ELEVATE-LABS-TASK-4

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:#1f191a'>
        <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

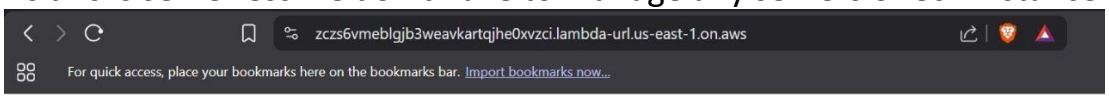
Function endpoint URL (if applicable)

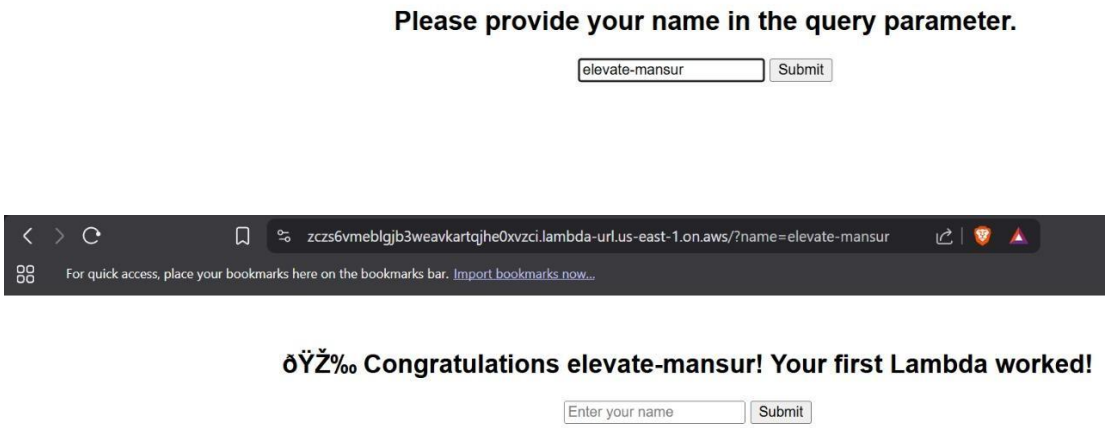
I got eg: URL - https://4rhpketsnv46m6zs24ncj5qxwy0csceh.lambda-url.us-east-1.onaws/

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.





Elevate-labs-Task #4

Task 4: Deploy a Serverless Function to the Cloud*

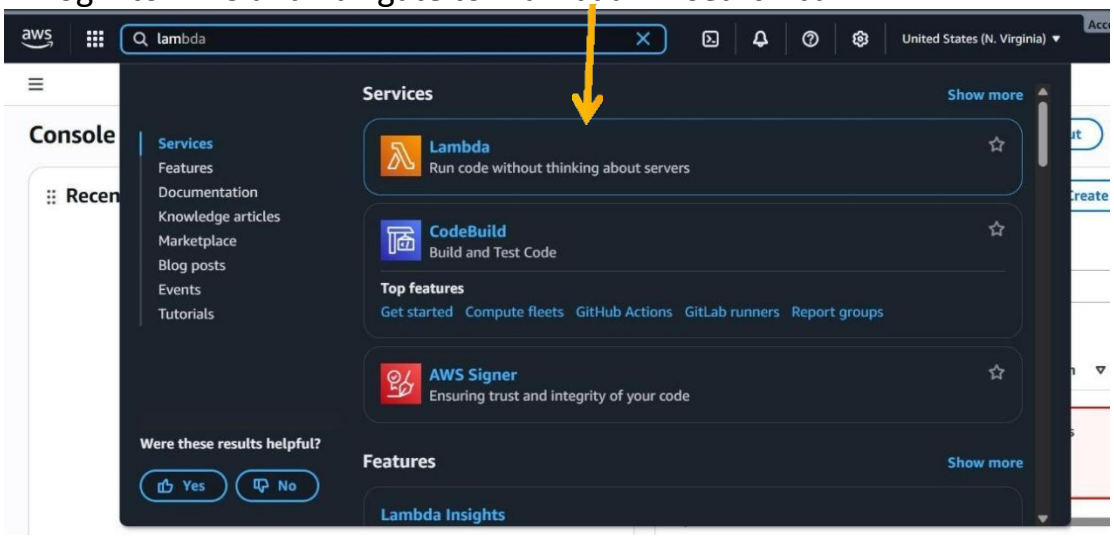
Objective:

