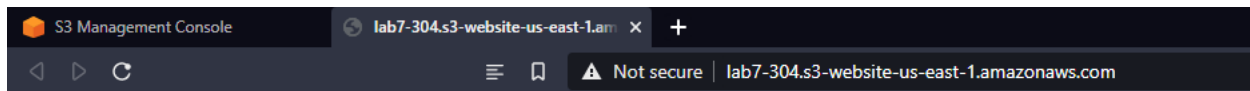
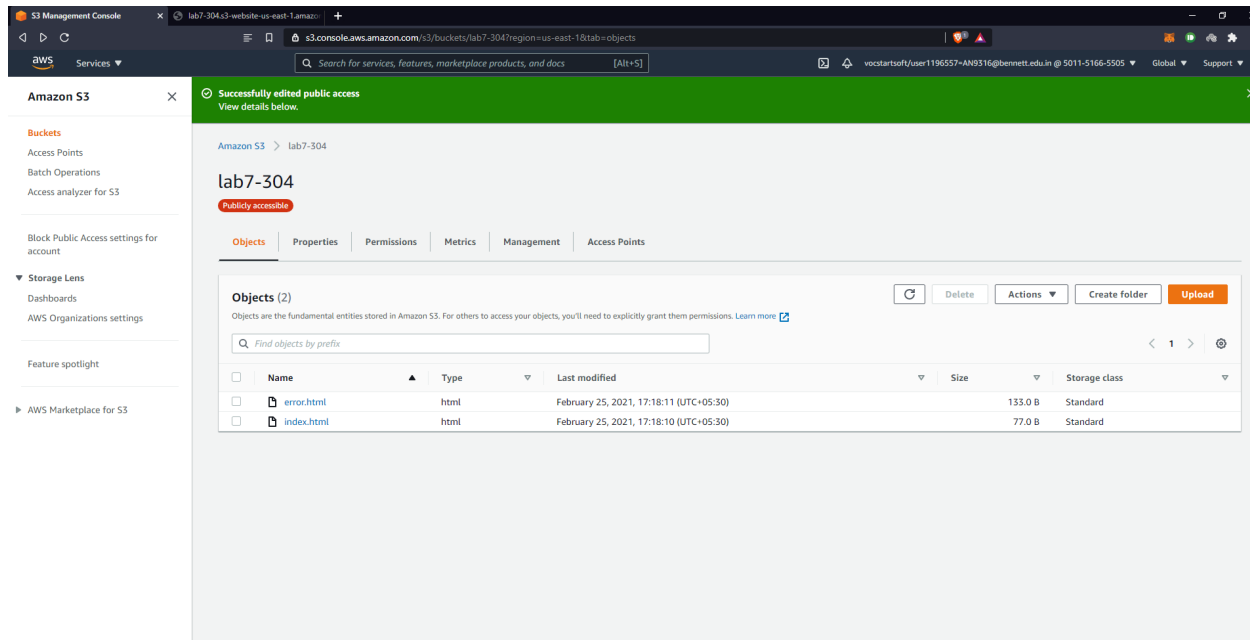


