

Lab Assignment-6

ECSE373L: Cloud Infrastructure and Services

Name: Tanvi Penumudy

Enroll no: E18CSE187

Batch: EB02

Lab Activities:

Task 0: Create the IAM Role of Full DynamoDB and CloudWatch service using IAM service.

Task 1: Create Lambda function and provide the created role as in task 0.

Task 2: Write the code in the Lambda function as given in sample code.

Task 3: Go to DynamoDB database service of AWS and Create a table named as weather and key name as city.

Task 4: Create the items in the weather table or you can fill automatically.

Example:

City	Temp
Delhi	18
Mumbai	25
London	5
Italy	3

Task 5: Create an Amazon API Gateway by selecting REST Architecture. API name you can choose as per your choice. (Endpoint type should be regional)

Task 6: Then Create API resource (any relevant name you can give)

Task 7: Create a GET method and select Lambda function as integration type and tick the Use Lambda Proxy Integration then provide the created Lambda function name.

Task 8: Now deploy the API.

Task 9: Then Test the deployed API by using invoke URL of API.

Task 10: Take the snapshots of all performed tasks and create a doc/pdf of your enrolment number_lab06 and upload the file on LMS.

Task 0

IAM Management Console

console.aws.amazon.com/iam/home?region=ap-south-1#/roles\$new?step=review&commonUseCase=Lambda%2BLambda&selected...

Create role

1 2 3 4

Review

Provide the required information below and review this role before you create it.

Role name* Lab6E18CSE187CIS
Use alphanumeric and '+', '@', '_' characters. Maximum 64 characters.

Role description Allows Lambda functions to call AWS services on your behalf.
Maximum 1000 characters. Use alphanumeric and '+', '@', '_' characters.

Trusted entities AWS service: lambda.amazonaws.com

Policies CloudWatchFullAccess AmazonDynamoDBFullAccess

Permissions boundary Permissions boundary is not set

No tags were added.

* Required Cancel Previous **Create role**

Feedback English (US) © 2008 - 2021, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use Cookie preferences

Type here to search

Task 1

Lambda

ap-south-1.console.aws.amazon.com/lambda/home?region=ap-south-1#/create/function

Basic information

Function name
Enter a name that describes the purpose of your function.
Lab6E18CSE187
Use only letters, numbers, hyphens, or underscores with no spaces.

Runtime info
Choose the language to use to write your function.
Python 3.8

Permissions info
By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.

Change default execution role

Execution role
Choose a role that defines the permissions of your function. To create a custom role, go to the IAM console.

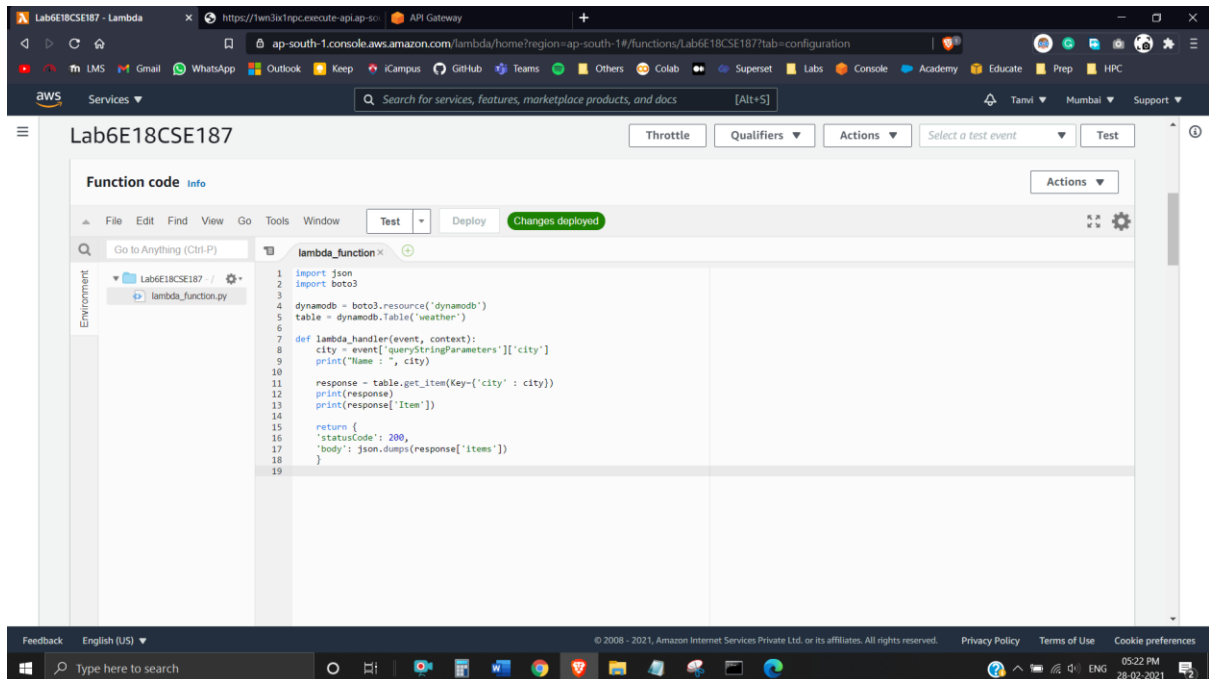
☐ Create a new role with basic Lambda permissions
☒ Use an existing role
☐ Create a new role from AWS policy templates

