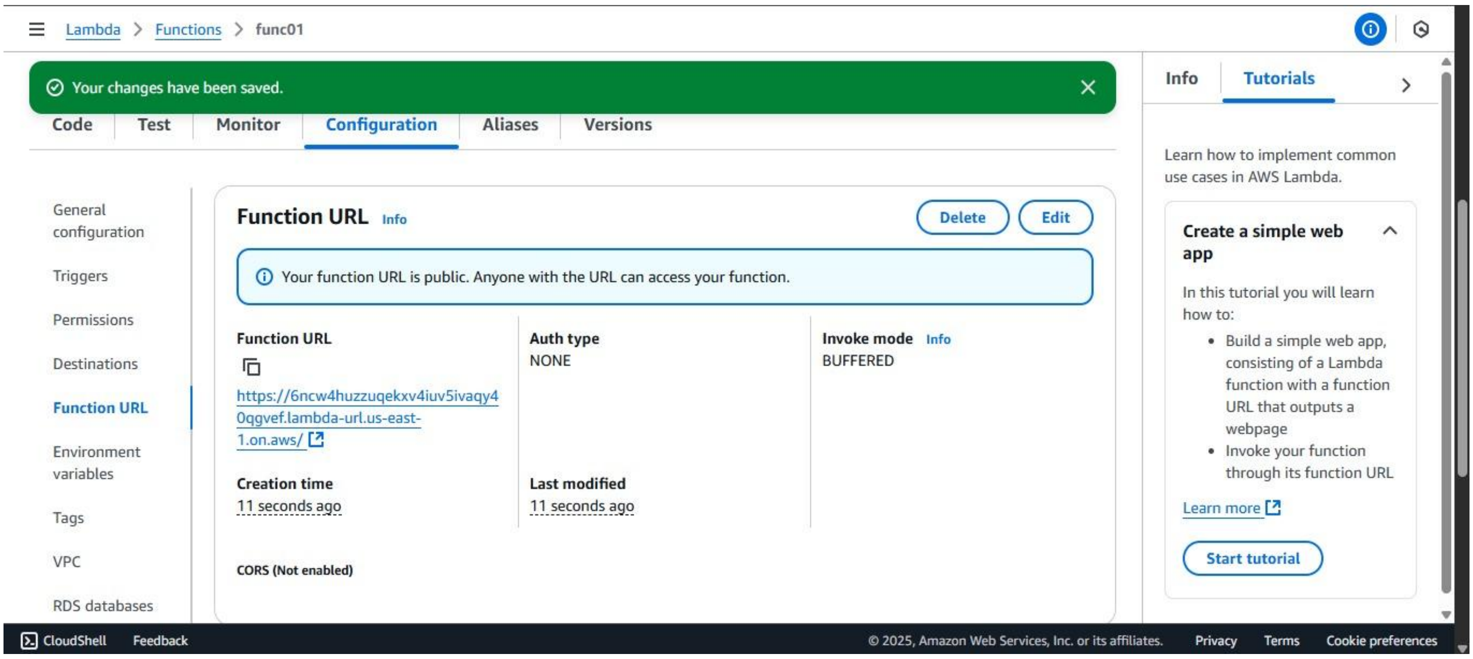
Step 7: Create Function URL
1. Go to the **Configuration** tab.
2. Under the left-side menu, click **Function URL**.
3. Click **Create function URL**.
4. For **Auth type**, choose **None**.
5. Click **Save**.**Assignment 15**

Step 8: Test the URL
1. Once the Function URL is created, click on it.
2. A new browser tab opens, showing your Lambda function output (e.g., "Welcome from Sneha!").

⚠ If you see an error, ensure your function code returns a valid HTTP response.

✓ Step 9: Delete Resources
1. Go back to **Configuration** > **Function URL** and **delete the URL**.
2. Then return to the **Lambda dashboard**, select your function, and click **Delete**.

Expected Output
• After deployment and testing, your function should return:
"Welcome from Sneha!"
• You should be able to view this output via the Test button and directly from the Function URL.



Your changes have been saved.

Function URL Info

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

Function URL
https://6ncw4huzzuqekxv4iuv5ivaqy40qgvef.lambda-url.us-east-1.on.aws/

Auth type
NONE

Invoke mode Info
BUFFERED

Creation time
11 seconds ago

Last modified
11 seconds ago

CORS (Not enabled)

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

© 2025, Amazon Web Services, Inc. or its affiliates.

Step 4: Modify the Code

1. Wait for the function page to load. You'll be taken to the function dashboard.
2. Under the **Code** tab, locate and open the index.mjs or main file (for Python, it might be lambda_function.py).
3. Replace any occurrence of the word **"lambda"** with **"sneha"** in the sample code.

