2. Creating a Serverless API

Objective: Develop a serverless API using AWS Lambda and API Gateway.

Approach:

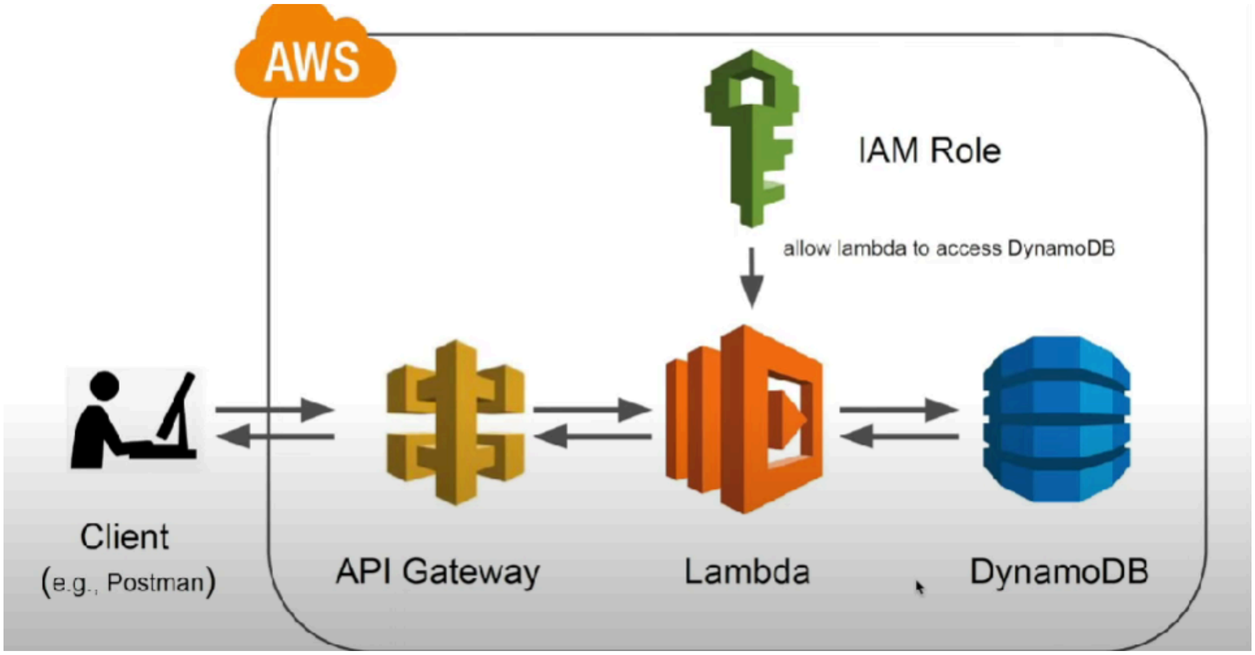
- **Define API**: Design a simple RESTful API (e.g., for a todo list application).
- **Lambda Functions**: Create Lambda functions for each API method (GET, POST, PUT, DELETE).
- **API Gateway Setup**: Use API Gateway to set up the API endpoints, connecting each endpoint to the corresponding Lambda function.
- **Testing**: Test the API using tools like Postman or AWS API Gateway test functionality.

Goal: Gain hands-on experience in building and deploying a serverless API, understanding the integration between Lambda and API Gateway.

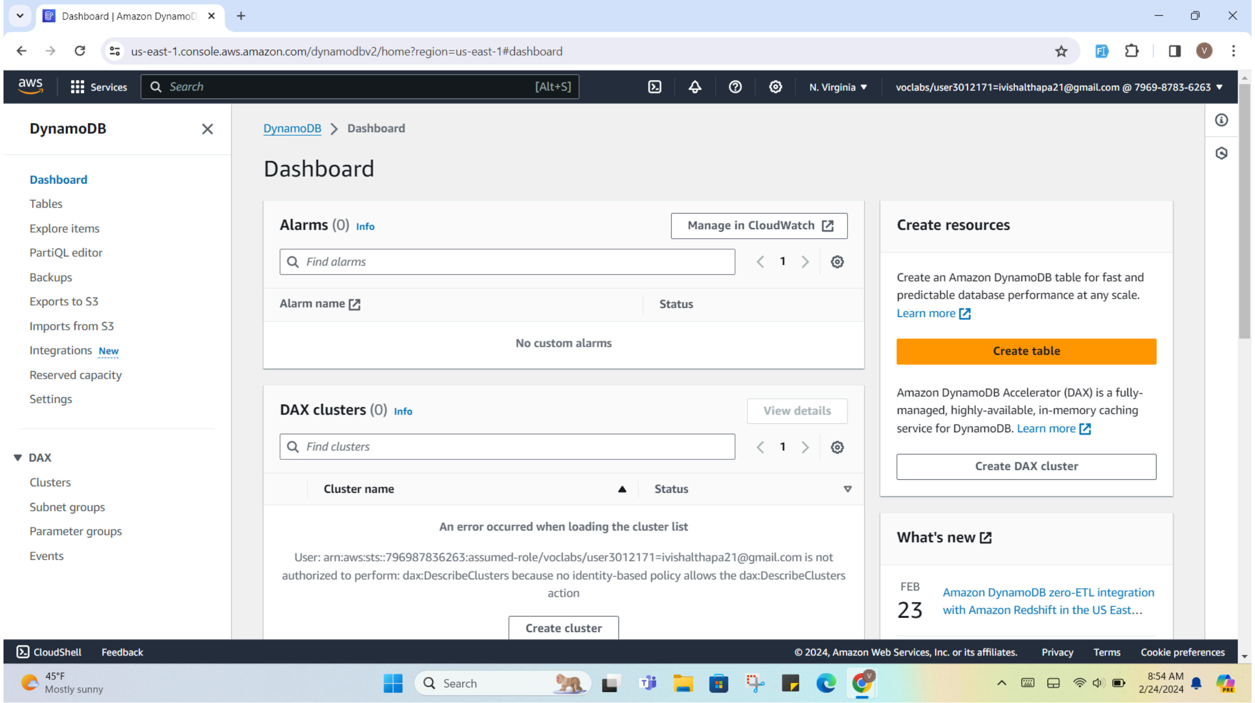
SOLUTION:

Serverless API Advantages:

1. No need to provision or manage the servers
2. Horizontal scaling
3. Never pay for idle
4. More reliable



1. First of all go to the searchbar and type dynamoDB to access the dynamoDB dashboard.



No custom alarms

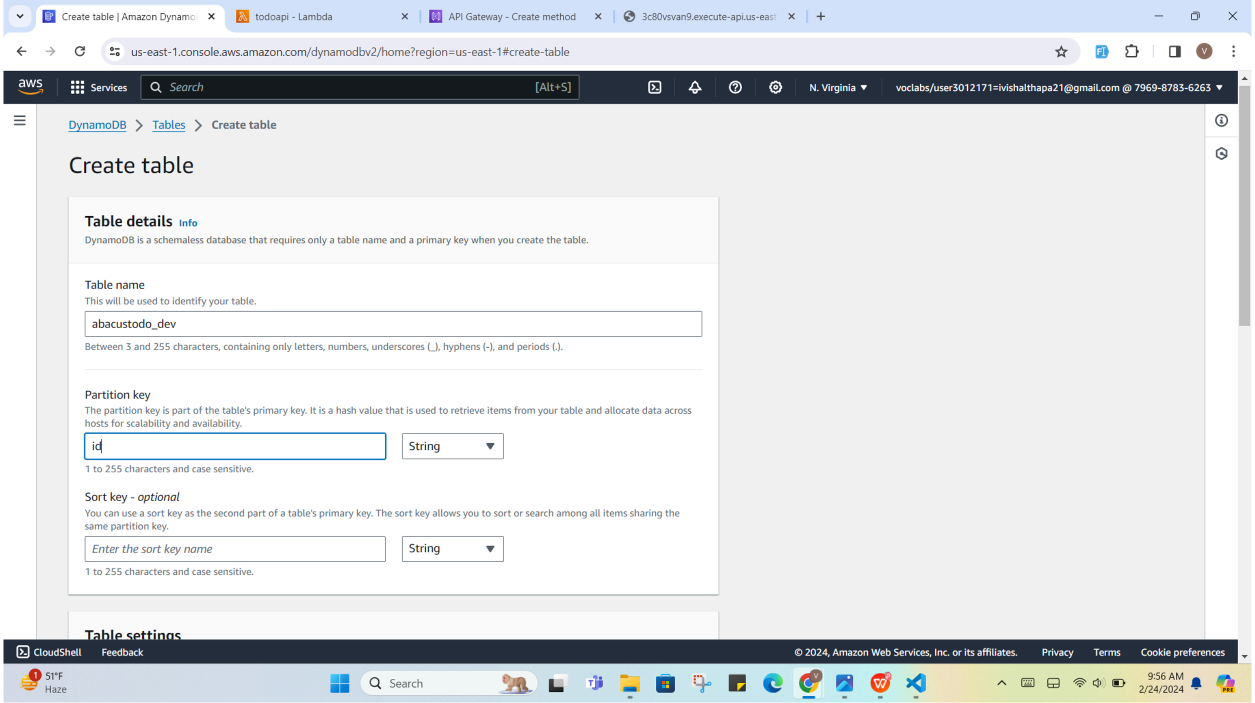
An error occurred when loading the cluster list

User: arn:aws:sts::796987836263:assumed-role/voclabs/user3012171=ivishalthapa21@gmail.com is not authorized to perform: dax:DescribeClusters because no identity-based policy allows the dax:DescribeClusters action

Create an Amazon DynamoDB table for fast and predictable database performance at any scale. Learn more

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

2. Click on the “create table” button to create a table in the dynamoDB using the table name and the partition key (id in our case).


Create table | Amazon Dynamo
todoapi - Lambda
API Gateway - Create method
3c80vsvan9.execute-api.us-east
us-east-1.console.aws.amazon.com/dynamodbv2/home?region=us-east-1#create-table
aws
Services
Search
[Alt+S]
N. Virginia ▼
voclabs/user3012171=ivishalthapa21@gmail.com @ 7969-8783-6263 ▼
DynamoDB > Tables > Create table
Create table
Table details Info
Table settings
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
51°F Haze
Search
9:56 AM 2/24/2024

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.

abacustodo_dev

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.

id String ▼

1 to 255 characters and case sensitive.

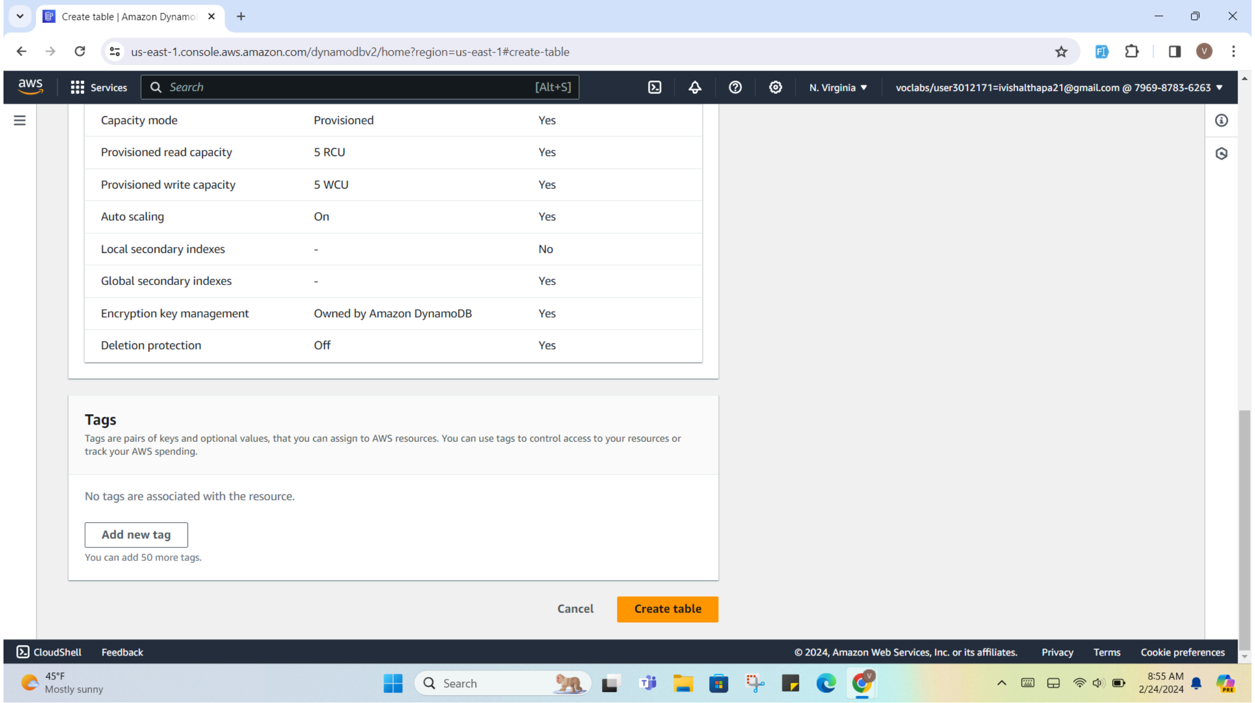
Sort key - *optional*

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.

3. Click on the “Create table” button and we just created a dynamoDB table that is just empty, we haven’t added anything here.



Create table | Amazon Dynamo
us-east-1.console.aws.amazon.com/dynamodbv2/home?region=us-east-1#create-table
aws
Services
Search
[Alt+S]
N. Virginia ▼
voclabs/user3012171=ivishalthapa21@gmail.com @ 7969-8783-6263 ▼
Cancel
Create table
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
45°F
Mostly sunny
Search
8:55 AM
2/24/2024

Capacity mode	Provisioned	Yes
Provisioned read capacity	5 RCU	Yes
Provisioned write capacity	5 WCU	Yes
Auto scaling	On	Yes
Local secondary indexes	-	No
Global secondary indexes	-	Yes
Encryption key management	Owned by Amazon DynamoDB	Yes
Deletion protection	Off	Yes

