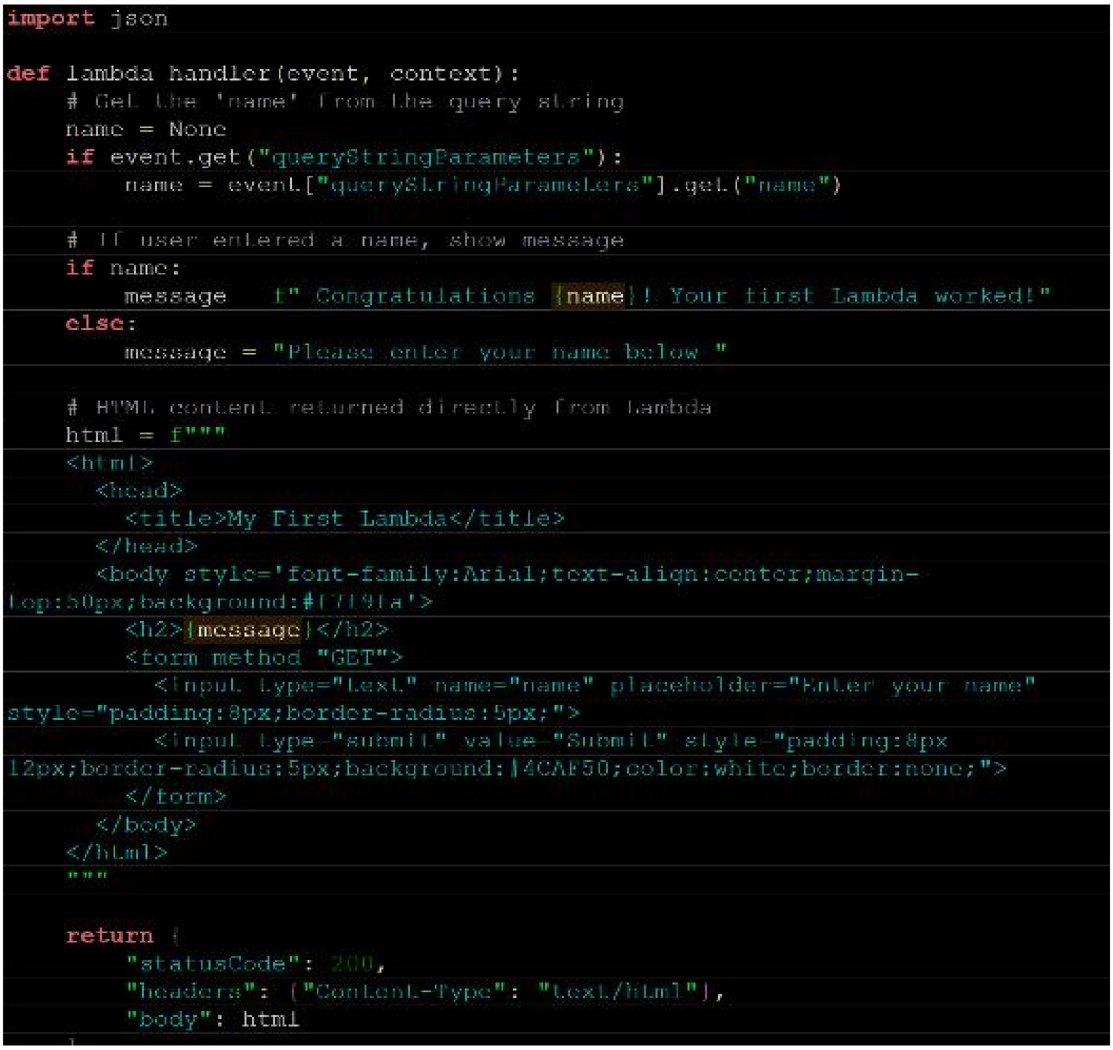
ELEVATE-LABS-TASK-3

Name - Mohammad Mansur
Gmail Id - mohdmansur1913@gmail.com
Designation - Cloud Intern
Github: https://github.com/mansur1913/elevate-labs-task-4

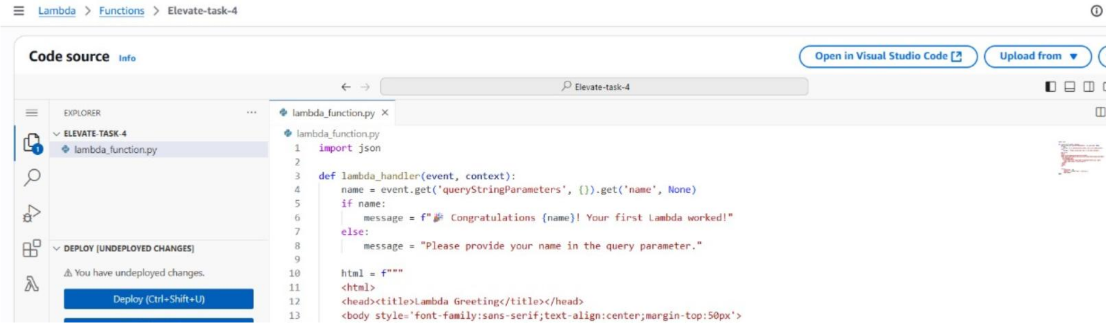
Deliverables:

Source code file (e.g., index.js or main.py)



```
import json

def lambda_handler(event, context):
    # Get the 'name' from the query string
    name = None
    if event.get("queryStringParameters"):
        name = event["queryStringParameters"].get("name")

    # If user entered a name, show message
    if name:
        message = f"Congratulations {name}! Your first Lambda worked!"
    else:
        message = "Please enter your name below "

    # HTML content returned directly from Lambda
    html = f"""
    <html>
      <head>
        <title>My First Lambda</title>
      </head>
      <body style='font-family:Arial;text-align:center;margin-
top:50px;background:#f7f9fa'>
        <h2>{message}</h2>
        <form method="GET">
          <input type="text" name="name" placeholder="Enter your name"
style="padding:8px;border-radius:5px;">
          <input type="submit" value="Submit" style="padding:8px
12px;border-radius:5px;background:#4CAF50;color:white;border:none;">
        </form>
      </body>
    </html>
    """

    return {
        "statusCode": 200,
        "headers": {"Content-Type": "text/html"},
        "body": html
    }
```

Screenshot(s) of successful deployment in the cloud console


Lambda > Functions > Elevate-task-4
Code source Info
Open in Visual Studio Code
Upload from
Elevate-task-4
EXPLORER
ELEVATE-TASK-4
lambda_function.py
DEPLOY [UNDEPLOYED CHANGES]
⚠ You have undeployed changes.
Deploy (Ctrl+Shift+U)

```python
import json

def lambda_handler(event, context):
    name = event.get('queryStringParameters', {}).get('name', None)
    if name:
        message = f"🎉 Congratulations {name}! Your first Lambda worked!"
    else:
        message = "Please provide your name in the query parameter."

    html = f"""
    <html>
    <head><title>Lambda Greeting</title></head>
    <body style='font-family:sans-serif;text-align:center;margin-top:50px'>
```

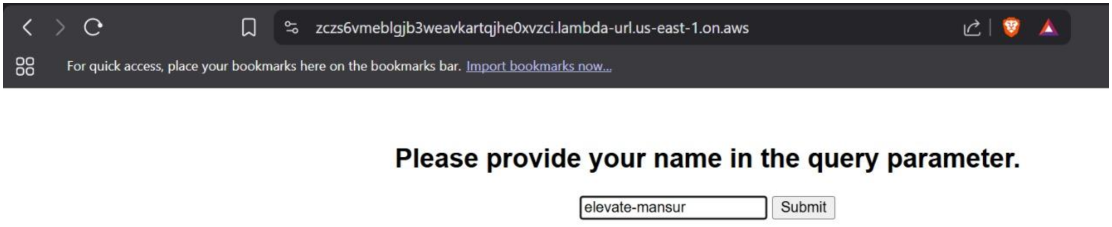
Function endpoint URL (if applicable)

I got eg: URL - https://4rhpketsnv46m6zs24ncj5qxwy0csceh.lambda-url.us-east-1.onaws/

Short note explaining how the function works

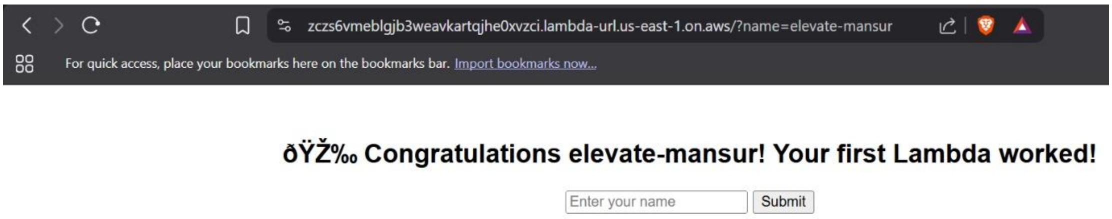
Create a serverless webserver with AWS lambda function, using python and html, which will ask to input for name and return a message,

As this is serverless we don't have to manage any servers or ec2 instance.


zczs6vmeblgjb3weavkartqjhe0xvzci.lambda-url.us-east-1.on.aws
For quick access, place your bookmarks here on the bookmarks bar. Import bookmarks now...

Please provide your name in the query parameter.

elevate-mansur Submit


zczs6vmeblgjb3weavkartqjhe0xvzci.lambda-url.us-east-1.on.aws/?name=elevate-mansur
For quick access, place your bookmarks here on the bookmarks bar. Import bookmarks now...