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

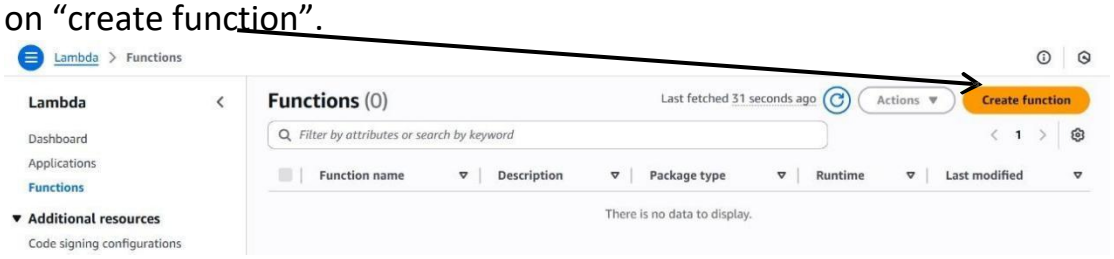
1. Lets create a serverless webserver with AWS lambda function, using python and html.

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

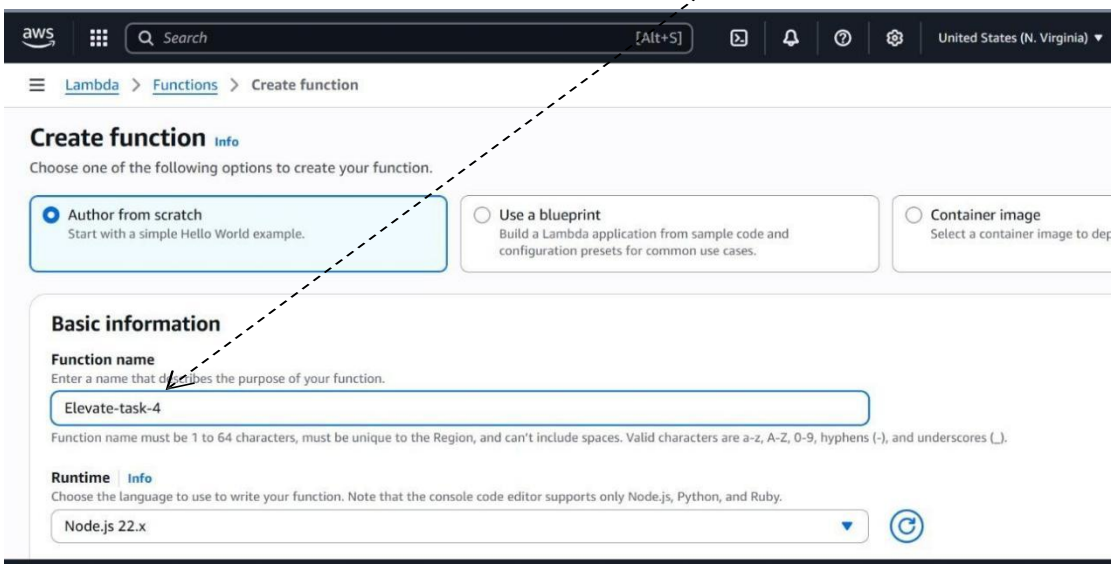
2. Login to AWS and navigate to "Lambda" in search bar.