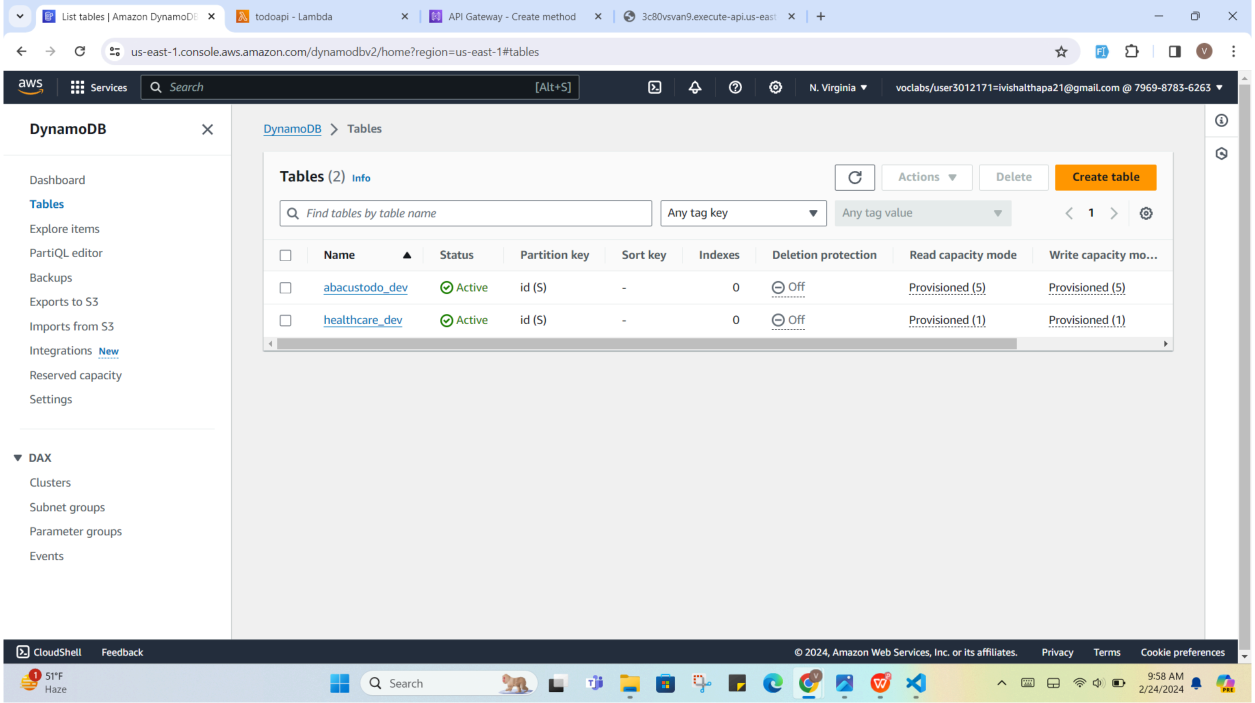
Tags

Tags are pairs of keys and optional values, that you can assign to AWS resources. You can use tags to control access to your resources or track your AWS spending.

No tags are associated with the resource.

Add new tag

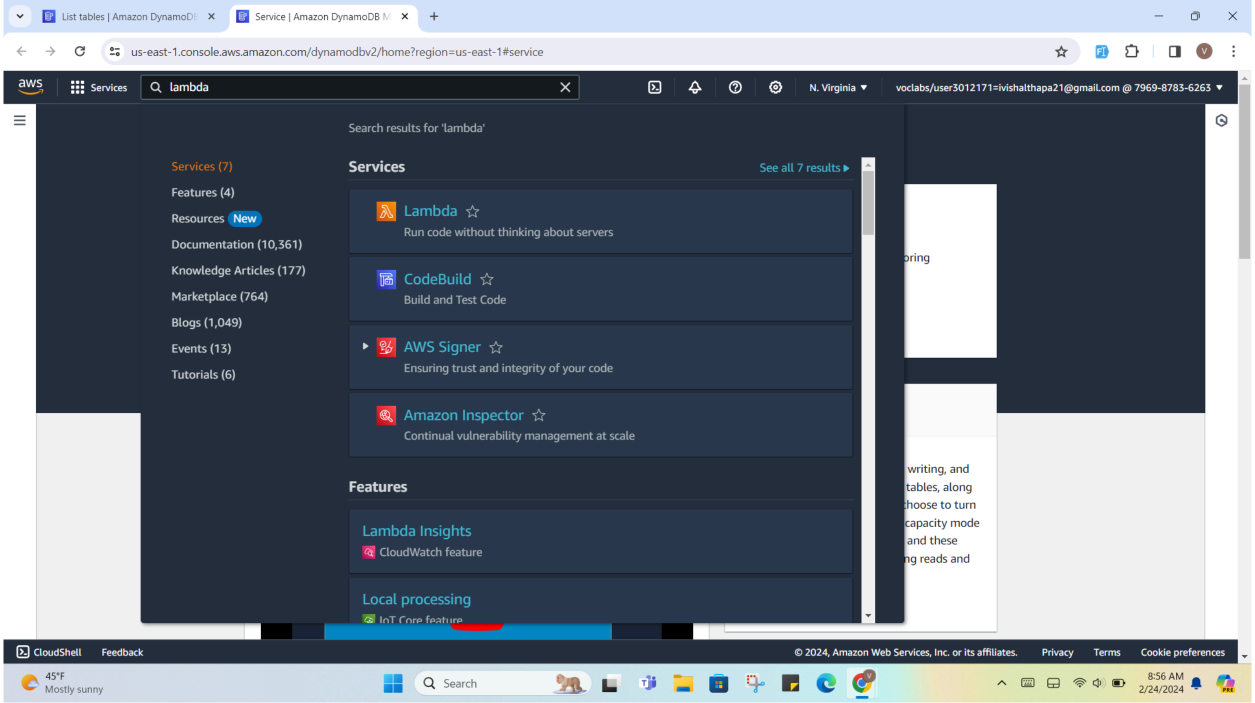
You can add 50 more tags.



Tables (2) Info

□	Name ▲	Status	Partition key	Sort key	Indexes	Deletion protection	Read capacity mode	Write capacity mo...
□	abacustodo_dev	✓ Active	id (S)	-	0	⊖ Off	Provisioned (5)	Provisioned (5)
□	healthcare_dev	✓ Active	id (S)	-	0	⊖ Off	Provisioned (1)	Provisioned (1)

4. Now we are going to create a lambda function, same process type dynamoDB from search bar where we need name, runtime details and we can use existing roles or create new. In our case we used an already existing jobRole. Click on the "create function" button to create the function.



List tables | Amazon DynamoDB | Service | Amazon DynamoDB

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

Services | lambda | N. Virginia ▼ | voclabs/user3012171=ivishalthapa21@gmail.com @ 7969-8783-6263 ▼

Search results for 'lambda'

Services (7)
Features (4)
Resources New
Documentation (10,361)
Knowledge Articles (177)
Marketplace (764)
Blogs (1,049)
Events (13)
Tutorials (6)