# Cloud Computing – Lab 7

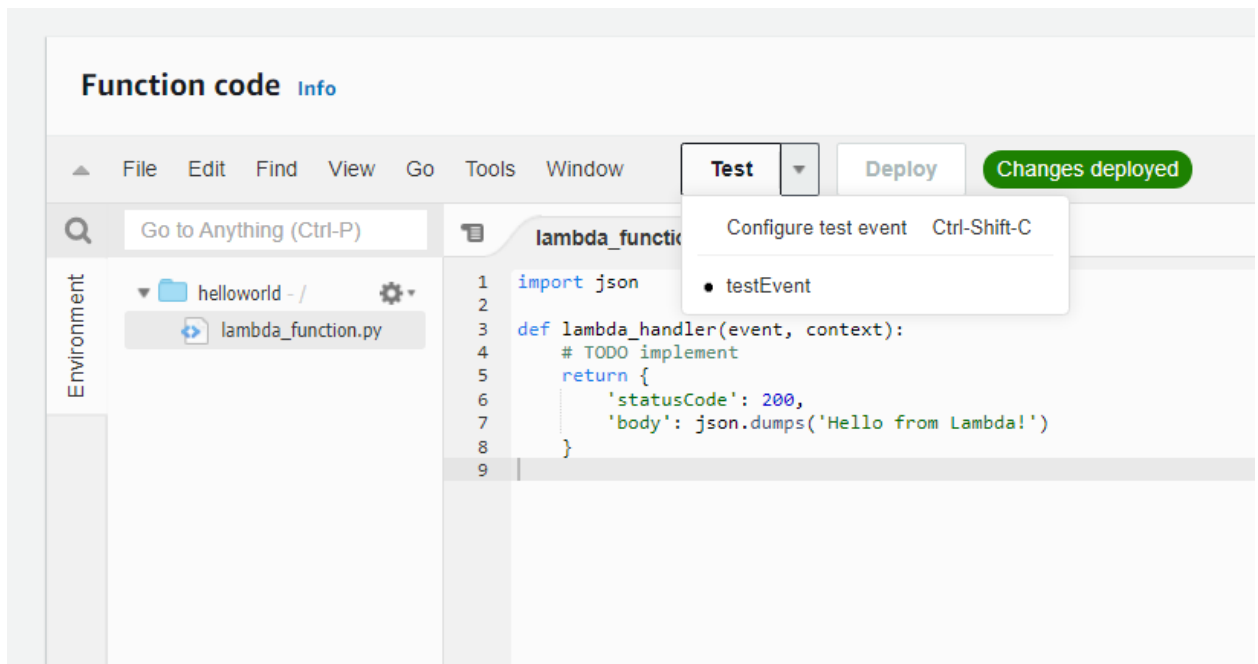
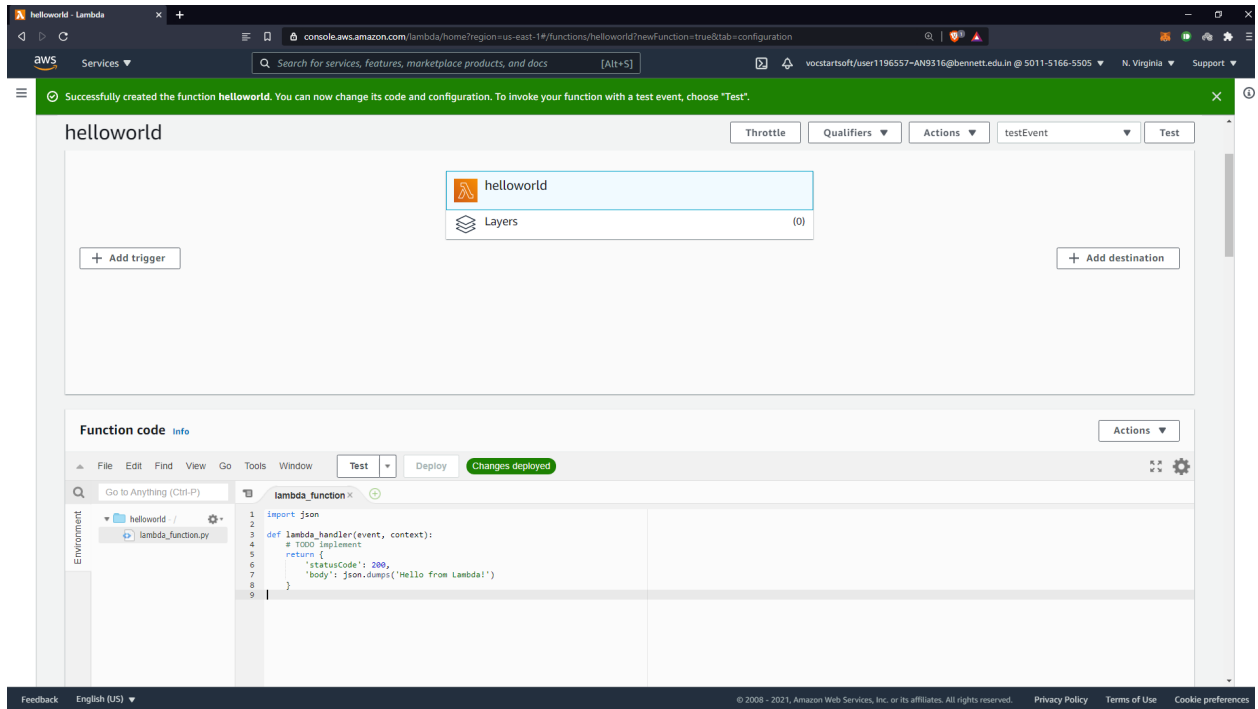
## Anudit Nagar – E18CSE024