ðŸŽ‰ Congratulations elevate-mansur! Your first Lambda worked!

Enter your name Submit

Elevate-labs-Task #4

Task 4: Deploy a Serverless Function to the Cloud*

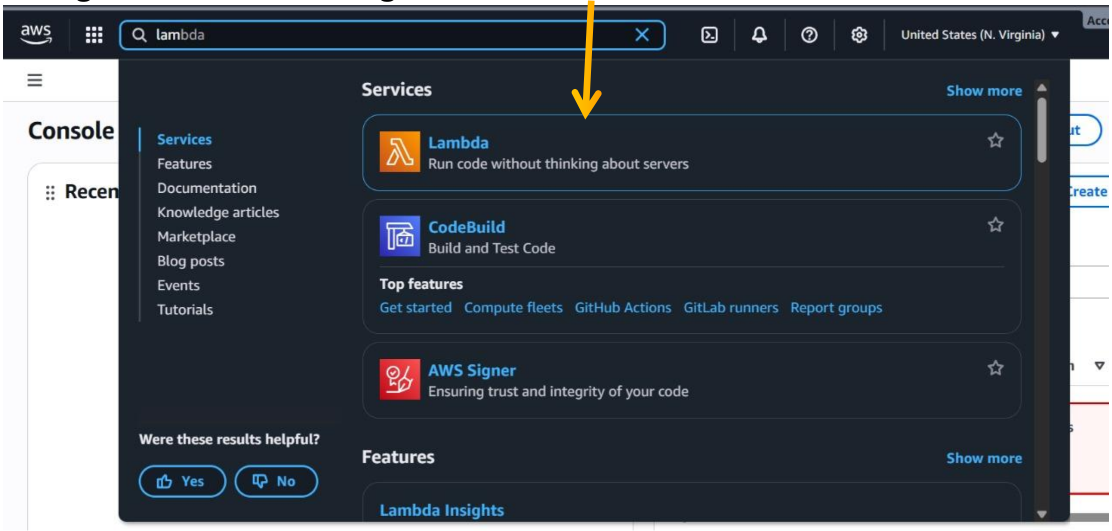
Objective:

To understand *serverless computing* by creating and deploying a simple *cloud function (FaaS)* that executes code automatically when triggered — without managing any servers.

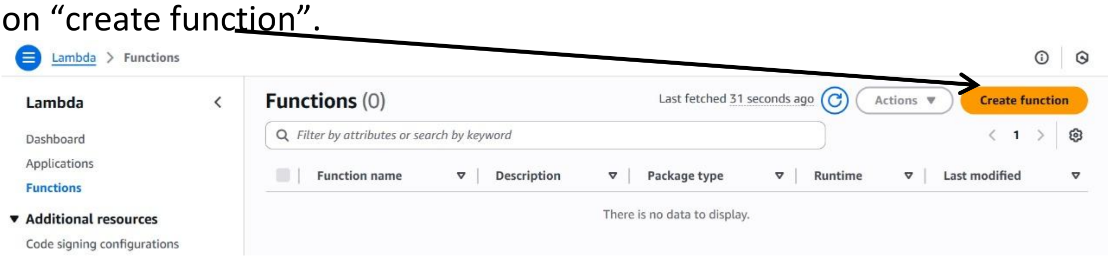
1. Lets create a serverless webserver with AWS lambda function, using python and html.

As this is serverless webserver we don't have to manage any server or ec2 instance.

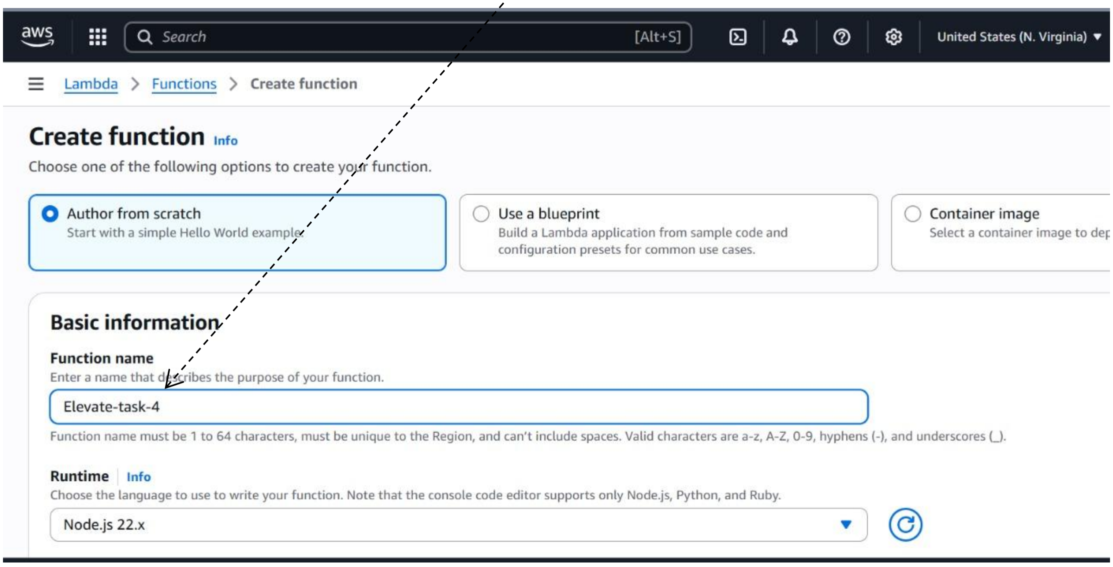
2. Login to AWS and navigate to "Lambda" in search bar.