Services — See all 7 results ▸

Lambda ☆
Run code without thinking about servers

CodeBuild ☆
Build and Test Code

AWS Signer ☆
Ensuring trust and integrity of your code

Amazon Inspector ☆
Continual vulnerability management at scale

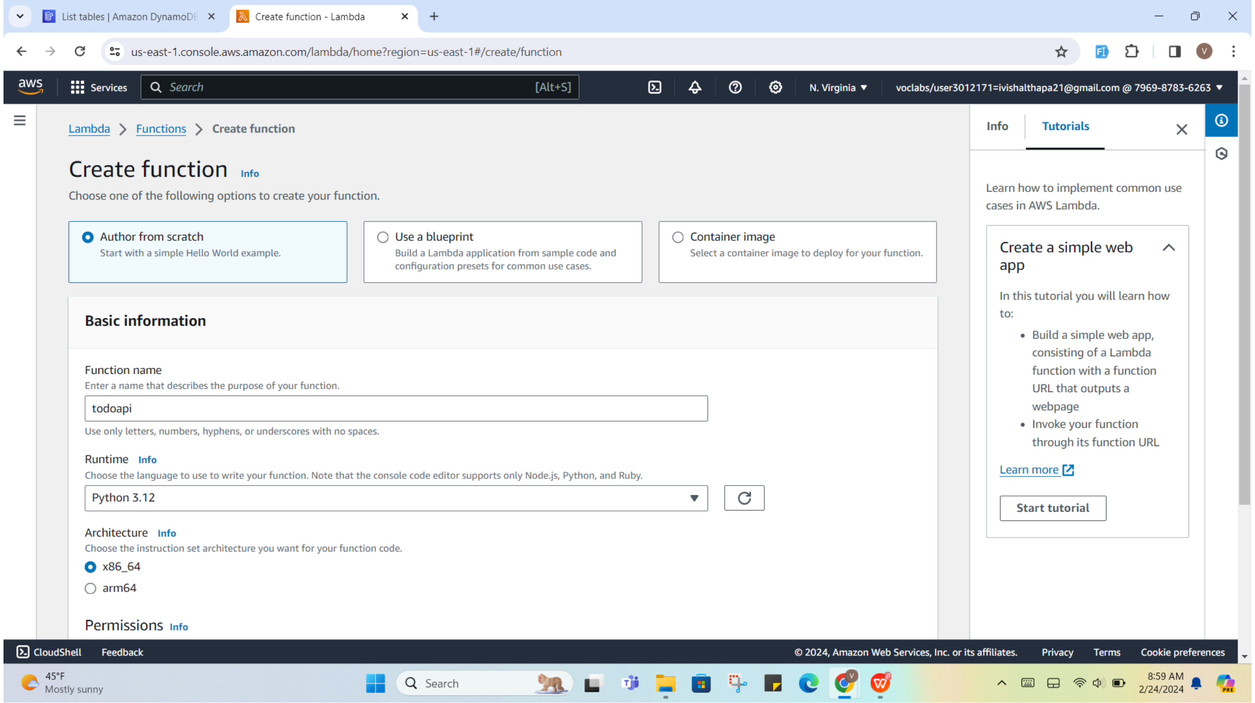
Features

Lambda Insights
CloudWatch feature

Local processing
IoT Core feature

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

45°F Mostly sunny | Search | 8:56 AM 2/24/2024



List tables | Amazon DynamoDB | Create function - Lambda

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

Services | Search [Alt+S] | N. Virginia ▼ | voclabs/user3012171=ivishalthapa21@gmail.com @ 7969-8783-6263 ▼

Lambda > Functions > Create function

Create function Info

Choose one of the following options to create your function.

● Author from scratch
Start with a simple Hello World example.

○ Use a blueprint
Build a Lambda application from sample code and configuration presets for common use cases.

○ Container image
Select a container image to deploy for your function.

Basic information

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

Runtime Info
Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.
Python 3.12

Architecture Info
Choose the instruction set architecture you want for your function code.
● x86_64
○ arm64

Permissions Info

Info | Tutorials

Learn how to implement common use cases in AWS Lambda.

Create a simple web app

In this tutorial you will learn how to:

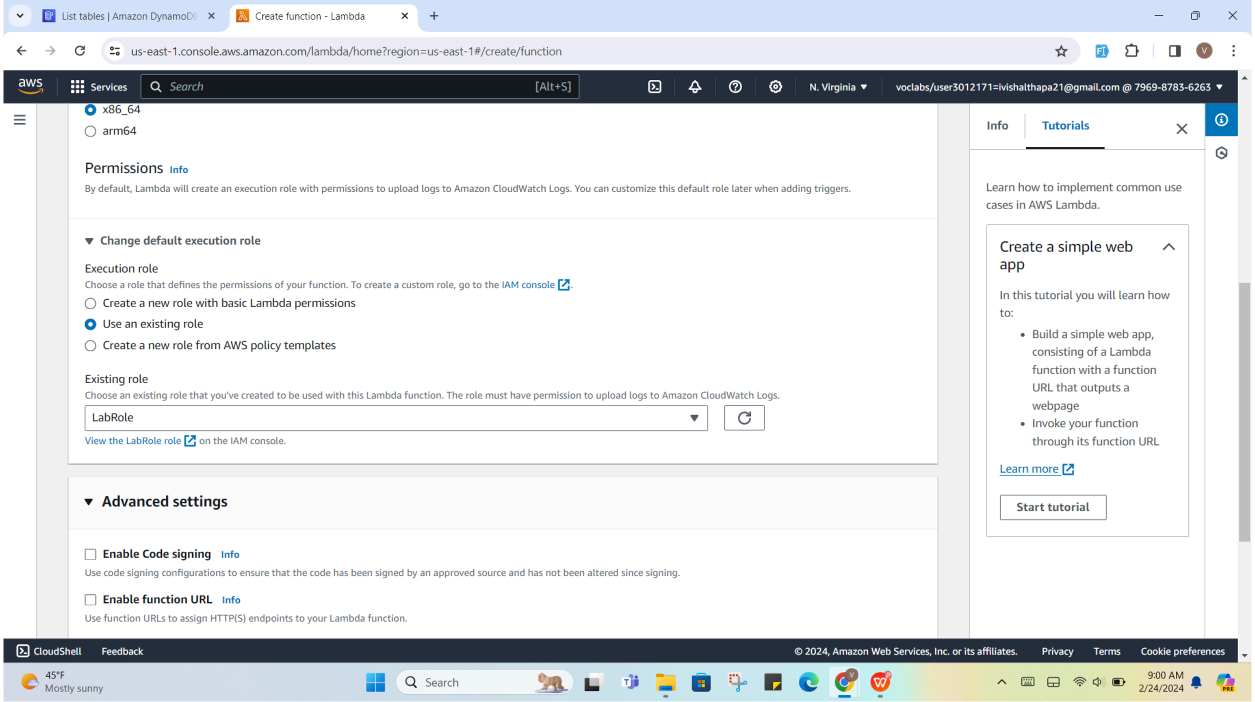
- Build a simple web app, consisting of a Lambda function with a function URL that outputs a webpage
- Invoke your function through its function URL

Learn more

Start tutorial

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

45°F Mostly sunny | Search | 8:59 AM 2/24/2024


List tables | Amazon DynamoDB
Create function - Lambda
us-east-1.console.aws.amazon.com/lambda/home?region=us-east-1#/create/function
aws
Services
Search
[Alt+S]
N. Virginia
voclabs/user3012171=ivishalthapa21@gmail.com @ 7969-8783-6263
Info
Tutorials
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
45°F
Mostly sunny
Search
9:00 AM
2/24/2024

● x86_64
○ arm64

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.

LabRole

View the LabRole role on the IAM console.

▼ Advanced settings

□ Enable Code signing Info

Use code signing configurations to ensure that the code has been signed by an approved source and has not been altered since signing.

□ Enable function URL Info

Use function URLs to assign HTTP(S) endpoints to your Lambda function.

Learn how to implement common use cases in AWS Lambda.

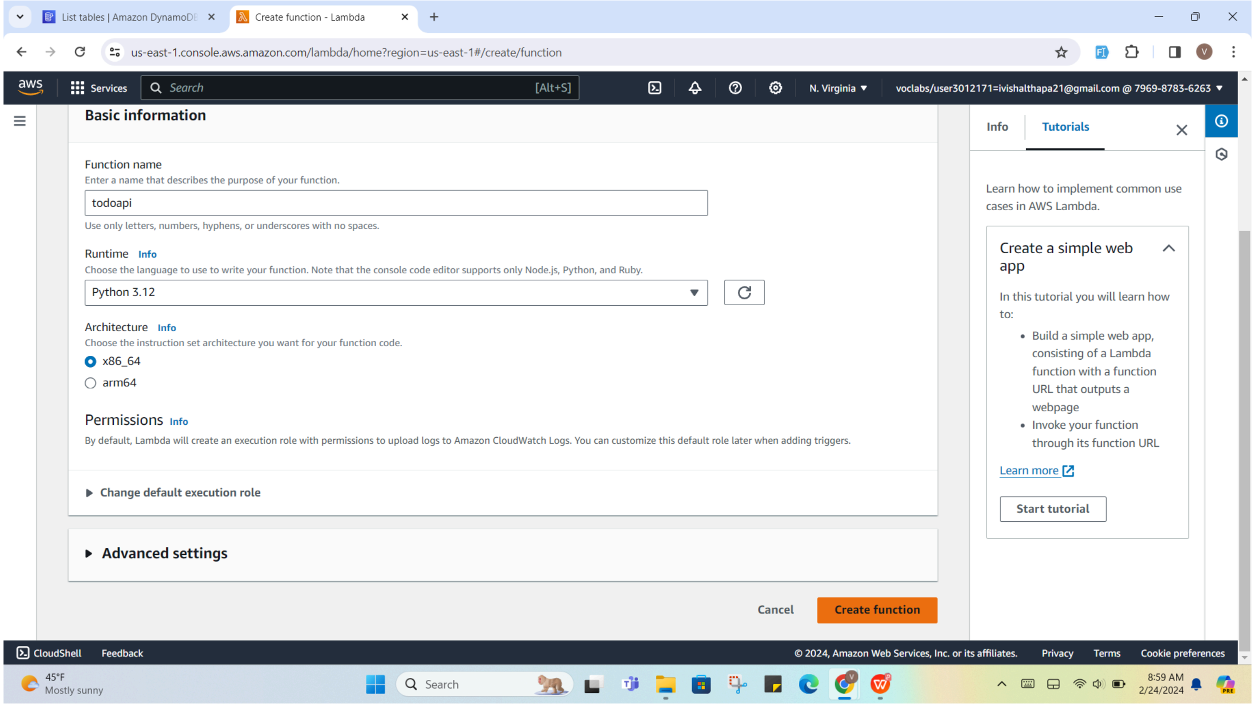
Create a simple web app

In this tutorial you will learn how to:

- Build a simple web app, consisting of a Lambda function with a function URL that outputs a webpage
- Invoke your function through its function URL

Learn more

Start tutorial


List tables | Amazon DynamoDB
Create function - Lambda
us-east-1.console.aws.amazon.com/lambda/home?region=us-east-1#/create/function
aws
Services
Search
[Alt+S]
N. Virginia
voclabs/user3012171=ivishalthapa21@gmail.com @ 7969-8783-6263
Info
Tutorials
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
45°F
Mostly sunny
Search
8:59 AM
2/24/2024

Basic information

Function name

Enter a name that describes the purpose of your function.

todoapi

Use only letters, numbers, hyphens, or underscores with no spaces.

Runtime Info

Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.

Python 3.12

Architecture Info

Choose the instruction set architecture you want for your function code.

● x86_64
○ arm64

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

▶ Advanced settings

Cancel Create function

Learn how to implement common use cases in AWS Lambda.

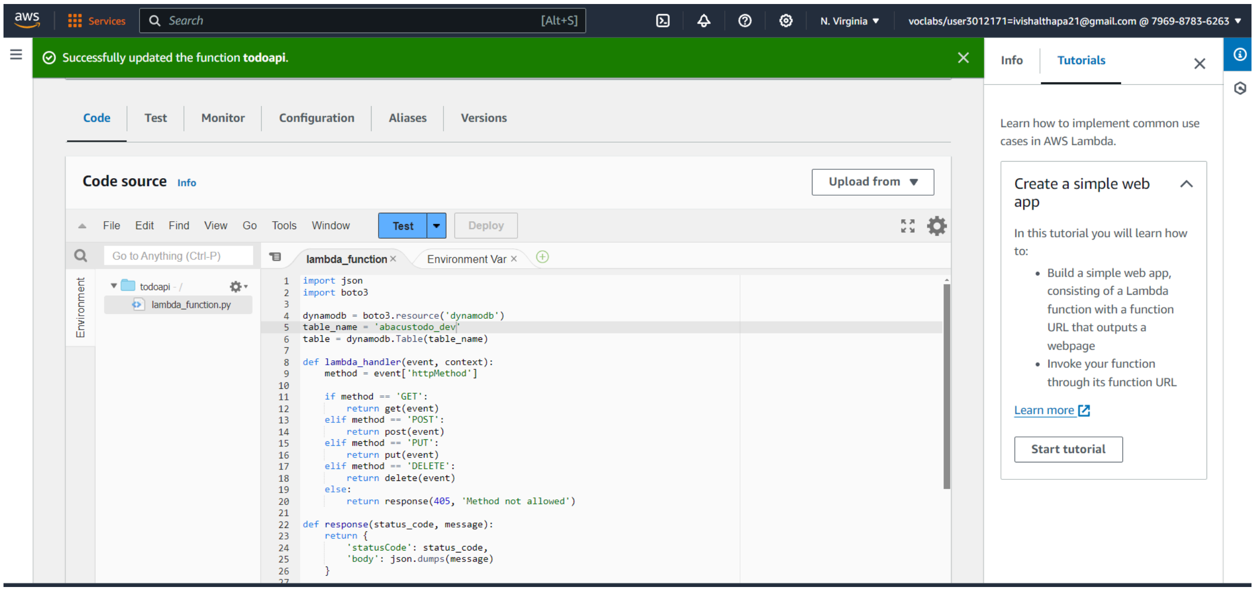
Create a simple web app

In this tutorial you will learn how to:

- Build a simple web app, consisting of a Lambda function with a function URL that outputs a webpage
- Invoke your function through its function URL

Learn more

Start tutorial



Successfully updated the function **todoapi**.

```python
import json
import boto3

dynamodb = boto3.resource('dynamodb')
table_name = 'abacustodo_dev'
table = dynamodb.Table(table_name)

def lambda_handler(event, context):
    method = event['httpMethod']

    if method == 'GET':
        return get(event)
    elif method == 'POST':
        return post(event)
    elif method == 'PUT':
        return put(event)
    elif method == 'DELETE':
        return delete(event)
    else:
        return response(405, 'Method not allowed')

def response(status_code, message):
    return {
        'statusCode': status_code,
        'body': json.dumps(message)
    }
```

Learn how to implement common use cases in AWS Lambda.

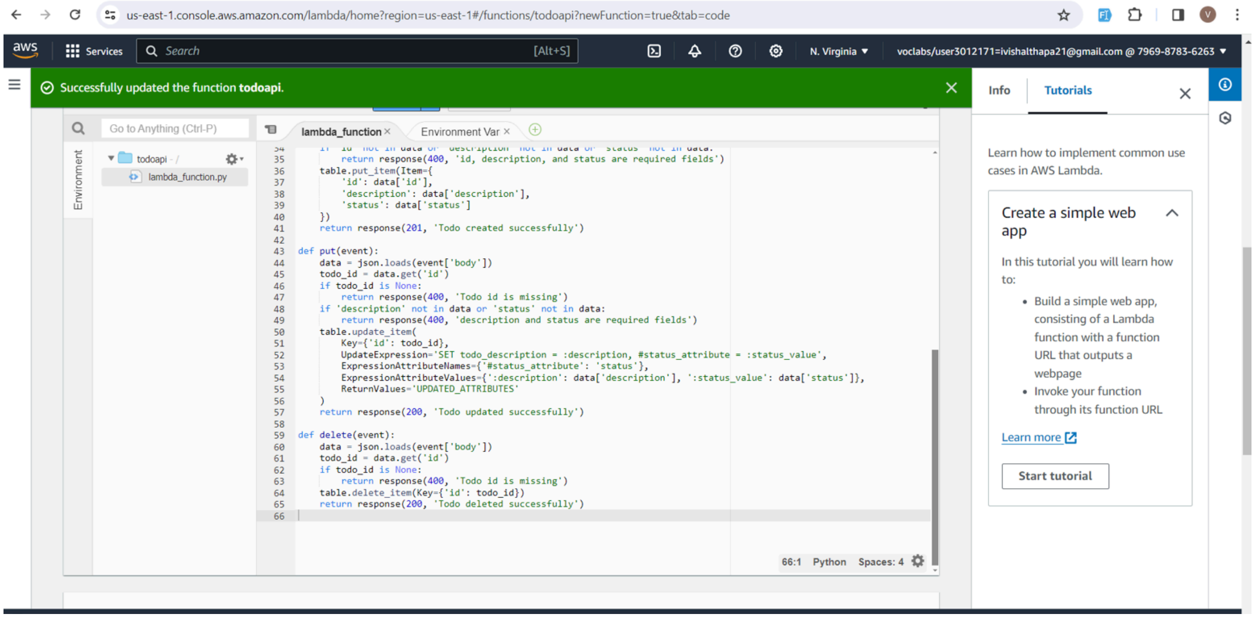
Create a simple web app

In this tutorial you will learn how to:

- Build a simple web app, consisting of a Lambda function with a function URL that outputs a webpage
- Invoke your function through its function URL

Learn more

Start tutorial


us-east-1.console.aws.amazon.com/lambda/home?region=us-east-1#/functions/todoapi?newFunction=true&tab=code