Existing role
Choose an existing role that you've created to be used with this Lambda function. The role must have permission to upload logs to Amazon CloudWatch Logs.
Lab6E18CSE187CIS
View the Lab6E18CSE187CIS role on the IAM console.

Feedback English (US) © 2008 - 2021, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use Cookie preferences

Type here to search

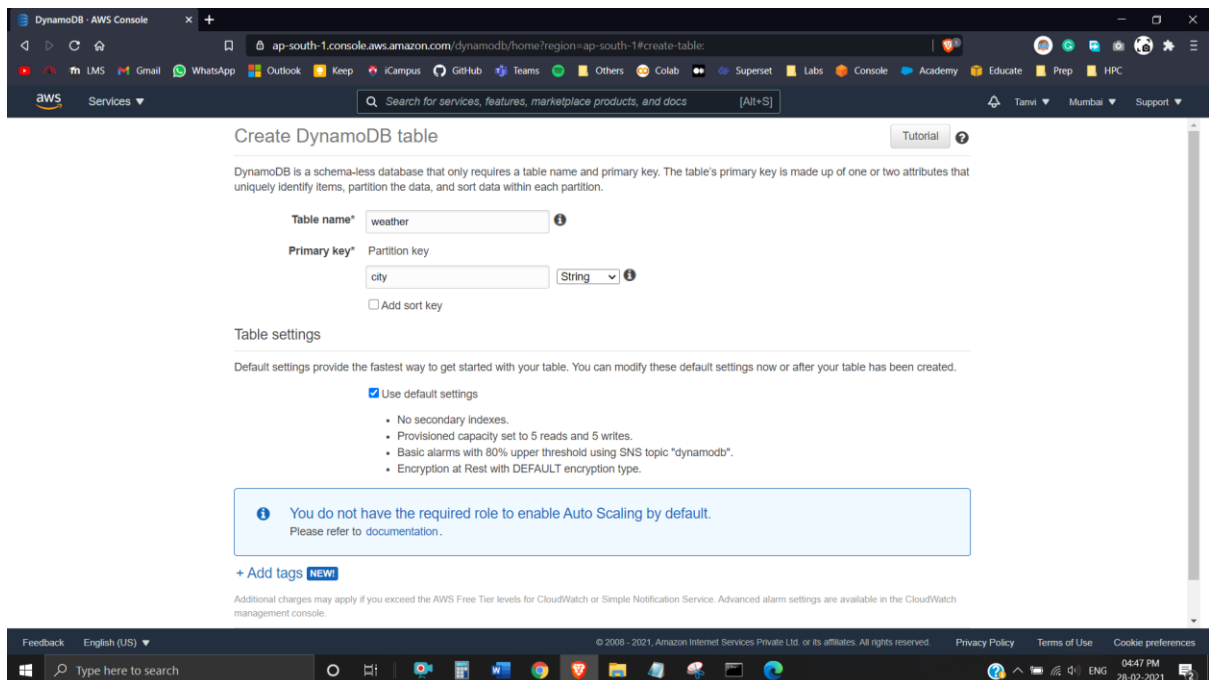
Task 2



The screenshot shows the AWS Lambda console for the function 'Lab6E18CSE187'. The function code is displayed in a text editor, showing a Python script that uses boto3 to interact with a DynamoDB table named 'weather'.

```
1 import json
2 import boto3
3
4 dynamodb = boto3.resource('dynamodb')
5 table = dynamodb.Table('weather')
6
7 def lambda_handler(event, context):
8     city = event['queryStringParameters']['city']
9     print("Name : ", city)
10
11     response = table.get_item(Key={'city': city})
12     print(response)
13     print(response['Item'])
14
15     return {
16         'statusCode': 200,
17         'body': json.dumps(response['Item'])
18     }
19
```

Task 3



The screenshot shows the 'Create DynamoDB table' wizard in the AWS console. The table name is 'weather', the primary key is 'city', and the table settings are configured with default settings.