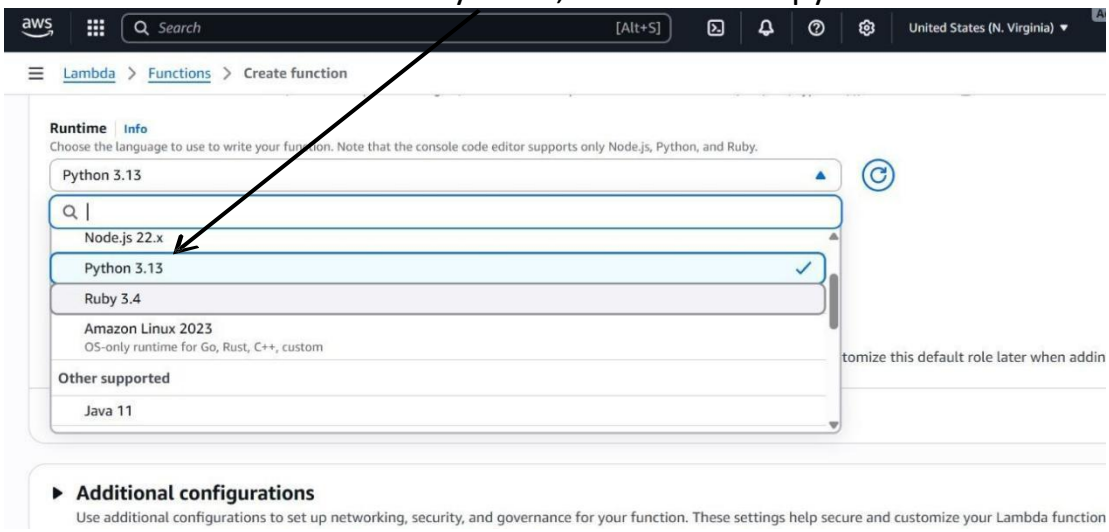
3. Click on "Functions" in your left and hand side menu.. and then click on "create function".



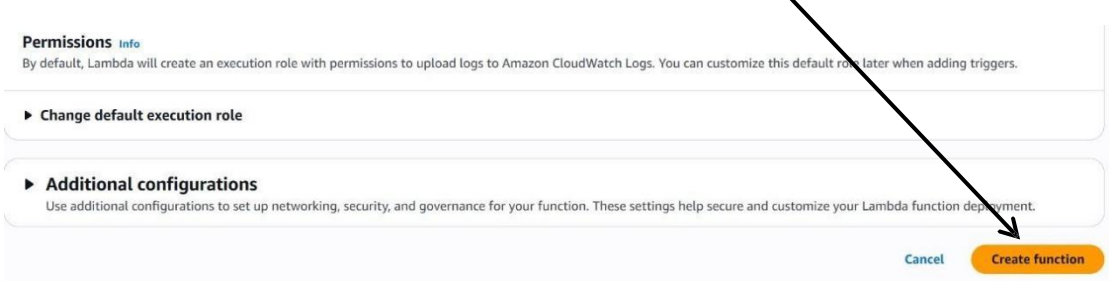
4. After that a configuration window will appear.. In that name your function.. As I am naming it as “Elevate-task-4”.



