

# 1. Building a Serverless Web Application

**Objective:** Create a serverless web application using AWS Lambda, API Gateway, S3, and DynamoDB.

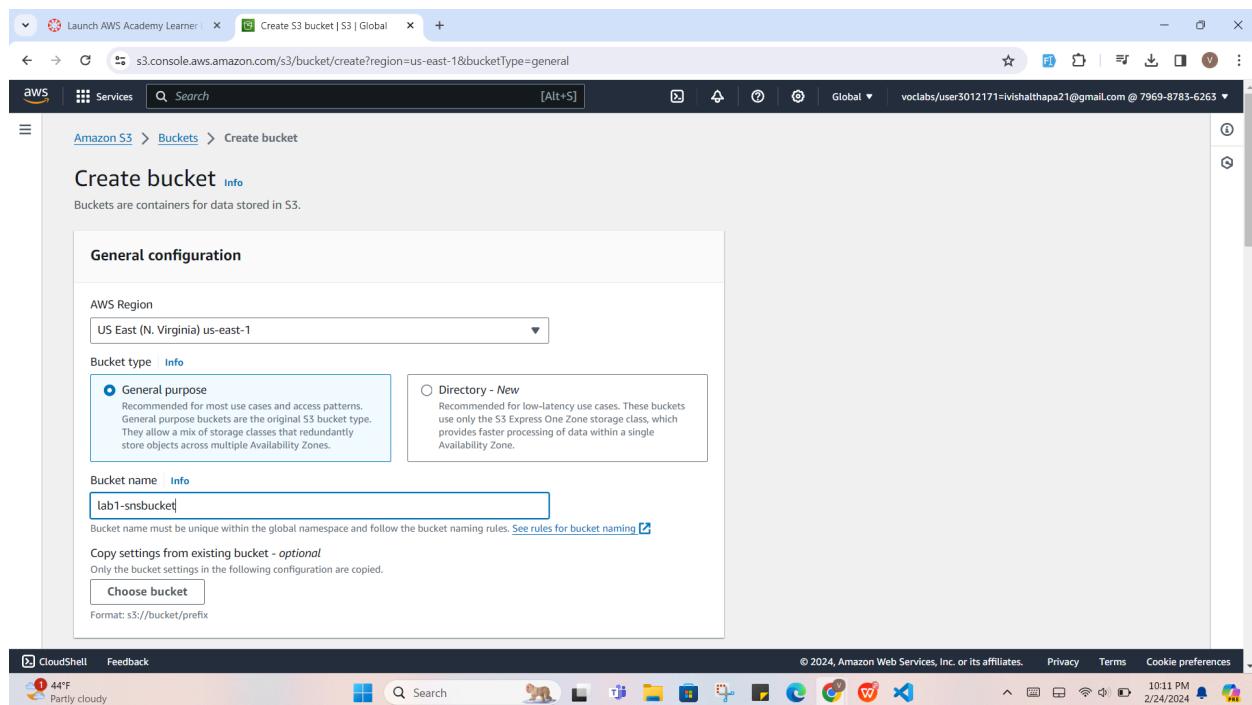
## Approach:

- **Set Up Backend:** Create Lambda functions to handle backend logic. These functions will interact with a DynamoDB table for data storage.
- **API Gateway:** Set up API Gateway to create RESTful endpoints that trigger the Lambda functions.
- **Frontend Hosting:** Host a static website on S3 that interacts with the backend via API Gateway.
- **Integration:** Ensure that the frontend can successfully send requests to the backend and display responses.

**Goal:** Understand the basics of building and connecting serverless backend services with a static frontend, enabling a fully serverless web application.

## SOLUTION:

1. Create a S3 bucket Since we need to upload a static webpage. After bucket creation, we need to edit the bucket policy generating the policy generator.



Launch AWS Academy Learner | Create S3 bucket | S3 | Global | +

s3.console.aws.amazon.com/s3/bucket/create?region=us-east-1&bucketType=general

aws Services Search [Alt+S]

**Block all public access**  
Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.

- Block public access to buckets and objects granted through new access control lists (ACLs)**  
S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.
- Block public access to buckets and objects granted through any access control lists (ACLs)**  
S3 will ignore all ACLs that grant public access to buckets and objects.
- Block public access to buckets and objects granted through new public bucket or access point policies**  
S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.
- Block public and cross-account access to buckets and objects through any public bucket or access point policies**  
S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

**Turning off block all public access might result in this bucket and the objects within becoming public**  
AWS recommends that you turn on block all public access, unless public access is required for specific and verified use cases such as static website hosting.

I acknowledge that the current settings might result in this bucket and the objects within becoming public.