### Task 1 – Create Bucket, Upload files, Enable Static Site hosting

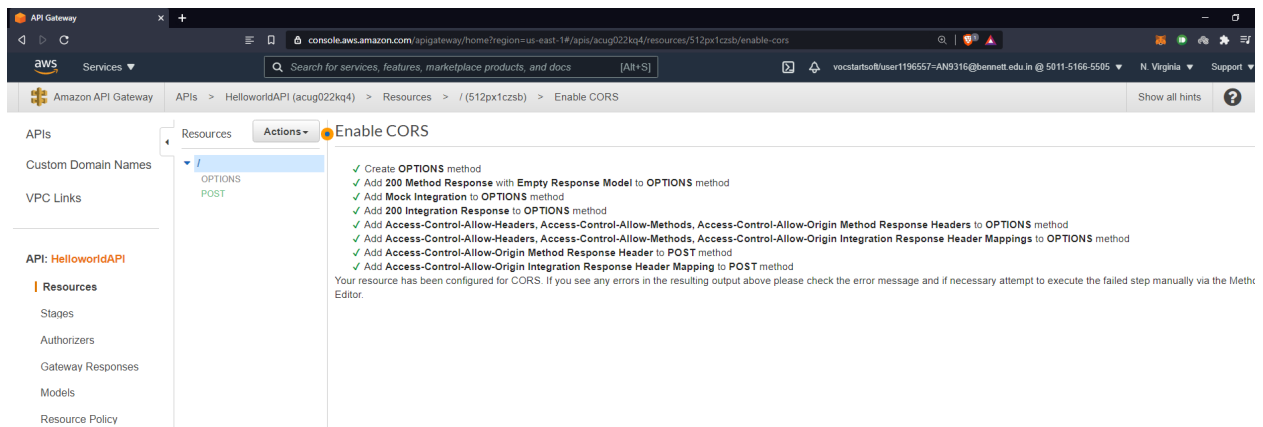
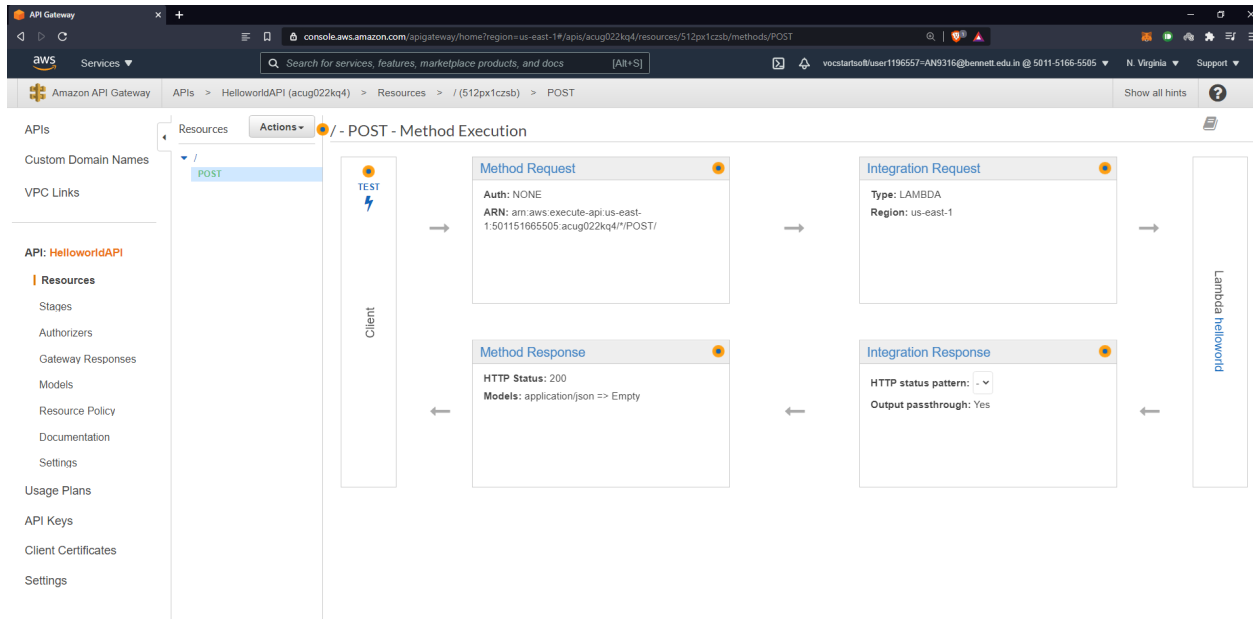