N. Virginia ▾ voclabs/user3012171=ivishalthapa21@gmail.com @ 7969-8783-6263 ▾

Successfully updated the function **todoapi**.

```
    if 'id' not in data or 'description' not in data or 'status' not in data:
        return response(400, 'id, description, and status are required fields')
    table.put_item(Item={
        'id': data['id'],
        'description': data['description'],
        'status': data['status']
    })
    return response(201, 'Todo created successfully')

def put(event):
    data = json.loads(event['body'])
    todo_id = data.get('id')
    if todo_id is None:
        return response(400, 'Todo id is missing')
    if 'description' not in data or 'status' not in data:
        return response(400, 'description and status are required fields')
    table.update_item(
        Key={'id': todo_id},
        UpdateExpression='SET todo_description = :description, #status_attribute = :status_value',
        ExpressionAttributeNames={'#status_attribute': 'status'},
        ExpressionAttributeValues={':description': data['description'], ':status_value': data['status']},
        ReturnValues='UPDATED_ATTRIBUTES'
    )
    return response(200, 'Todo updated successfully')

def delete(event):
    data = json.loads(event['body'])
    todo_id = data.get('id')
    if todo_id is None:
        return response(400, 'Todo id is missing')
    table.delete_item(Key={'id': todo_id})
    return response(200, 'Todo deleted successfully')
```

66:1 Python Spaces: 4

Learn how to implement common use cases in AWS Lambda.

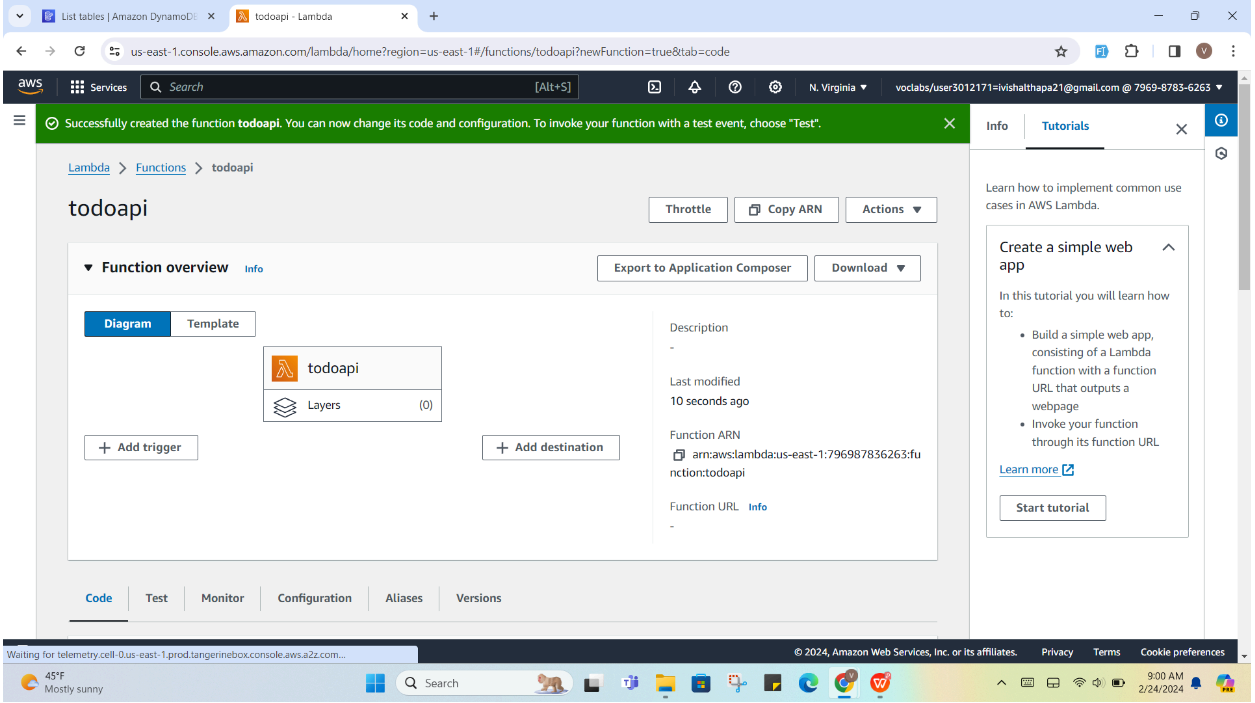
Create a simple web app

In this tutorial you will learn how to:

- Build a simple web app, consisting of a Lambda function with a function URL that outputs a webpage
- Invoke your function through its function URL

Learn more

5. Now we need to create an API gateway. We create a new REST API using name as todo api, endpoint as regional and create API. From resource on left nav , Now create resource from the left navbar with name todo and enable API gateway cors and create resource. Inside the todo resource, we will create all required methods for API endpoints. We need to deploy API as well


us-east-1.console.aws.amazon.com/lambda/home?region=us-east-1#/functions/todoapi?newFunction=true&tab=code

N. Virginia ▾ voclabs/user3012171=ivishalthapa21@gmail.com @ 7969-8783-6263 ▾

Successfully created the function **todoapi**. You can now change its code and configuration. To invoke your function with a test event, choose "Test".

Lambda > Functions > todoapi

todoapi

Function overview

Description
-

Last modified
10 seconds ago

Function ARN
arn:aws:lambda:us-east-1:796987836263:function:todoapi

Function URL
-

Learn how to implement common use cases in AWS Lambda.

Create a simple web app

In this tutorial you will learn how to:

- Build a simple web app, consisting of a Lambda function with a function URL that outputs a webpage
- Invoke your function through its function URL

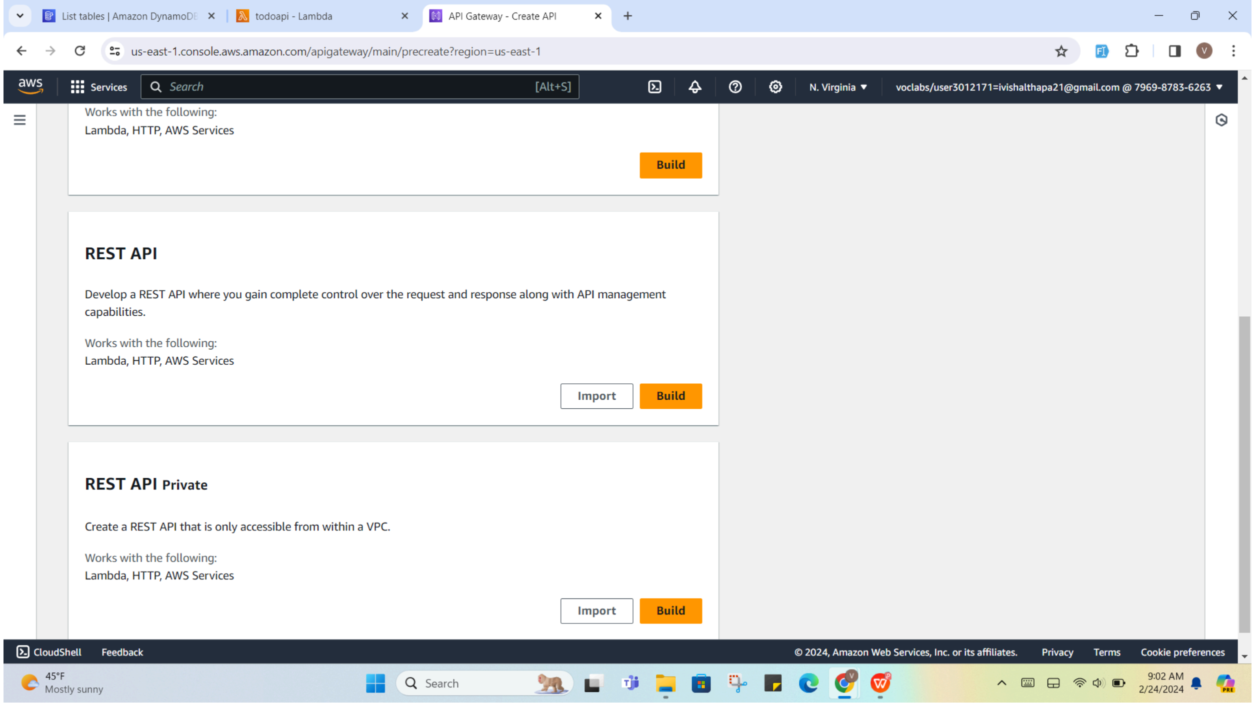
Learn more

Waiting for telemetry.cell-0.us-east-1.prod.tangerinebox.console.aws.a2z.com...