**Bucket Versioning**  
Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences 44°F Partly cloudy 10:11 PM 2/24/2024

Launch AWS Academy Learner | Upload objects - S3 bucket lab | +

s3.console.aws.amazon.com/s3/upload/lab1-snsbucket?region=us-east-1&bucketType=general

aws Services Search [Alt+S]

Amazon S3 > Buckets > lab1-snsbucket > Upload

**Upload** Info

Add the files and folders you want to upload to S3. To upload a file larger than 160GB, use the AWS CLI, AWS SDK or Amazon S3 REST API. [Learn more](#)

Drag and drop files and folders you want to upload here, or choose Add files or Add folder.

**Files and folders (1 Total, 2.0 KB)**

	Name	Folder	Type
<input type="checkbox"/>	index.html	-	text/html

**Destination** Info

Destination  
`s3://lab1-snsbucket`

**Destination details**

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences 44°F Partly cloudy 10:15 PM 2/24/2024

Launch AWS Academy Learner | Edit static website hosting - S3

s3.console.aws.amazon.com/s3/bucket/lab1-snsbucket/property/website/edit?region=us-east-1&bucketType=general

Amazon S3 > Buckets > lab1-snsbucket > Edit static website hosting

## Edit static website hosting Info

**Static website hosting**  
Use this bucket to host a website or redirect requests. [Learn more](#)

**Static website hosting**  
 Disable  
 Enable

**Hosting type**  
 Host a static website  
Use the bucket endpoint as the web address. [Learn more](#)  
 Redirect requests for an object  
Redirect requests to another bucket or domain. [Learn more](#)

**Index document**  
Specify the home or default page of the website.  
index.html

**Error document - optional**  
This is returned when an error occurs.  
error.html

**Redirection rules - optional**  
Redirection rules, written in JSON, automatically redirect webpage requests for specific content. [Learn more](#)

CloudShell Feedback 44°F Partly cloudy © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences 10:16 PM 2/24/2024

Launch AWS Academy Learner | Edit static website hosting - S3

s3.console.aws.amazon.com/s3/bucket/lab1-snsbucket/property/website/edit?region=us-east-1&bucketType=general

Amazon S3 > Buckets > lab1-snsbucket > Edit static website hosting

## Edit static website hosting Info

**Static website hosting**  
For your customers to access content at the website endpoint, you must make all your content publicly readable. To do so, you can edit the S3 Block Public Access settings for the bucket. For more information, see [Using Amazon S3 Block Public Access](#)

**Index document**  
Specify the home or default page of the website.  
index.html

**Error document - optional**  
This is returned when an error occurs.  
error.html

**Redirection rules - optional**  
Redirection rules, written in JSON, automatically redirect webpage requests for specific content. [Learn more](#)

CloudShell Feedback 44°F Partly cloudy © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences 10:16 PM 2/24/2024

Launch AWS Academy Learner | lab1-snsbucket - S3 bucket | s3.console.aws.amazon.com

Successfully edited static website hosting.

Object Lock  
Disabled

**Requester pays**  
When enabled, the requester pays for requests and data transfer costs, and anonymous access to this bucket is disabled. [Learn more](#)

Requester pays  
Disabled

**Static website hosting**  
Use this bucket to host a website or redirect requests. [Learn more](#)

Static website hosting  
Enabled  
Hosting type  
Bucket hosting  
Bucket website endpoint  
When you configure your bucket as a static website, the website is available at the AWS Region-specific website endpoint of the bucket. [Learn more](#)

<http://lab1-snsbucket.s3-website-us-east-1.amazonaws.com>

CloudShell Feedback

44°F Partly cloudy

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

10:16 PM 2/24/2024

This screenshot shows the AWS S3 console with the 'Static website hosting' section selected. It displays the status as 'Enabled', the hosting type as 'Bucket hosting', and the endpoint as 'http://lab1-snsbucket.s3-website-us-east-1.amazonaws.com'. A success message at the top indicates that static website hosting was successfully edited. The left sidebar includes options like Buckets, Access Grants, Access Points, Object Lambda Access Points, Multi-Region Access Points, Batch Operations, IAM Access Analyzer for S3, and Block Public Access settings for this account.

Launch AWS Academy Learner | Edit bucket policy - S3 bucket | s3.console.aws.amazon.com

AWS Policy Generator

The AWS Policy Generator is a tool that enables you to create policies that control access to Amazon Web Services (AWS) products and resources. For more information about creating policies, see [key concepts in Using AWS Identity and Access Management](#). Here are [sample policies](#).

**Step 1: Select Policy Type**  
A Policy is a container for permissions. The different types of policies you can create are an [IAM Policy](#), an [S3 Bucket Policy](#), an [SNS Topic Policy](#), a [VPC Endpoint Policy](#), and an [SQS Queue Policy](#).

Select Type of Policy: S3 Bucket Policy

**Step 2: Add Statement(s)**  
A statement is the formal description of a single permission. See [a description of elements](#) that you can use in statements.