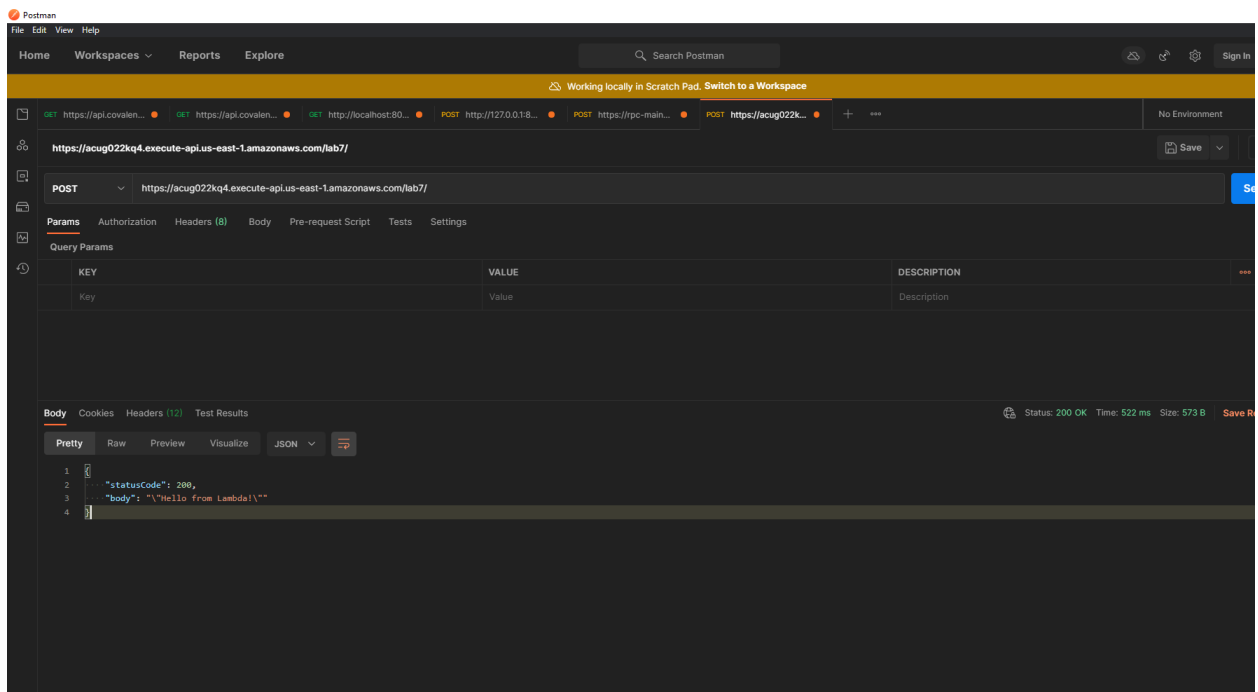
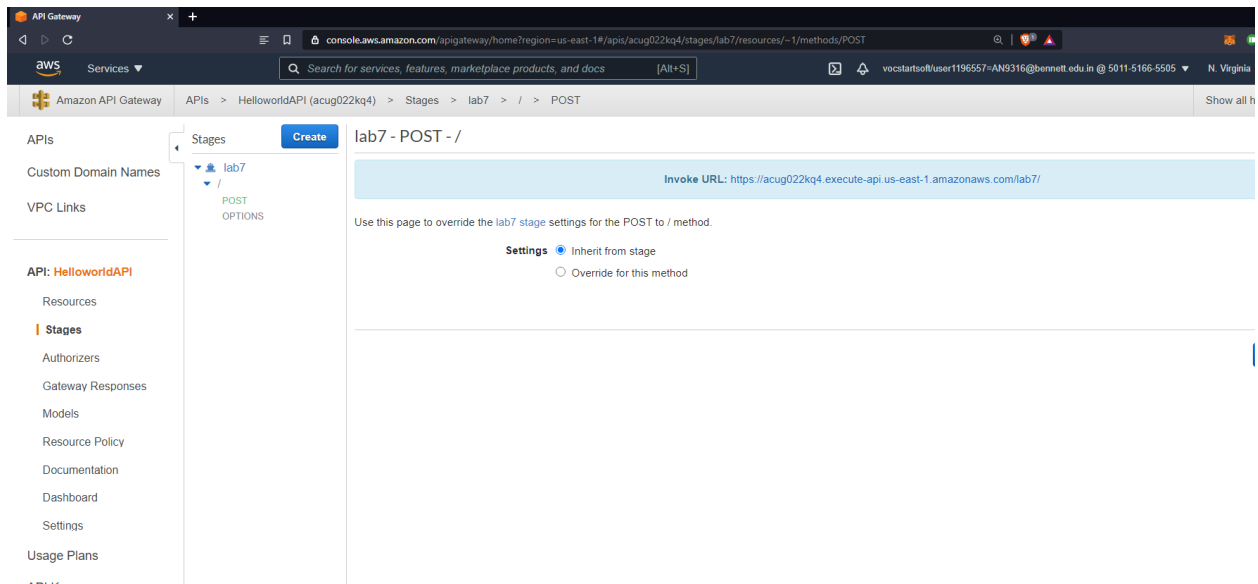
Hello World