Create DynamoDB table

DynamoDB is a schema-less database that only requires a table name and primary key. The table's primary key is made up of one or two attributes that uniquely identify items, partition the data, and sort data within each partition.

Table name* weather

Primary key* Partition key

city

☐ Add sort key

Table settings

Default settings provide the fastest way to get started with your table. You can modify these default settings now or after your table has been created.

☒ Use default settings

- No secondary indexes.
- Provisioned capacity set to 5 reads and 5 writes.
- Basic alarms with 80% upper threshold using SNS topic "dynamodb".
- Encryption at Rest with DEFAULT encryption type.

You do not have the required role to enable Auto Scaling by default.
Please refer to documentation.

+ Add tags

Additional charges may apply if you exceed the AWS Free Tier levels for CloudWatch or Simple Notification Service. Advanced alarm settings are available in the CloudWatch management console.

Task 4

The screenshot shows the AWS DynamoDB console interface. On the left, there's a sidebar with 'Create table' and 'Delete table' buttons, and a search bar for table names. The main area displays the 'weather' table with a 'Scan' operation. The scan results show a table with two columns: 'city' and 'temp'. The data is as follows:

city	temp
Delhi	18
Italy	3
London	5
Mumbai	25

Task 5

The screenshot shows the AWS API Gateway console interface. The 'Create new API' page is displayed, with the 'New API' option selected. The settings are as follows:

API name*: Lab6E18CSE187

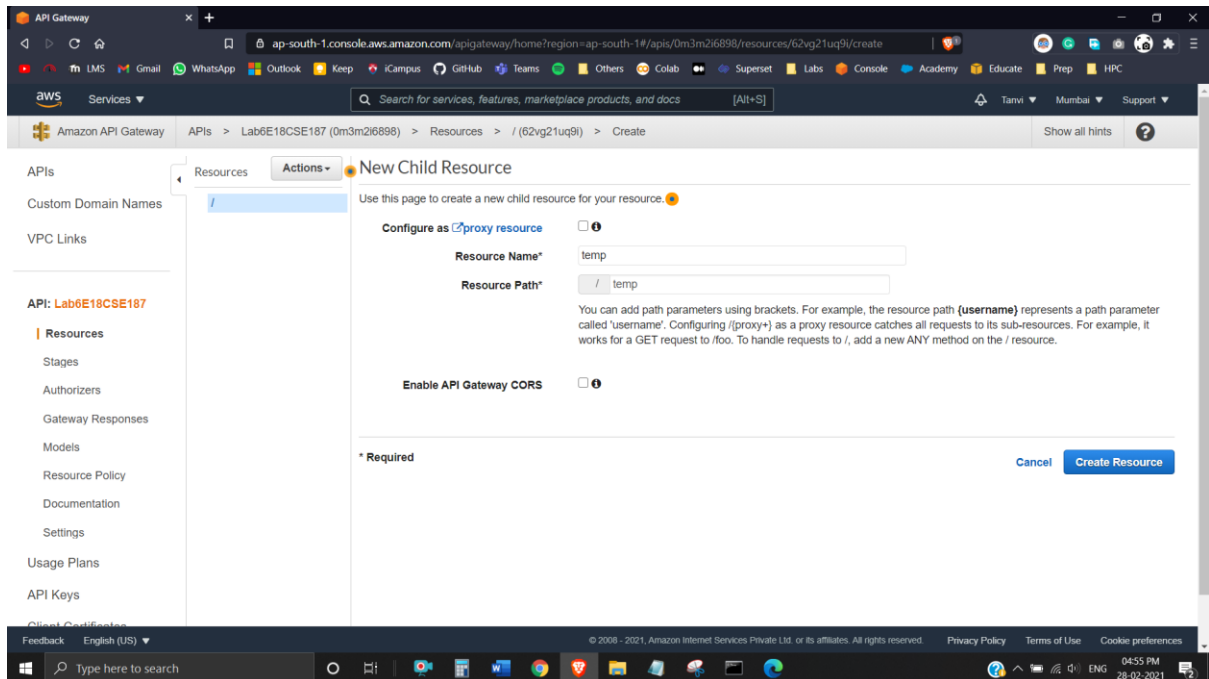
Description:

Endpoint Type: Regional

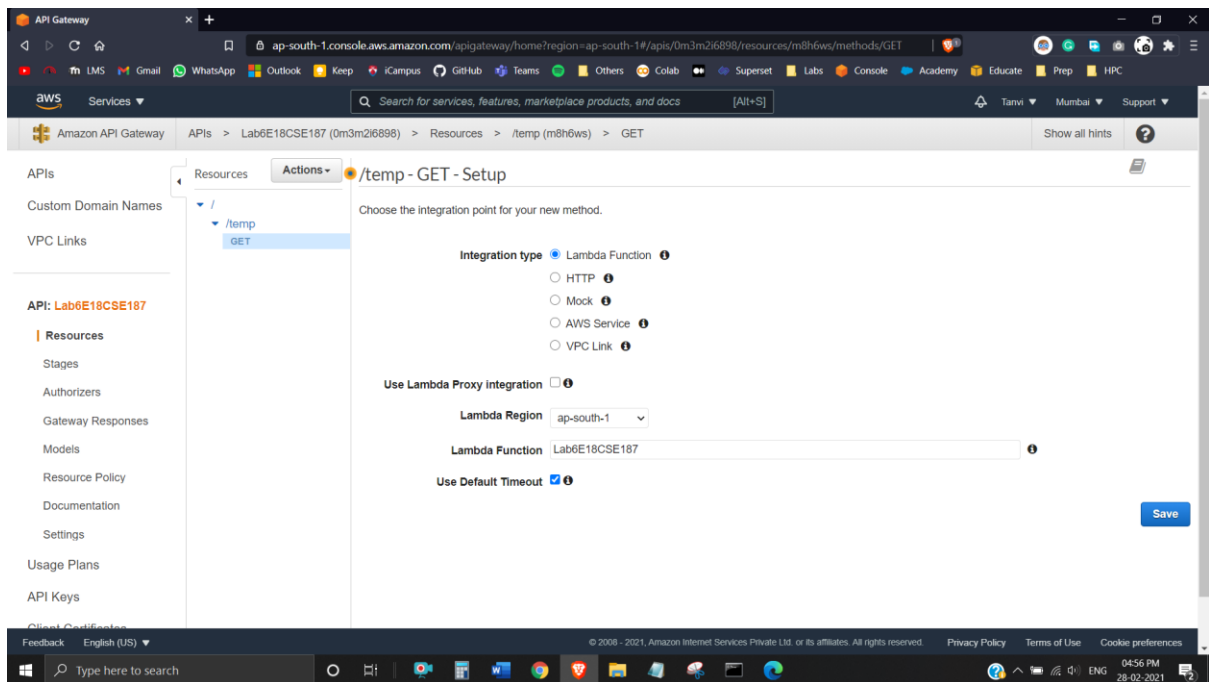
* Required

Create API