Effect: Allow (radio button selected) Deny

Principal:

Use a comma to separate multiple values.

AWS Service: Amazon S3 (dropdown menu) All Services (\*)

Actions: 1 Action(s) Selected All Actions (\*)

Amazon Resource Name (ARN): arn:aws:s3:::lab1-snsbucket

ARN should follow the following format: arn:aws:s3:::\${BucketName}/\${KeyName}.  
Use a comma to separate multiple values.

Add Conditions (Optional)

Add Statement

**Step 3: Generate Policy**  
A policy is a document (written in the [Access Policy Language](#)) that acts as a container for one or more statements.

Add one or more statements above to generate a policy.

CloudShell Feedback

44°F Partly cloudy

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

10:18 PM 2/24/2024

This screenshot shows the AWS Policy Generator interface. It's on Step 1, 'Select Policy Type', with 'S3 Bucket Policy' selected. In Step 2, 'Add Statement(s)', there is one action selected: 'arn:aws:s3:::lab1-snsbucket'. The ARN field contains 'arn:aws:s3:::lab1-snsbucket'. The 'Add Statement' button is highlighted in yellow. Step 3, 'Generate Policy', is shown below with the instruction 'Add one or more statements above to generate a policy.' The browser toolbar at the bottom shows various icons and the date/time '10:18 PM 2/24/2024'.

Screenshot of the AWS Policy Generator interface. The URL is [awspolicygen.s3.amazonaws.com/policygen.html](https://awspolicygen.s3.amazonaws.com/policygen.html).

**AWS Service:** Amazon S3  
 All Services (\*)

**Actions:**   All Actions (\*)

**Amazon Resource Name (ARN):**   
ARN should follow the following format: arn:aws:s3:::\${BucketName}/\${KeyName}. Use a comma to separate multiple values.

**Add Conditions (Optional)**:

You added the following statements. Click the button below to Generate a policy.

Principal(s)	Effect	Action	Resource	Conditions
*	Allow	s3:GetObject	arn:aws:s3:::lab1-snsbucket	None

**Step 3: Generate Policy**  
A policy is a document (written in the [Access Policy Language](#)) that acts as a container for one or more statements.

This AWS Policy Generator is provided for informational purposes only, you are still responsible for your use of Amazon Web Services technologies and ensuring that your use is in compliance with all applicable terms and conditions. This AWS Policy Generator is provided as-is without warranty of any kind, whether express, implied, or statutory. This AWS Policy Generator does not modify the applicable terms and conditions governing your use of Amazon Web Services technologies.

©2010, Amazon Web Services LLC or its affiliates. All rights reserved.  
An [amazon.com](#) company

Screenshot of the AWS S3 Bucket Policy editor. The URL is [s3.console.aws.amazon.com/s3/bucket/lab1-snsbucket/property/policy/edit?region=us-east-1&bucketType=general](https://s3.console.aws.amazon.com/s3/bucket/lab1-snsbucket/property/policy/edit?region=us-east-1&bucketType=general).

**Amazon S3**

- Buckets
- Access Grants
- Access Points
- Object Lambda Access Points
- Multi-Region Access Points
- Batch Operations
- IAM Access Analyzer for S3
- Block Public Access settings for this account
- Storage Lens
- Dashboards
- Storage Lens groups
- AWS Organizations settings
- Feature spotlight ?
- AWS Marketplace for S3

**Bucket policy**

The bucket policy, written in JSON, provides access to the objects stored in the bucket. Bucket policies don't apply to objects owned by other accounts. [Learn more](#)

**Bucket ARN:** arn:aws:s3:::lab1-snsbucket

**Policy:**

```

1  {
2   "Id": "Policy1708792404052",
3   "Version": "2012-10-17",
4   "Statement": [
5     {
6       "Sid": "Stmt1708792398103",
7       "Action": [
8         "s3:GetObject"
9       ],
10      "Effect": "Allow",
11      "Resource": "arn:aws:s3:::lab1-snsbucket",
12      "Principal": "*"
13    }
14  ]
15 }
```

**Edit statement**

Select a statement  
Select an existing statement in the policy or add a new statement.

2. We go to dynamoDB and create a table in dynamoDB.

Dashboard | Amazon DynamoDB

us-east-1.console.aws.amazon.com/dynamodbv2/home?region=us-east-1#dashboard

DynamoDB Services Search [Alt+S]

N. Virginia vocabs/use3012171=vishalthapa21@gmail.com @ 7969-8783-6263

## Dashboard

**Alarms (0) Info** Manage in CloudWatch

Find alarms

Alarm name Status

No custom alarms

**DAX clusters (0) Info** View details

Find clusters

Cluster name Status

An error occurred when loading the cluster list

User: arn:aws:sts::796987836263:assumed-role/voclabs/user3012171=vishalthapa21@gmail.com is not authorized to perform: dax:DescribeClusters because no identity-based policy allows the dax:DescribeClusters action

Create cluster

**Create resources**

Create table

Amazon DynamoDB Accelerator (DAX) is a fully-managed, highly-available, in-memory caching service for DynamoDB. Learn more

Create DAX cluster

**What's new**

FEB 23 Amazon DynamoDB zero-ETL integration with Amazon Redshift in the US East...