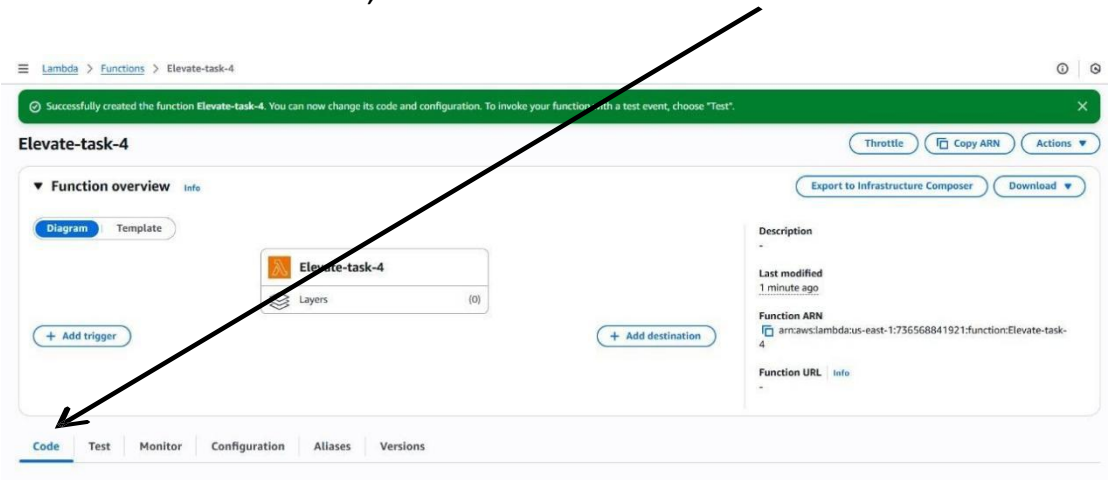
5. In runtime select “Python”, As we will use python code.



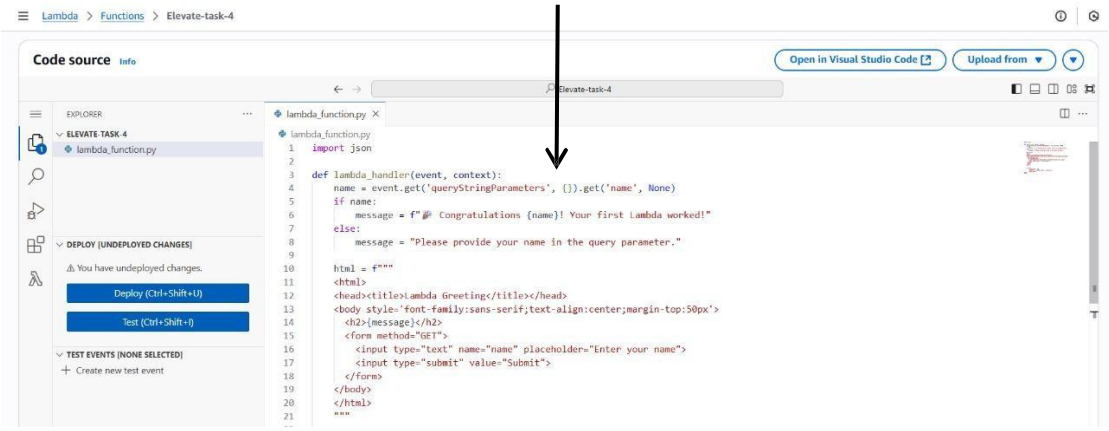
6. Leave everyhting as default and click on “create function”



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



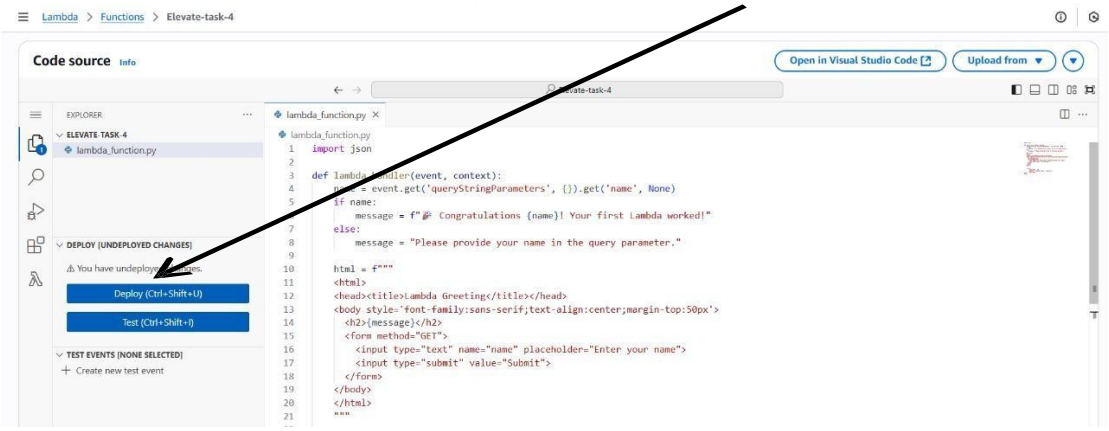
8. You can write the code in the given editor.