3.Click on "Functions" in your left and hand side menu.. and then click on "create function".



4. After that a configuration window will appear.. In that name your function.. As I am naming it as “Elevate-task-4”.



aws
Search [Alt+S]
United States (N. Virginia) ▾
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 dep

Basic information

Function name
Enter a name that describes the purpose of your function.

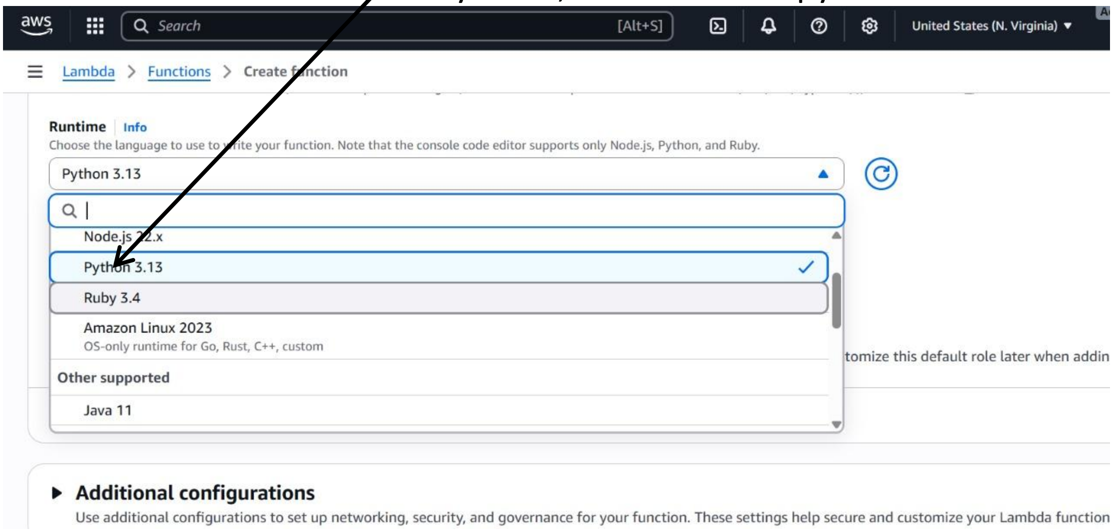
Elevate-task-4

Function name must be 1 to 64 characters, must be unique to the Region, and can't include spaces. Valid characters are a-z, A-Z, 0-9, hyphens (-), and underscores (_).

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

Node.js 22.x ▾

5. In runtime select “Python”, As we will use python code.



aws
Search [Alt+S]
United States (N. Virginia) ▾
Lambda > Functions > Create function

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

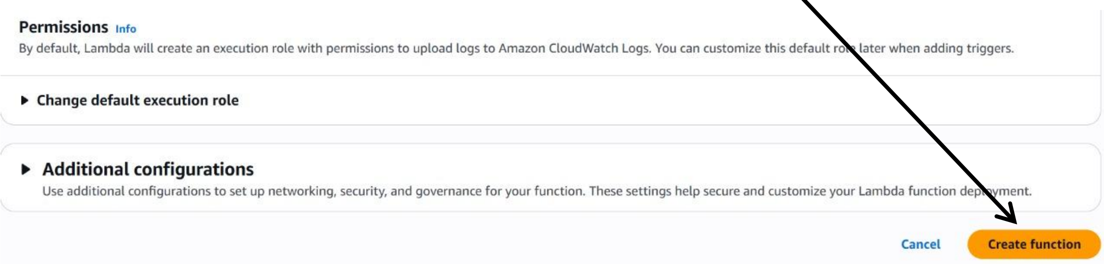
Python 3.13 ▴

Node.js 22.x
Python 3.13 ✓
Ruby 3.4
Amazon Linux 2023
OS-only runtime for Go, Rust, C++, custom
Other supported
Java 11

tomize this default role later when addin

Additional configurations
Use additional configurations to set up networking, security, and governance for your function. These settings help secure and customize your Lambda function