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

45°F Mostly sunny

9:00 AM 2/24/2024


List tables | Amazon DynamoD
todoapi - Lambda
API Gateway - Create API
us-east-1.console.aws.amazon.com/apigateway/main/precreate?region=us-east-1
Services
Search
[Alt+S]
N. Virginia
voclabs/user3012171=ivishalthapa21@gmail.com @ 7969-8783-6263

Works with the following:
Lambda, HTTP, AWS Services

REST API

Develop a REST API where you gain complete control over the request and response along with API management capabilities.

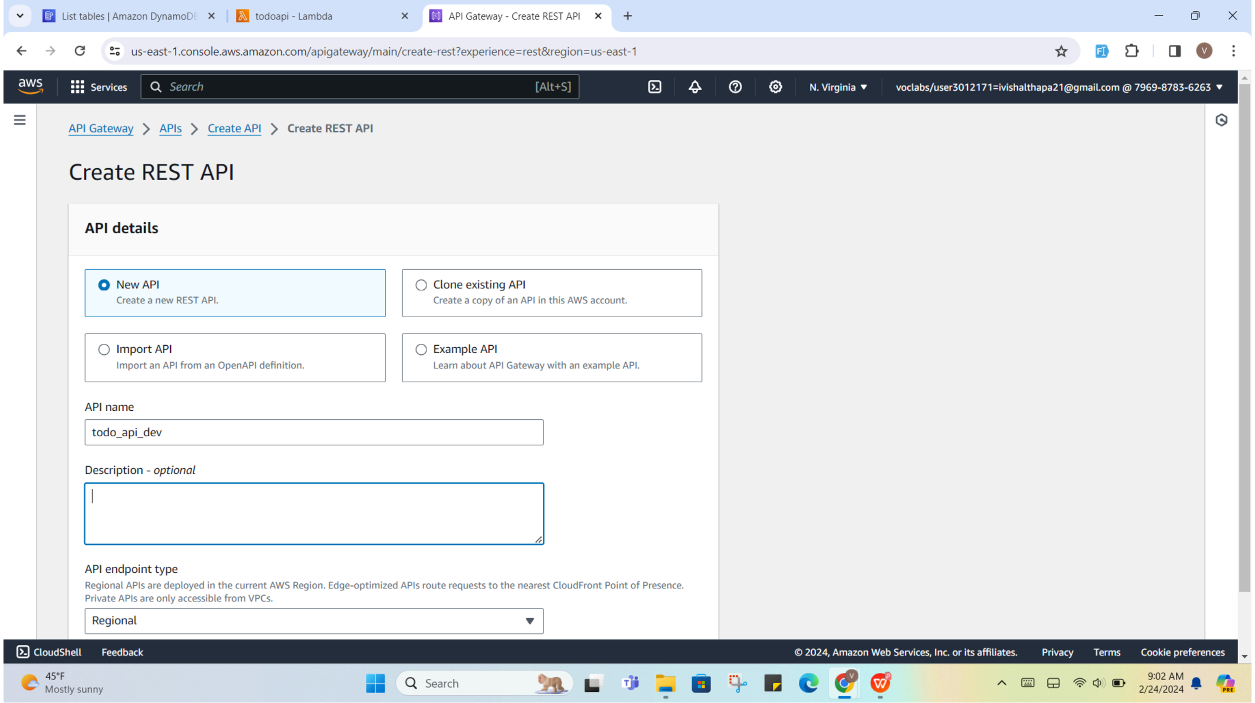
Works with the following:
Lambda, HTTP, AWS Services

REST API Private

Create a REST API that is only accessible from within a VPC.

Works with the following:
Lambda, HTTP, AWS Services

Import
Build
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
45°F
Mostly sunny
Search
9:02 AM
2/24/2024


List tables | Amazon DynamoD
todoapi - Lambda
API Gateway - Create REST API
us-east-1.console.aws.amazon.com/apigateway/main/create-rest?experience=rest&region=us-east-1
Services
Search
[Alt+S]
N. Virginia
voclabs/user3012171=ivishalthapa21@gmail.com @ 7969-8783-6263
API Gateway > APIs > Create API > Create REST API

Create REST API

API details

● 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.

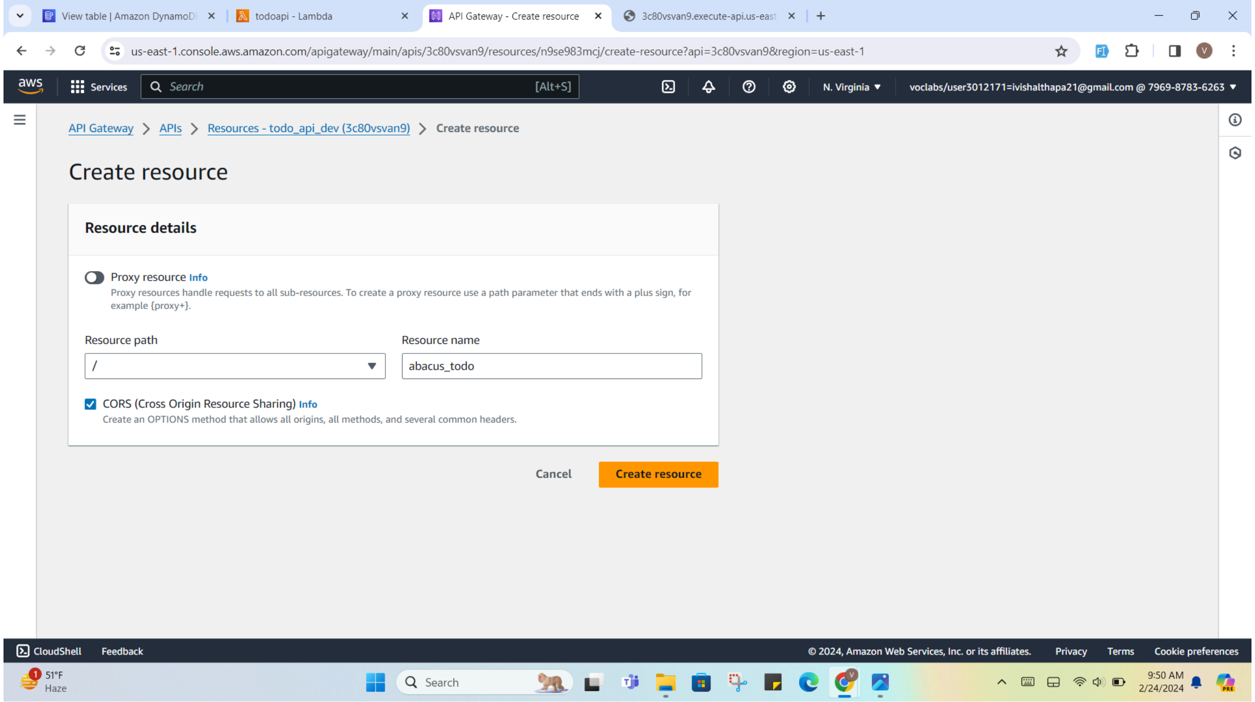
○ Example API
Learn about API Gateway with an example API.