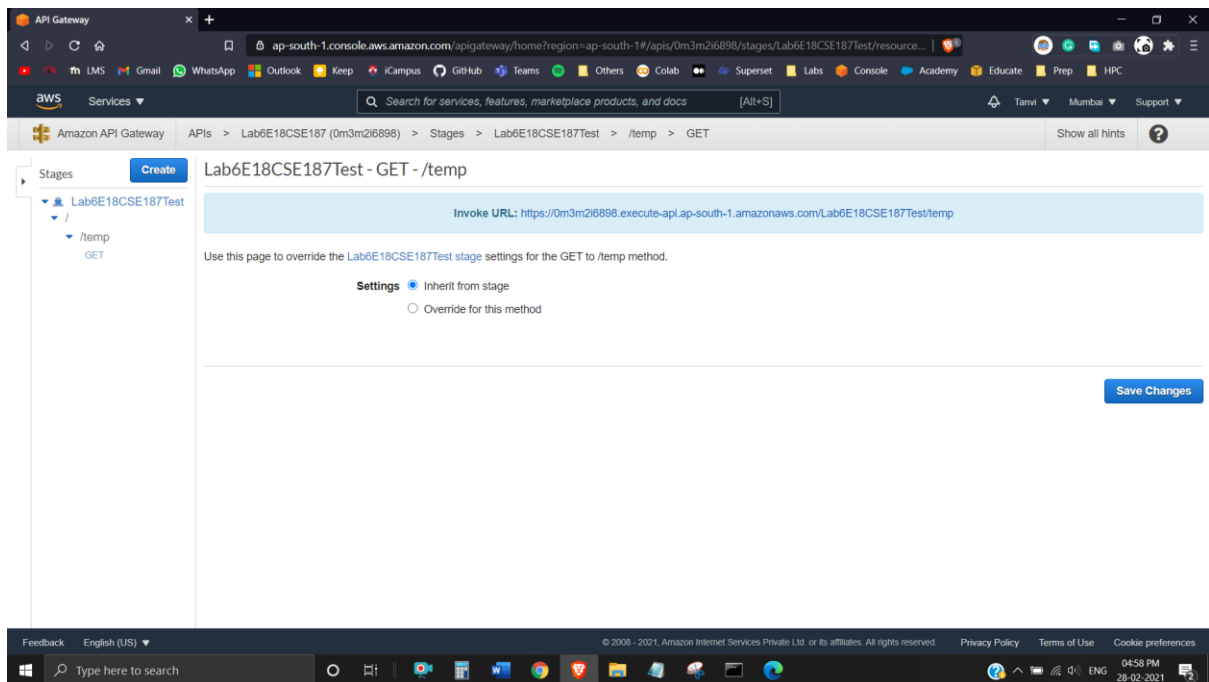
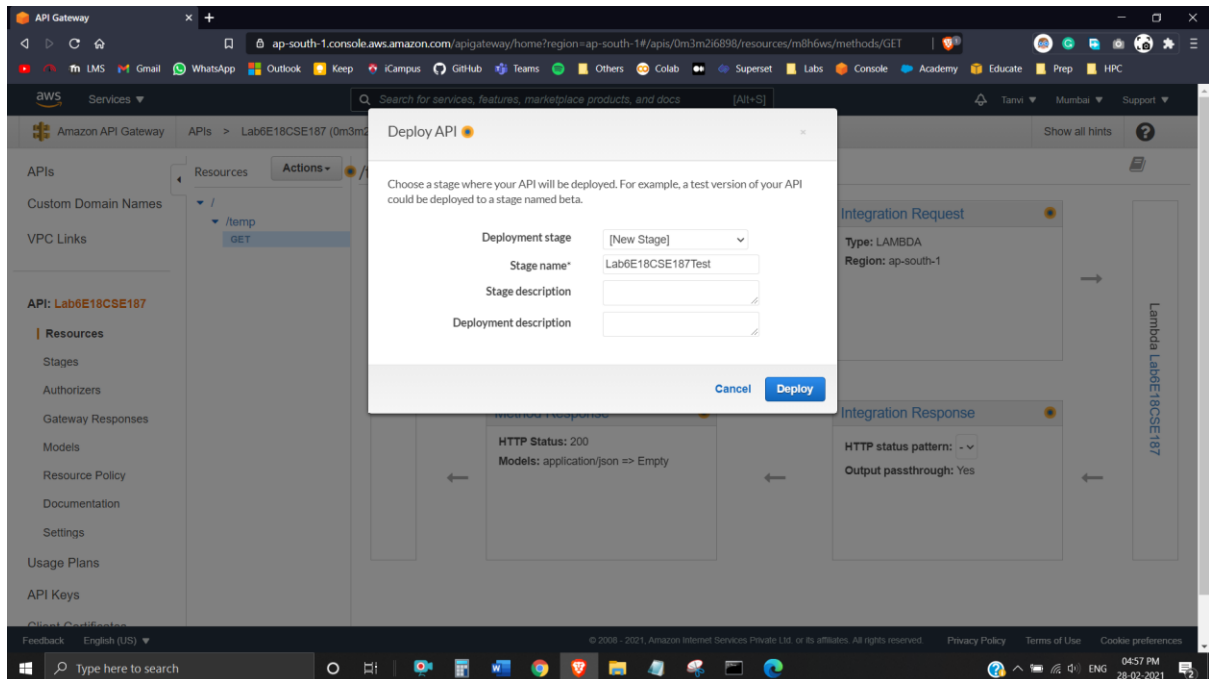
Task 6



Task 7



Task 8



Task 9