6. Leave everyhting as default and click on “create function”

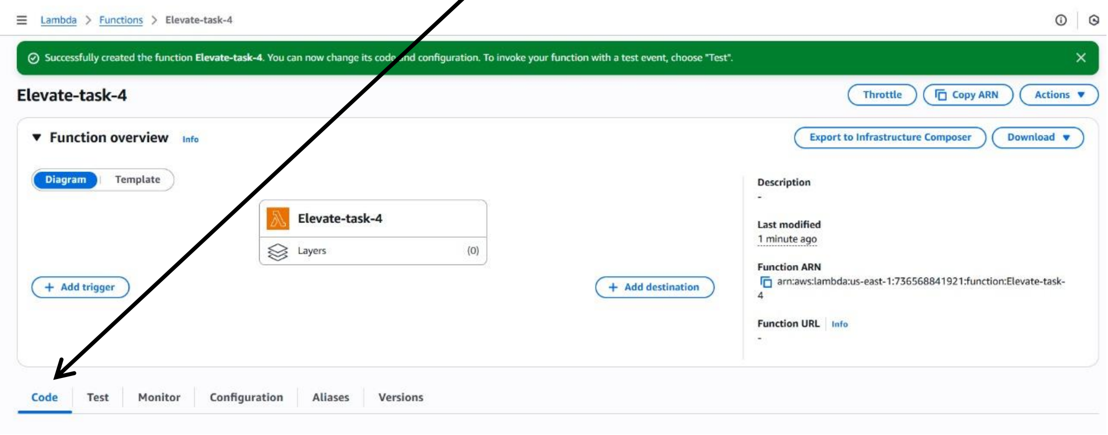


Permissions Info
By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.

▸ Change default execution role

Additional configurations
Use additional configurations to set up networking, security, and governance for your function. These settings help secure and customize your Lambda function deployment.

7. Once created a page will open like below, Were we can deploy our “code”, Scroll down and select “code”.



Successfully created the function **Elevate-task-4**. You can now change its code and configuration. To invoke your function with a test event, choose "Test".