API name
todo_api_dev

Description - *optional*

API endpoint type
Regional APIs are deployed in the current AWS Region. Edge-optimized APIs route requests to the nearest CloudFront Point of Presence. Private APIs are only accessible from VPCs.

Regional


View table | Amazon DynamoD
todoapi - Lambda
API Gateway - Create resource
3c80vsvan9.execute-api.us-east
us-east-1.console.aws.amazon.com/apigateway/main/apis/3c80vsvan9/resources/n9se983mcj/create-resource?api=3c80vsvan9&region=us-east-1
aws
Services
Search
[Alt+S]
N. Virginia
voclabs/user3012171=ivishalthapa21@gmail.com @ 7969-8783-6263
API Gateway > APIs > Resources - todo_api_dev (3c80vsvan9) > Create resource
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
51°F
Haze
Search
9:50 AM
2/24/2024

Create resource

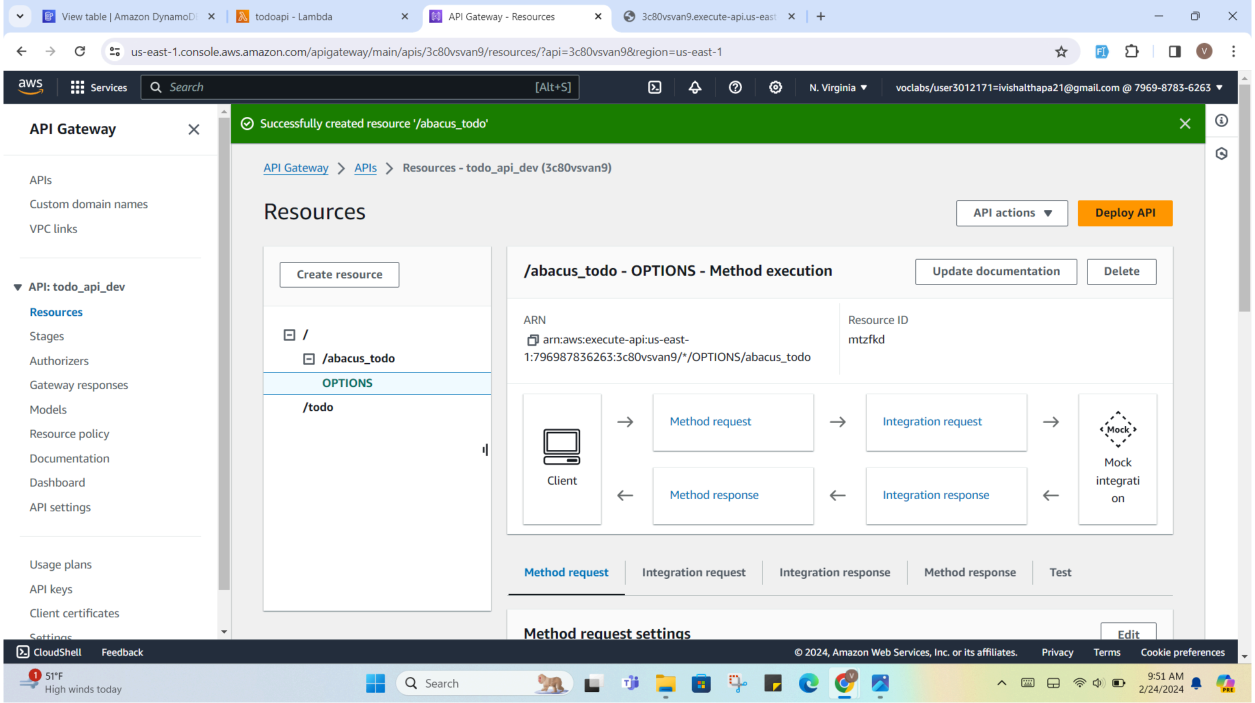
Resource details

○ Proxy resource Info
Proxy resources handle requests to all sub-resources. To create a proxy resource use a path parameter that ends with a plus sign, for example {proxy+}.

Resource path	Resource name
/	abacus_todo

☑ CORS (Cross Origin Resource Sharing) Info
Create an OPTIONS method that allows all origins, all methods, and several common headers.

Cancel | Create resource



Successfully created resource '/abacus_todo'

Resources

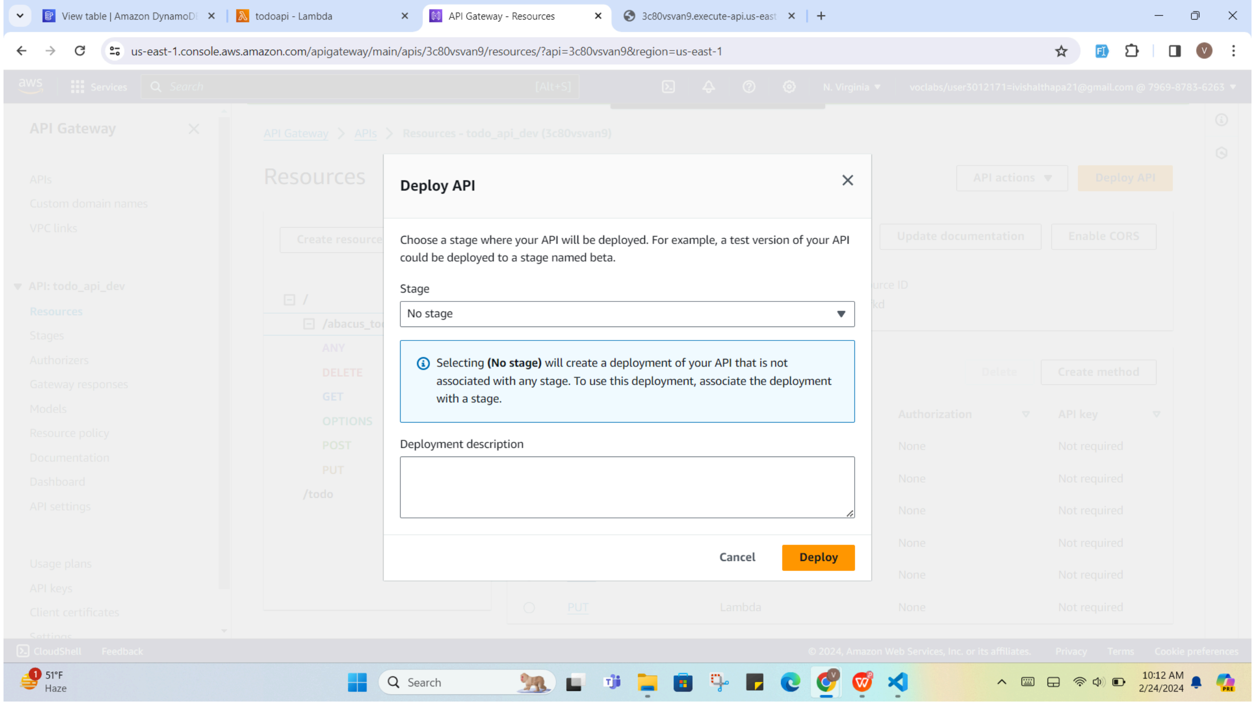
/
/abacus_todo
OPTIONS
/todo

/abacus_todo - OPTIONS - Method execution

ARN
arn:aws:execute-api:us-east-1:796987836263:3c80vsvan9/*/OPTIONS/abacus_todo

Resource ID
mtzfkd

Method request settings



us-east-1.console.aws.amazon.com/apigateway/main/apis/3c80vsvan9/resources/?api=3c80vsvan9&region=us-east-1

Deploy API

Choose a stage where your API will be deployed. For example, a test version of your API could be deployed to a stage named beta.

Stage

No stage

Selecting **(No stage)** will create a deployment of your API that is not associated with any stage. To use this deployment, associate the deployment with a stage.

Deployment description

Cancel Deploy

Now from the API settings inside the API gateway we can use the default endpoint to access the api methods and perform actions in dynamoDB.