## Task 2 – create lambda function, create test event.



## Task 3 – Create API, Create Routes, Enable CORS, Deploy API, Check API in Postman





## Task 4 – Create DynamoDB

The screenshot shows the AWS DynamoDB console interface. On the left, there's a sidebar with a search bar and a list of tables, including 'HelloWorldDatabase'. The main area displays the 'HelloWorldDatabase' overview page. It includes tabs for Overview, Items, Metrics, Alarms, Capacity, Indexes, Global Tables, Backups, Contributor Insights, Triggers, and Access control. The 'Overview' tab is active, showing 'Kinesis data stream details' and 'DynamoDB stream details'. The 'Table details' section lists various attributes for the 'HelloWorldDatabase' table.

Table name	HelloWorldDatabase
Primary partition key	ID (String)
Primary sort key	-
Point-in-time recovery	DISABLED <a href="#">Enable</a>
Encryption Type	DEFAULT <a href="#">Manage Encryption</a>
KMS Master Key ARN	Not Applicable
Encryption Status	-
CloudWatch Contributor Insights	DISABLED <a href="#">Manage Contributor Insights</a> <b>NEW</b>
Time to live attribute	DISABLED <a href="#">Manage TTL</a>
Table status	Active
Creation date	February 25, 2021 at 5:45:29 PM UTC+5:30
Read/write capacity mode	Provisioned
Last change to on-demand mode	-
Provisioned read capacity units	5 (Auto Scaling Error)
Provisioned write capacity units	5 (Auto Scaling Error)
Last decrease time	-
Last increase time	-
Storage size (in bytes)	0 bytes
Item count	0 <a href="#">Manage live count</a>
Region	US East (N. Virginia)
Amazon Resource Name (ARN)	arn:aws:dynamodb:us-east-1:501151665505:table/HelloWorldDatabase

## Task 5 – Update Lambda Permission and add policy.

The screenshot shows the AWS Lambda console interface for the 'helloworld' function. The 'Permissions' tab is active, displaying the 'Resource-based policy' for the function. The policy is a JSON document that grants permissions to the function.

```
1 {
2   "Version": "2012-10-17",
3   "Id": "default",
4   "Statement": [
5     {
6       "Sid": "c719943b-eea5-4204-833b-4889bdbfa2a5",
7       "Effect": "Allow",
8       "Principal": {
9         "Service": "apigateway.amazonaws.com"
10      },
11      "Action": "lambda:InvokeFunction",
12      "Resource": "arn:aws:lambda:us-east-1:501151665505:function:helloworld",
13      "Condition": {
14        "ArnLike": {
15          "AWS:SourceArn": "arn:aws:execute-api:us-east-1:501151665505:acug022kq4/*/*POST/*"
16        }
17      }
18    }
19  ]
20 }
```