Elevate-task-4

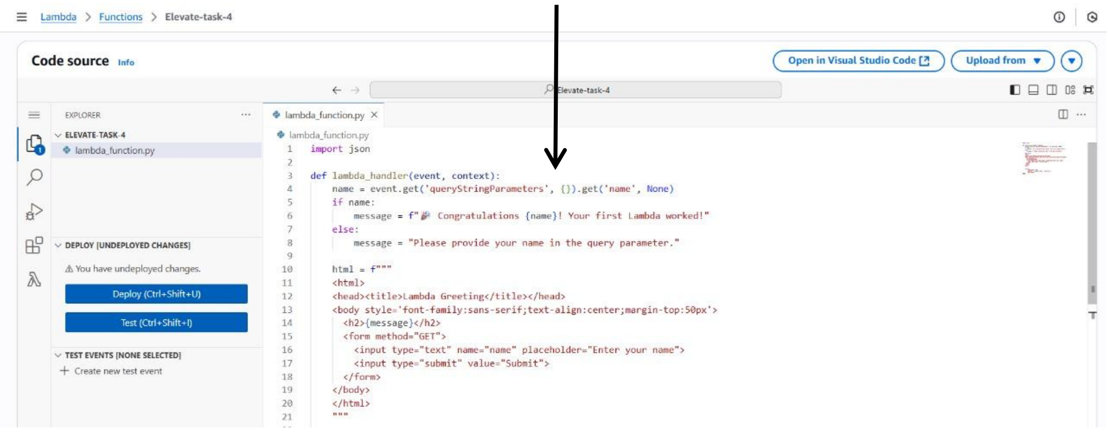
Description
-

Last modified
1 minute ago

Function ARN
arn:aws:lambda:us-east-1:736568841921:function:Elevate-task-4

Function URL Info
-

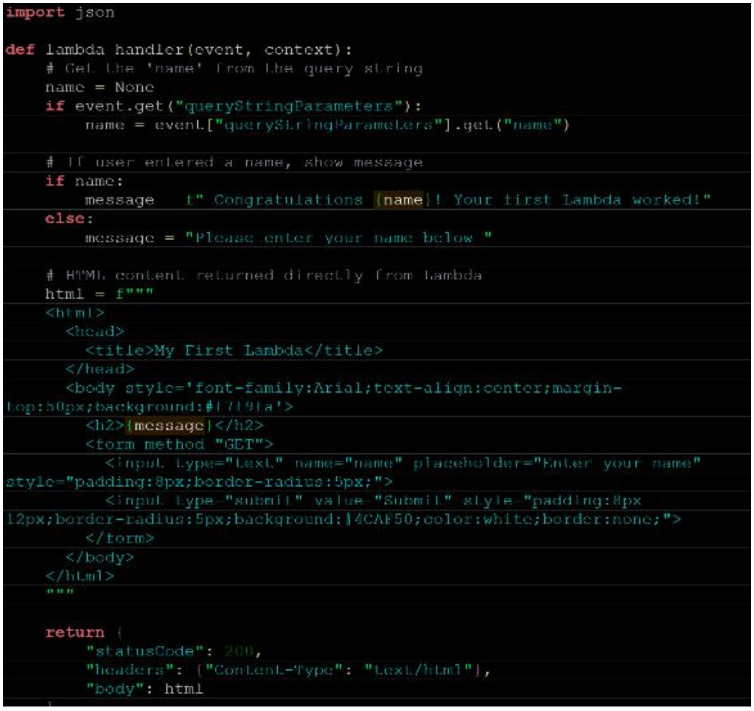
8. You can write the code in the given editor.



⚠ You have undeployed changes.

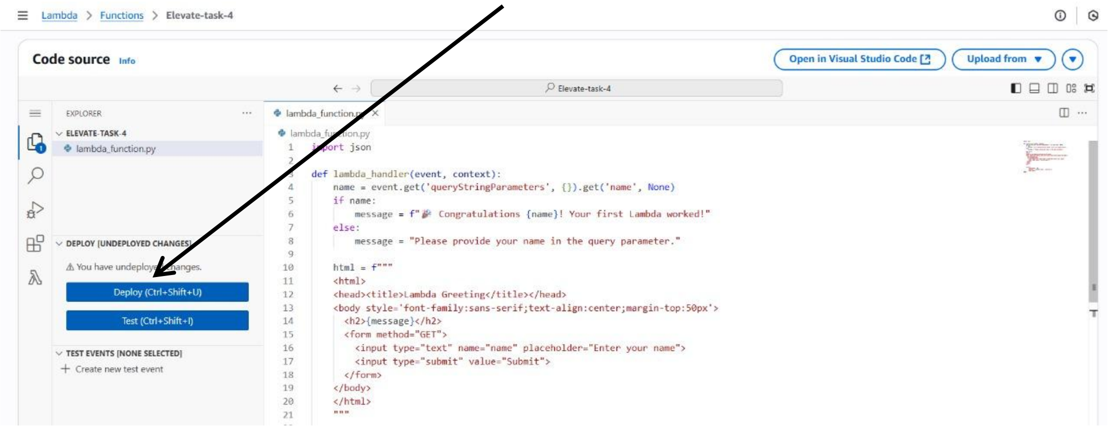
```python
import json

def lambda_handler(event, context):
    name = event.get('queryStringParameters', {}).get('name', None)
    if name:
        message = f"🎉 Congratulations {name}! Your first Lambda worked!"
    else:
        message = "Please provide your name in the query parameter."

    html = f"""
    <html>
    <head><title>Lambda Greeting</title></head>
    <body style='font-family:sans-serif;text-align:center;margin-top:50px'>
      <h2>{message}</h2>
      <form method="GET">
        <input type="text" name="name" placeholder="Enter your name">
        <input type="submit" value="Submit">
      </form>
    </body>
    </html>
    """
```

Python Code Below:



```python
import json

def lambda_handler(event, context):
    # Get the 'name' from the query string
    name = None
    if event.get("queryStringParameters"):
        name = event["queryStringParameters"].get("name")

    # If user entered a name, show message
    if name:
        message = f" Congratulations {name}! Your first Lambda worked!"
    else:
        message = "Please enter your name below "

    # HTML content returned directly from Lambda
    html = f"""
    <html>
      <head>
        <title>My First Lambda</title>
      </head>
      <body style='font-family:Arial;text-align:center;margin-
top:50px;background:#f7f9fa'>
        <h2>{message}</h2>
        <form method="GET">
          <input type="text" name="name" placeholder="Enter your name"
style="padding:8px;border-radius:5px;">
          <input type="submit" value="Submit" style="padding:8px
12px;border-radius:5px;background:#4CAF50;color:white;border:none;">
        </form>
      </body>
    </html>
    """

    return {
        "statusCode": 200,
        "headers": {"Content-Type": "text/html"},
        "body": html
    }
```