CloudShell Feedback

53°F Cloudy

Search

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

8:20 PM 2/24/2024

This screenshot shows the Amazon DynamoDB Dashboard. The left sidebar has sections for Dashboard, Tables, Explore items, PartiQL editor, Backups, Exports to S3, Imports from S3, Integrations, Reserved capacity, and Settings. Under DAX, it lists Clusters, Subnet groups, Parameter groups, and Events. The main area shows 'Alarms (0)' and 'DAX clusters (0)'. A message indicates an unauthorized access attempt. On the right, there's a 'Create resources' section with 'Create table' and 'Create DAX cluster' buttons, and a 'What's new' section with a recent update about zero-ETL integration.

Create table | Amazon DynamoDB

us-east-1.console.aws.amazon.com/dynamodbv2/home?region=us-east-1#create-table

DynamoDB Services Search [Alt+S]

N. Virginia vocabs/use3012171=vishalthapa21@gmail.com @ 7969-8783-6263

## Create table

**Table details** Info

DynamoDB is a schemaless database that requires only a table name and a primary key when you create the table.

**Table name**

This will be used to identify your table.

serverless-dynamoDB-tableid

Between 3 and 255 characters, containing only letters, numbers, underscores (\_), hyphens (-), and periods (.)

**Partition key**

The partition key is part of the table's primary key. It is a hash value that is used to retrieve items from your table and allocate data across hosts for scalability and availability.

Enter the partition key name String

1 to 255 characters and case sensitive.

**Sort key - optional**

You can use a sort key as the second part of a table's primary key. The sort key allows you to sort or search among all items sharing the same partition key.

Enter the sort key name String

1 to 255 characters and case sensitive.

**Table settings**

CloudShell Feedback

53°F Cloudy

Search

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

8:21 PM 2/24/2024

This screenshot shows the 'Create table' wizard in the Amazon DynamoDB console. It includes fields for the table name ('serverless-dynamoDB-tableid'), partition key ('String'), and sort key ('String'). The 'Table settings' section is partially visible at the bottom. The status bar at the bottom right shows the date and time as 8:21 PM on 2/24/2024.

View table | Amazon DynamoDB

us-east-1.console.aws.amazon.com/dynamodbv2/home?region=us-east-1#table?name=serverless-dynamoDB-tableid

The serverless-dynamoDB-tableid table was created successfully.

DynamoDB

Tables

Explore items

PartiQL editor

Backups

Exports to S3

Imports from S3

Integrations

Reserved capacity

Settings

DAX

Clusters

Subnet groups

Parameter groups

Events

CloudShell Feedback

53°F Cloudy

Search

[Alt+S]

N. Virginia

voclabs/use3012171=vishalhapa21@gmail.com @ 7969-8783-6263

DynamoDB > Tables > serverless-dynamoDB-tableid

Tables (3)

Any tag key

Any tag value

Find tables by table name

abacustodo\_dev

healthcare\_dev

serverless-dynamoDB-tableid

serverless-dynamoDB-tableid

C Actions Explore table items

Overview Indexes Monitor Global tables Backups Exports and streams Add

Protect your DynamoDB table from accidental writes and deletes

When you turn on point-in-time recovery (PITR), DynamoDB backs up your table data automatically so that you can restore to any given second in the preceding 35 days.

Additional charges apply. Learn more

Edit PITR

General information Info

Partition key id (String)

Sort key -

Capacity mode Provisioned

Table status Active

Alarms No active alarms

Point-in-time recovery (PITR) Info Off

Additional info

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

8:22 PM 2/24/2024

Items | Amazon DynamoDB Main

us-east-1.console.aws.amazon.com/dynamodbv2/home?region=us-east-1#item-explorer?operation=SCAN&table=serverless-dynamoDB-tableid

DynamoDB

Tables

Explore items

PartiQL editor

Backups

Exports to S3

Imports from S3

Integrations

Reserved capacity

Settings

DAX

Clusters

Subnet groups

Parameter groups

Events

CloudShell Feedback

43°F Partly cloudy

Search

[Alt+S]

N. Virginia

voclabs/use3012171=vishalhapa21@gmail.com @ 7969-8783-6263

DynamoDB > Explore items > serverless-dynamoDB-tableid

Tables (3)