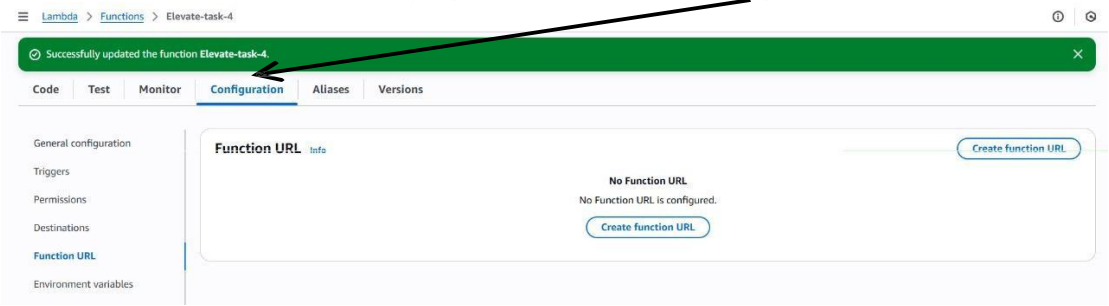
Python Code Below:

```
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
```

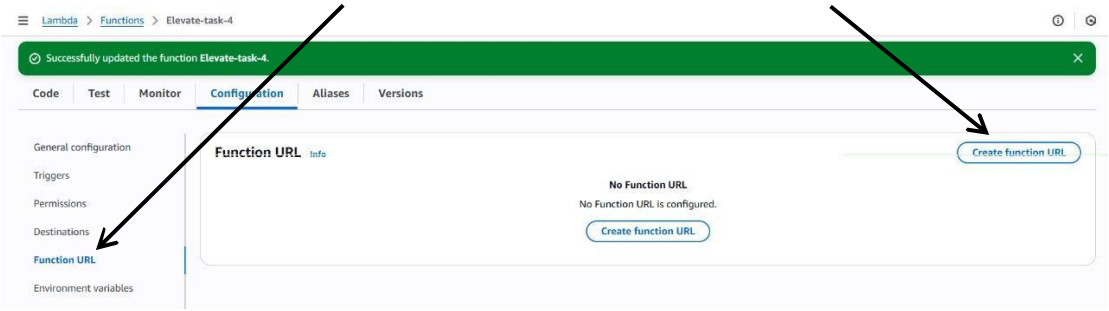
9. After writing code, click on “deploy”.



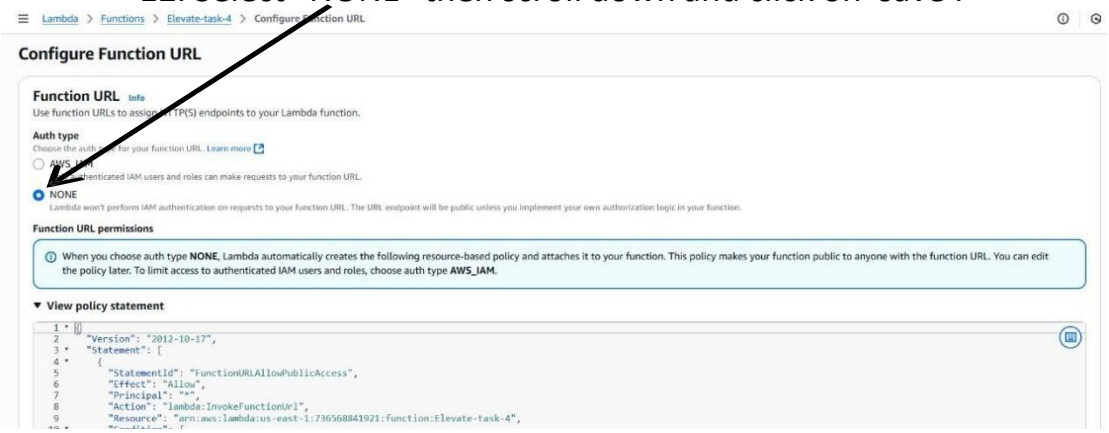
10. Now we have successfully deployed the function, but we need URL to access the webpage, So click on “configuration” tab.