9. After writing code, click on “deploy”.

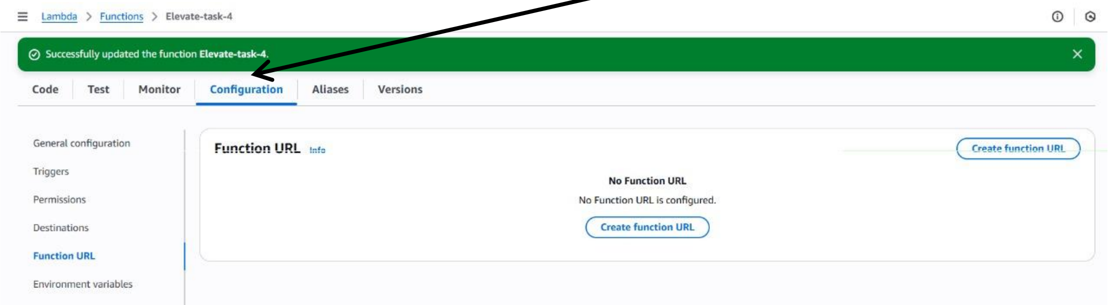


⚠ You have undeployed changes.

lambda_function.py

```
import json

def lambda_handler(event, context):
    name = event.get('queryStringParameters', {}).get('name', None)
    if name:
        message = f"🎉 Congratulations {name}! Your first Lambda worked!"
    else:
        message = "Please provide your name in the query parameter."

    html = f"""
    <html>
    <head><title>Lambda Greeting</title></head>
    <body style='font-family:sans-serif;text-align:center;margin-top:50px'>
      <h2>{message}</h2>
      <form method="GET">
        <input type="text" name="name" placeholder="Enter your name">
        <input type="submit" value="Submit">
      </form>
    </body>
    </html>
    """
```

10. Now we have successfully deployed the function, but we need URL to access the webpage, So click on “configuration” tab.

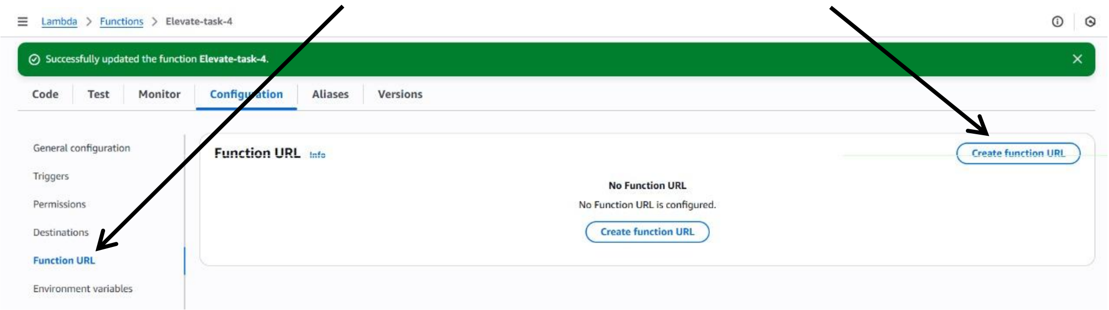


Successfully updated the function Elevate-task-4.

No Function URL

No Function URL is configured.

11. Then click on “Function URL” and click on “create function URL”

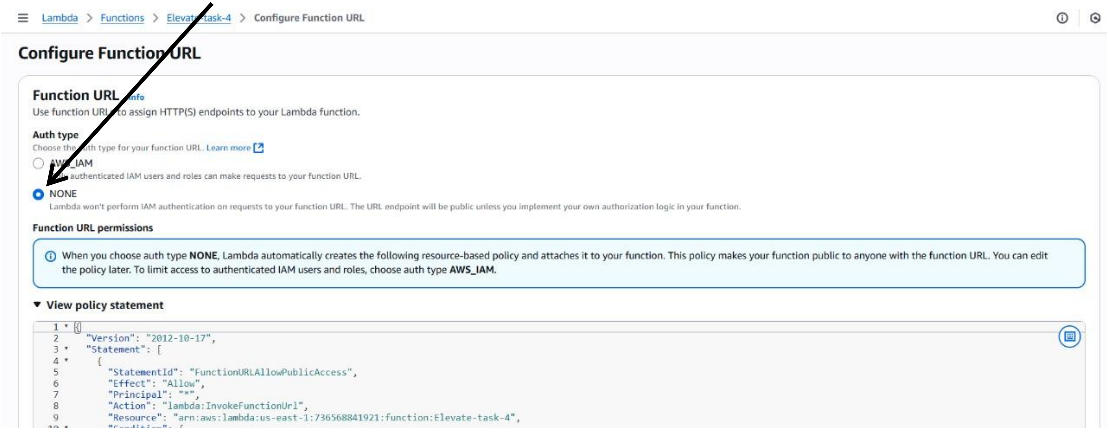


Successfully updated the function Elevate-task-4.

No Function URL

No Function URL is configured.

12. Select “NONE” then scroll down and click on ‘save’.


Lambda > Functions > Elevate-task-4 > Configure Function URL
Info
Learn more
View policy statement

Configure Function URL

Function URL

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

Auth type

Choose the auth type for your function URL.

○ AWS_IAM
Only authenticated IAM users and roles can make requests to your function URL.