Any tag key

Any tag value

Find tables by table name

abacustodo\_dev

healthcare\_dev

serverless-dynamoDB-tableid

serverless-dynamoDB-tableid

Autopreview View table details

Scan or query items

Scan Query

Select a table or index Table - serverless-dynamoDB-tableid

Select attribute projection All attributes

Filters

Run Reset

Completed. Read capacity units consumed: 0.5

Items returned (0)

C Actions Create item

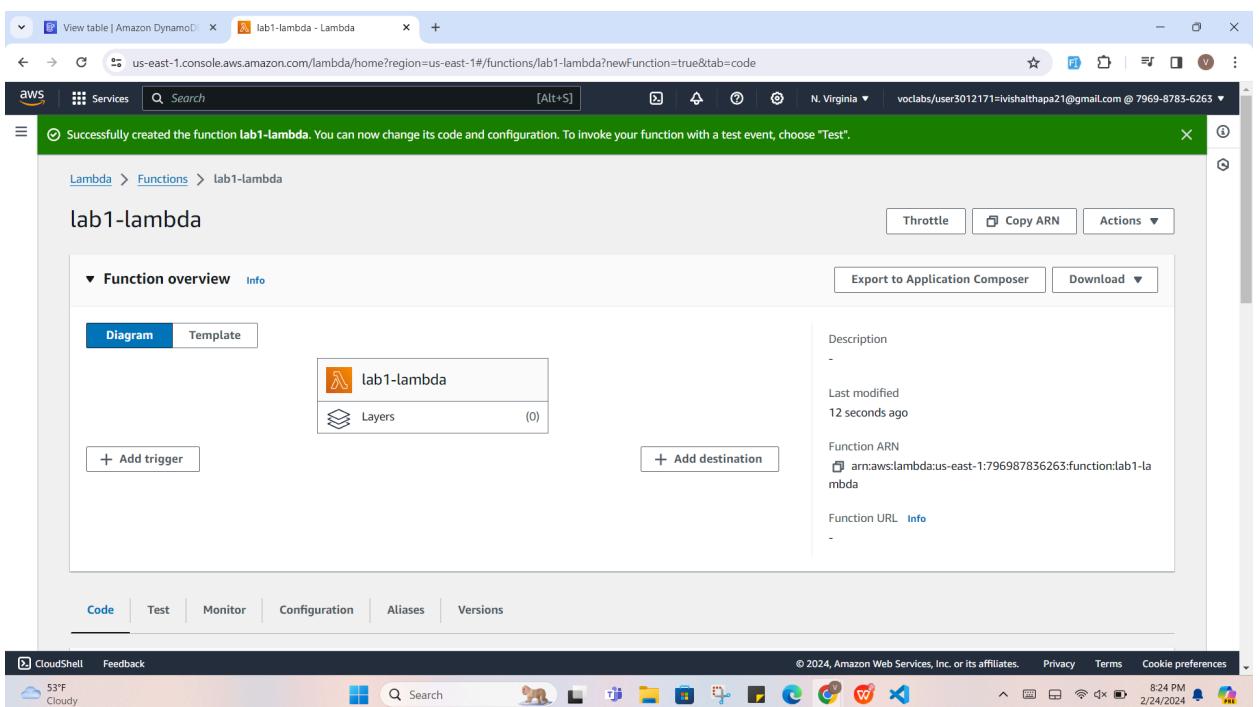
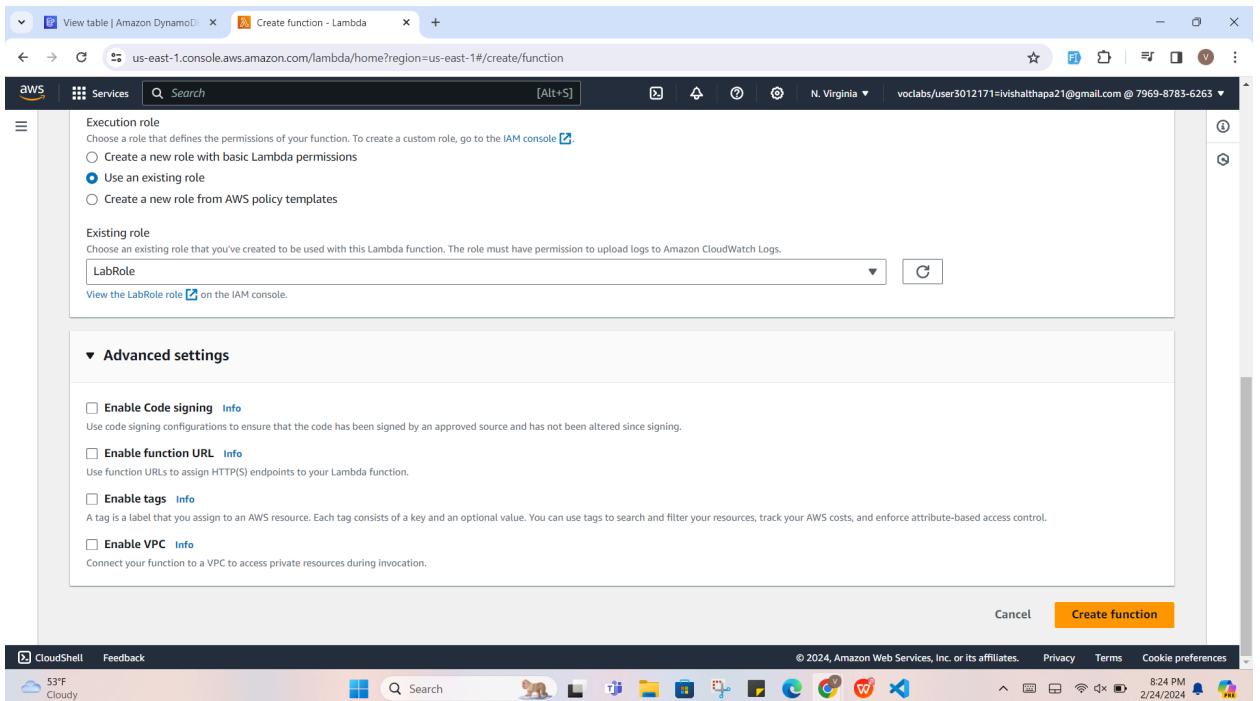
© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