## Task 6 – Save and Deploy new Lambda, Test Lamda.

The screenshot shows the AWS Lambda console interface for a function named 'helloworld'. The 'Function code' tab is selected, displaying a Python script. The script imports the 'json' module and the 'AWS SDK (for Python)' package 'boto3'. It creates a 'DynamoDB' object using the 'AWS SDK', uses the 'DynamoDB' object to select a table named 'HelloWorldDatabase', and stores the current time in a human-readable format in a variable. The script then defines a handler function that takes an event and context as input, extracts values from the event object, and returns a properly formatted JSON object. The function is currently in a 'Changes deployed' state.

```
1 # Import the json utility package since we will be working with a JSON object
2 import json
3 # Import the AWS SDK (for Python the package name is boto3)
4 import boto3
5 # Import two packages to help us with dates and date formatting
6 from time import strftime, localtime
7
8 # create a DynamoDB object using the AWS SDK
9 dynamodb = boto3.resource('dynamodb')
10 # use the DynamoDB object to select our table
11 table = dynamodb.Table('HelloWorldDatabase')
12 # store the current time in a human readable format in a variable
13 now = strftime("%a, %d %b %Y %H:%M:%S +0000", localtime())
14
15 # define the handler function that the Lambda service will use as an entry point
16 def lambda_handler(event, context):
17     # extract values from the event object we got from the Lambda service and store in a variable
18     name = event['firstName'] + ' ' + event['lastName']
19     # write name and time to the DynamoDB table using the object we instantiated and save response in a variable
20     response = table.put_item(
21         Item={
22             'ID': name,
23             'LatestGreetingTime': now
24         })
25     # return a properly formatted JSON object
26     return {
27         'statusCode': 200,
28         'body': json.dumps('Hello from Lambda, ' + name)
29     }
```

The screenshot shows the AWS Lambda console interface for the 'helloworld' function, with the 'Execution results' tab selected. The 'Response' section displays the output of the function: a JSON object with 'statusCode' 200 and 'body' containing a greeting. The 'Function Logs' section shows the execution details, including the request ID, start and end times, and the duration. The 'Request ID' is also displayed.

**Response**

```
{
  "statusCode": 200,
  "body": "\"Hello from Lambda, Anudit Nagar\""
}
```

**Function Logs**

```
START RequestId: c933ca8c-df8e-458f-89e7-d35246aac392 Version: $LATEST
END RequestId: c933ca8c-df8e-458f-89e7-d35246aac392
REPORT RequestId: c933ca8c-df8e-458f-89e7-d35246aac392 Duration: 203.49 ms Billed Duration: 204 ms Memory Size: 128 MB Max Memory Used: 75 MB
```

**Request ID**

```
c933ca8c-df8e-458f-89e7-d35246aac392
```

## Task 7 – Add new html file to S3 and test.

The screenshot shows a web browser with two tabs: 'DynamoDB - AWS Console' and 'S3 Management Console'. The active tab is 'Hello World', which displays a form with two input fields: 'First Name : Anudit' and 'Last Name : Nagar'. A 'Call API' button is located to the right of the 'Last Name' field. The browser's address bar shows the URL 'lab7-304.s3-website-us-east-1.amazonaws.com'.

## Task 8 – Check if value is in DynamoDB.

The screenshot shows the AWS DynamoDB console interface. The left sidebar contains a 'Create table' button and a 'Delete table' button. Below these is a search bar labeled 'Filter by table name' with a dropdown menu 'Choose a table ...' and an 'Actions' button. The main content area displays the 'HelloWorldDatabase' table. The 'Items' tab is selected, showing a list of items. The first item is 'Anudit Nagar' with a 'LatestGreetingTime' of 'Thu, 25 Feb 2021 12:25:11 +0000'. The table's primary key is 'ID'.

ID	LatestGreetingTime
Anudit Nagar	Thu, 25 Feb 2021 12:25:11 +0000