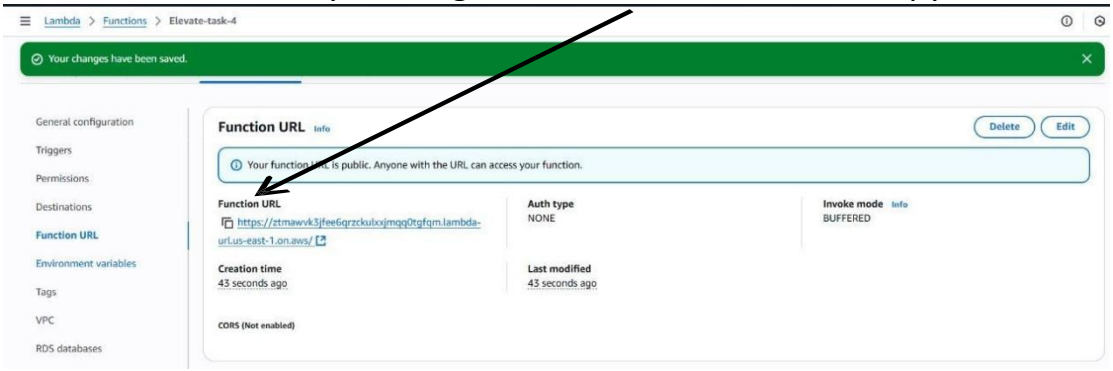
11. Then click on “Function URL” and click on “create function URL”



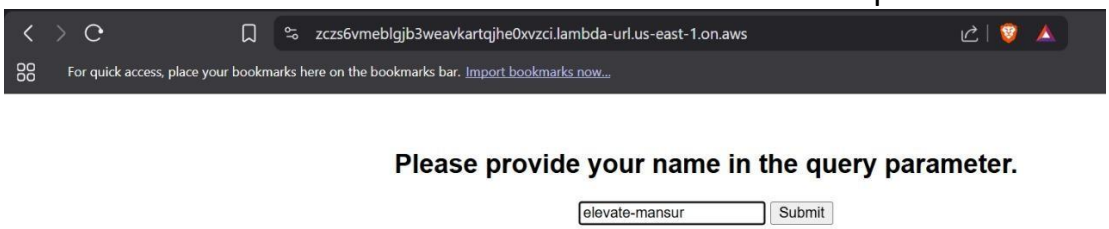
12. Select “NONE” then scroll down and click on ‘save’.



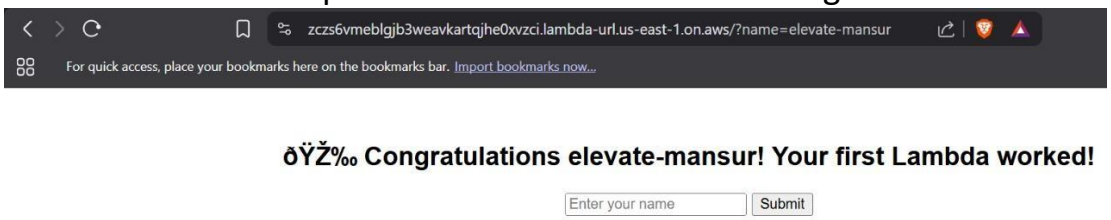
13. After this you will get “Function URL” for now copy it.



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



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



End of task