10:49 PM 2/24/2024

### 3. Create Lambda function

The screenshot shows the 'Create function' wizard in the AWS Lambda console. The 'Basic information' step is selected. The 'Function name' field contains 'lab1-lambda'. The 'Runtime' dropdown is set to 'Python 3.12'. The 'Architecture' dropdown is set to 'x86\_64'. The 'Permissions' section is collapsed.

The screenshot shows the 'Create function' wizard in the AWS Lambda console. The 'Execution role' section is expanded. The 'Use an existing role' radio button is selected. A search bar shows the role ARN: 'c108787a256082815641115t1w79698783626-LambdaSLRRole-QyobkvY04YXY'. The 'Existing role' dropdown also lists 'LabRole' and 'RedshiftRole'. The 'Create function' button is visible at the bottom right.



```

1 import json
2 import botocore
3
4 def lambda_handler(event, context):
5     dynamodb = boto3.resource('dynamodb')
6     table = dynamodb.Table("replyback-table")
7     try:
8         http_method = event['httpMethod']
9         if http_method == "POST":
10             request_body = json.loads(event['body'])
11             full_name = request_body.get('fullname', '')
12             email = request_body.get('email', '')
13
14             item = {
15                 "full_name": full_name,
16                 "email": email
17             }
18             table.put_item(Item=item)
19             response = {
20                 "statusCode": 200,
21                 "body": json.dumps({"message": "Request successful"})
22             }
23         except Exception as e:
24             print(f"Error: {e}")
25             response = {
26                 "statusCode": 500,
27                 "body": json.dumps({"message": "Internal Server Error"})
28             }
29
30     return response

```

#### 4. And then create a REST API with a resource and method inside it.

The HTTP proxy integration, designated by `HTTP_PROXY` in the API Gateway REST API, is for integrating a method request with a backend HTTP endpoint. With this integration type, API Gateway simply passes the entire request and response between the frontend and the backend, subject to certain restrictions and limitations.

We can post the user input from user using this endpoint and update the dynamoDB table:  
`arn:aws:execute-api:us-east-1:796987836263:2sft2tz7ti/*/POST/message-endpoint`

The screenshot shows the AWS API Gateway - Create API page. At the top, there are tabs for 'View table | Amazon DynamoDB' and 'lab1-lambda - Lambda'. The main content area has a title 'REST API' with a sub-section 'REST API Private'. Both sections include descriptions, lists of compatible services (Lambda, HTTP, AWS Services), and 'Import' and 'Build' buttons. The status bar at the bottom indicates it's 8:26 PM on 2/24/2024.

The screenshot shows the AWS API Gateway - Create REST API page. The URL in the address bar is 'us-east-1.console.aws.amazon.com/apigateway/main/create-rest?experience=rest&region=us-east-1'. The page title is 'Create REST API'. It includes a 'CloudShell' and 'Feedback' button, a search bar, and a toolbar with various icons. The main form has sections for 'API details' (radio buttons for 'New API', 'Clone existing API', 'Import API', and 'Example API'), 'API name' (text input 'My REST API'), 'Description - optional' (text area), and 'API endpoint type' (dropdown set to 'Regional'). The status bar at the bottom indicates it's 8:26 PM on 2/24/2024.

The screenshot shows the 'Create REST API' page in the AWS Management Console. The 'API details' section contains four options:

- New API: Create a new REST API.
- Clone existing API: Create a copy of an API in this AWS account.
- Import API: Import an API from an OpenAPI definition.
- Example API: Learn about API Gateway with an example API.