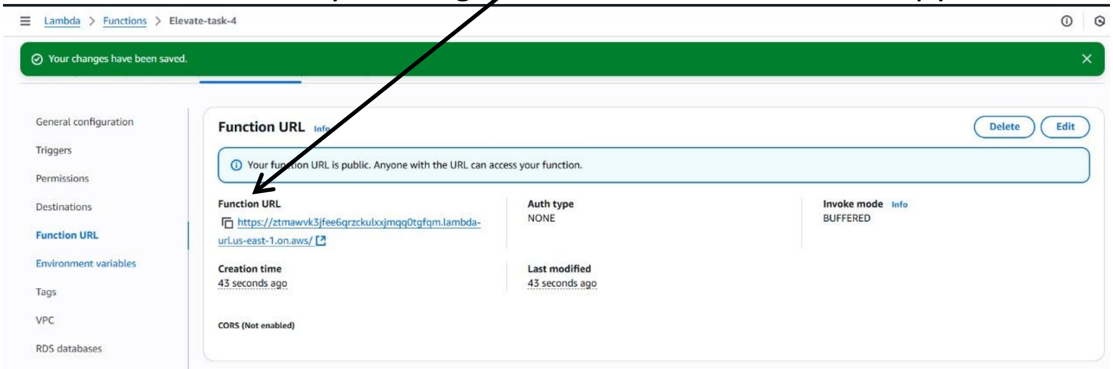
● NONE
Lambda won't perform IAM authentication on requests to your function URL. The URL endpoint will be public unless you implement your own authorization logic in your function.

Function URL permissions

When you choose auth type **NONE**, Lambda automatically creates the following resource-based policy and attaches it to your function. This policy makes your function public to anyone with the function URL. You can edit the policy later. To limit access to authenticated IAM users and roles, choose auth type **AWS_IAM**.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "StatementId": "FunctionURLAllowPublicAccess",
      "Effect": "Allow",
      "Principal": "*",
      "Action": "lambda:InvokeFunctionUrl",
      "Resource": "arn:aws:lambda:us-east-1:736568841921:function:Elevate-task-4",
```

13. After this you will get “Function URL” for now copy it.



Your changes have been saved.

Function URL

Your function URL is public. Anyone with the URL can access your function.

Function URL
https://ztmawvk3jfee6qrzckulxxjmqq0tgfqm.lambda-url.us-east-1.on.aws/

Auth type
NONE

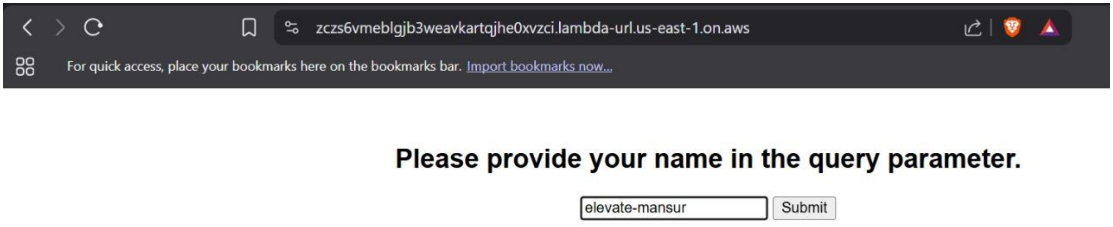
Invoke mode
BUFFERED

Creation time
43 seconds ago

Last modified
43 seconds ago

CORS (Not enabled)

14. Paste the URL in the browser and it will ask for input name.



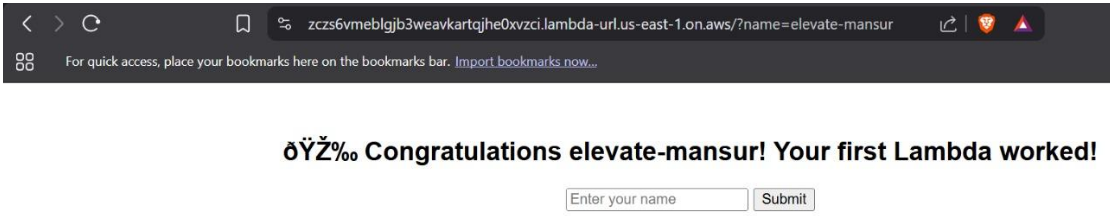
zczs6vmeblgjb3weavkartqjhe0xvzci.lambda-url.us-east-1.on.aws

For quick access, place your bookmarks here on the bookmarks bar.

Please provide your name in the query parameter.

elevate-mansur

15. Once we input the name it will return message as below:



zczs6vmeblgjb3weavkartqjhe0xvzci.lambda-url.us-east-1.on.aws/?name=elevate-mansur

For quick access, place your bookmarks here on the bookmarks bar.

ðŸŽ‰ Congratulations elevate-mansur! Your first Lambda worked!

Enter your name

End of task