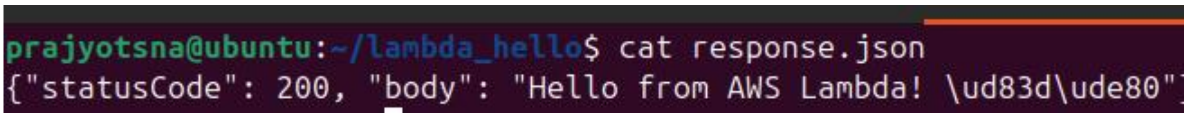
```
{"statusCode": 200, "body": "Hello from AWS Lambda! \ud83d\ude80"
```

8. View the output

Check the JSON output to confirm the function executed and returned the expected result.

bash

cat response.json



```
prajyotsna@ubuntu:~/lambda_hello$ cat response.json
{"statusCode": 200, "body": "Hello from AWS Lambda! \ud83d\ude80"
```

Conclusion: AWS Lambda simplifies cloud application deployment by eliminating server management, scaling automatically with demand, and charging only for actual usage, making development fast and cost-effective for a wide range of tasks and users.