The 'API name' field is filled with 'lab1-serverlessAPI'. The 'Description - optional' field is empty. The 'API endpoint type' dropdown is set to 'Regional'. The browser status bar at the bottom indicates the URL is `us-east-1.console.aws.amazon.com/apigateway/main/create-rest?experience=rest&region=us-east-1`.

The screenshot shows the 'Create resource' page in the AWS Management Console. A green success message at the top states 'Successfully created REST API 'lab1-serverlessAPI (2sft2tz7t)''. The 'Resource details' section includes:

- Proxy resource info: Describes proxy resources handling requests to all sub-resources.
- Resource path: '/'
- Resource name: 'message-endpoint'
- CORS (Cross Origin Resource Sharing) info: Describes creating an OPTIONS method for CORS.

At the bottom are 'Cancel' and 'Create resource' buttons. The browser status bar at the bottom indicates the URL is `us-east-1.console.aws.amazon.com/apigateway/main/apis/2sft2tz7t/resources/5qjz7u1868/create-resource?api=2sft2tz7t&experience=rest&region=us-east-1`.

Screenshot of the AWS API Gateway Resources page showing a successfully created resource '/message-endpoint'.

The 'Resource details' section shows:

- Path: /message-endpoint
- Resource ID: r4asdo

The 'Methods (1)' section shows:

Method type	Integration type	Authorization	API key
OPTIONS	Mock	None	Not required

The left sidebar shows the API structure:

- API: lab1-serverlessAPI
- Resources
  - /
  - /message-endpoint
- Stages
- Authorizers
- Gateway responses
- Models
- Resource policy
- Documentation
- Dashboard
- API settings

The bottom status bar indicates: © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences 8:32 PM 2/24/2024

Screenshot of the AWS API Gateway Create method page.

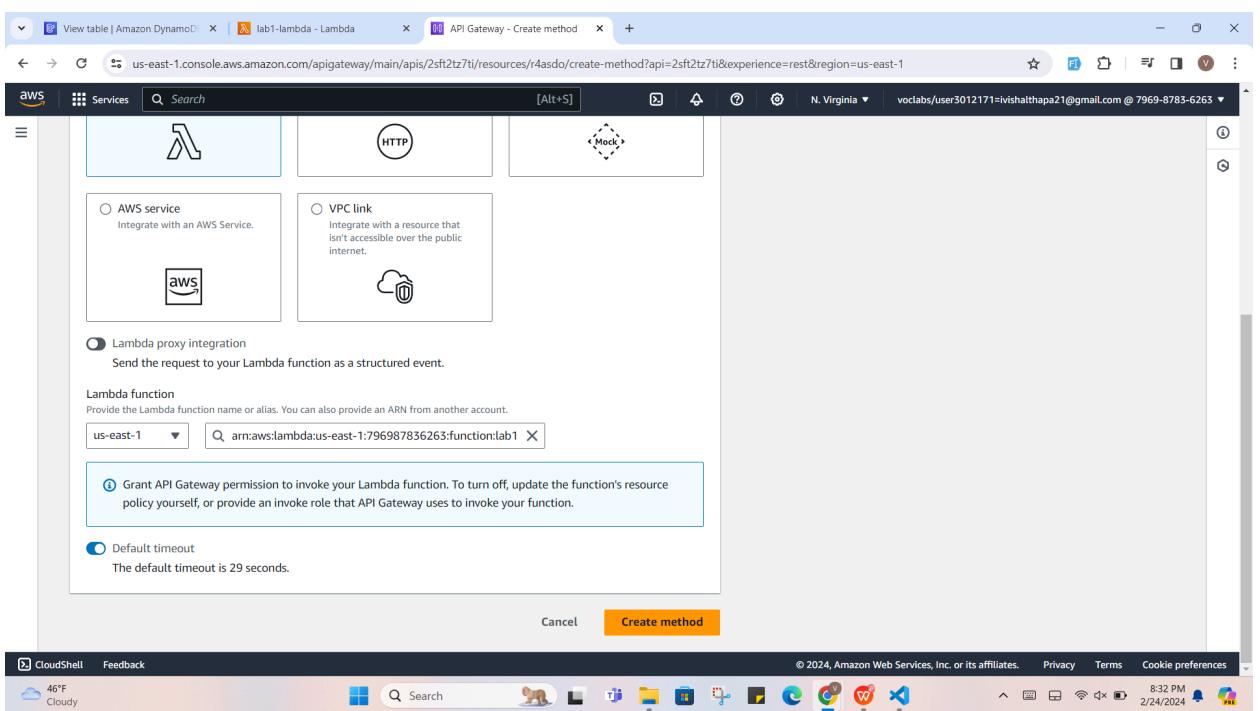
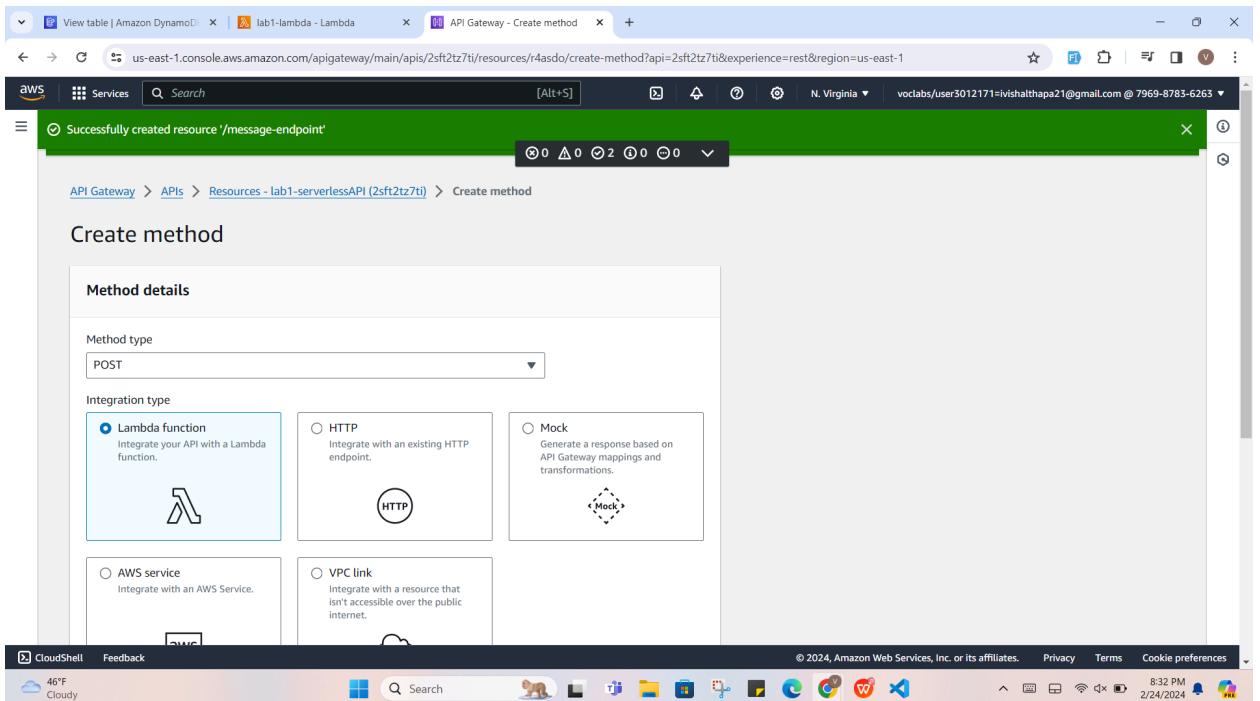
The 'Method details' section shows:

- Method type: ANY
- Integration type: Mock (selected)
- Authorization: None
- API key: Not required

The 'AWS service' and 'VPC link' options are shown as disabled.

The 'Lambda proxy integration' option is selected, with the note: "Send the request to your Lambda function as a structured event." A Lambda icon is displayed.

The bottom status bar indicates: © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences 8:32 PM 2/24/2024



Screenshot of the AWS API Gateway Resources page for the 'lab1-lambda - Lambda' API.

The left sidebar shows the API structure:

- APIs
- Custom domain names
- VPC links
- API: lab1-serverlessAPI**
  - Resources**
  - Stages
  - Authorizers
  - Gateway responses
  - Models
  - Resource policy
  - Documentation
  - Dashboard
  - API settings
- Usage plans
- API keys
- Client certificates
- Skills

The main panel displays the 'Resources' section for the 'lab1-serverlessAPI' (2sft2tz7ti). It shows a single resource path: /message-endpoint. Underneath it, two methods are listed: OPTIONS and POST. The POST method is currently selected.

Key details for the POST method:

- ARN:** arn:aws:execute-api:us-east-1:796987836263:2sft2tz7ti/\*/POST/message-endpoint
- Resource ID:** r4asdo

A flow diagram illustrates the request handling process:

```
graph LR; Client[Client] --> MethodRequest[Method request]; MethodRequest --> IntegrationRequest[Integration request]; IntegrationRequest --> LambdaIntegration[Lambda integration]; LambdaIntegration --> IntegrationResponse[Integration response]; IntegrationResponse --> MethodResponse[Method response]; MethodResponse --> Client
```

Below the flow diagram, tabs allow switching between Method request, Integration request, Integration response, Method response, and Test.

At the bottom right, there are 'Edit' and 'Deploy API' buttons.