✍ **Example (Node.js)**:

```
export const handler = async (event) => {
const response = {
statusCode: 200,
body: JSON.stringify('Welcome from Sneha!'),
};
return response;
};
```

4. Click **File > Save** to save the code

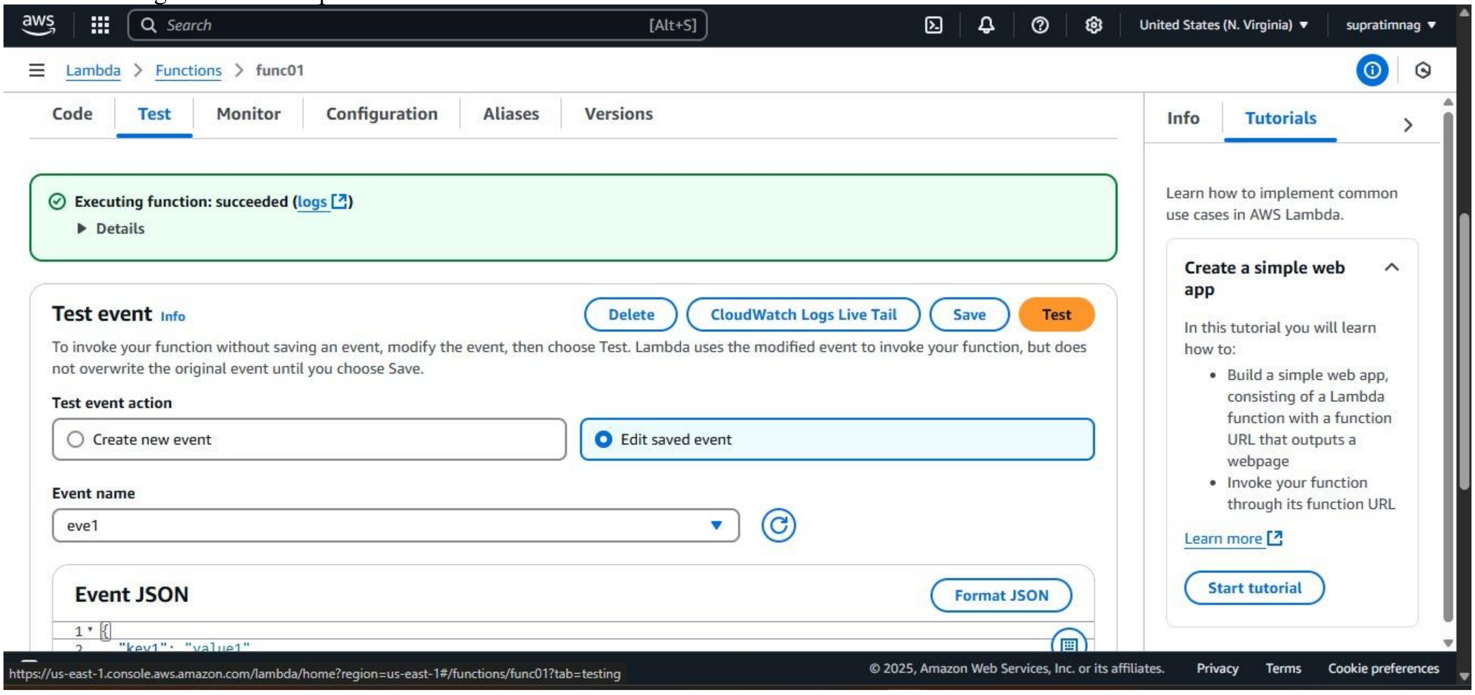
Step 5: Create and Run a Test Event

1. Click on the **Test** button (top-right).
2. Select **"Create new test event."**
3. Give it an **Event name**, e.g., eve1.
4. Leave the default JSON data as is (you don't need to change anything).
5. Click **Save**.
6. Now click **Test** to execute the Lambda function.

   Note: If you don't see your message change (e.g., "sneha"), it means you haven't deployed the latest code yet.

Step 6: Deploy and Re-Test

1. Click the **Deploy** button to apply your code changes.
2. Click **Test** again to see the updated result.



Executing function: succeeded (logs)
▶ Details

Test event Info

To invoke your function without saving an event, modify the event, then choose Test. Lambda uses the modified event to invoke your function, but does not overwrite the original event until you choose Save.

Test event action

○ Create new event ● Edit saved event

Event name

eve1

Event JSON

```
{
    "key1": "value1"
```

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

https://us-east-1.console.aws.amazon.com/lambda/home?region=us-east-1#/functions/func01?tab=testing

© 2025, Amazon Web Services, Inc. or its affiliates.

Assignment 15
Create Serverless Computing Service using AWS Lambda

📝 Objective

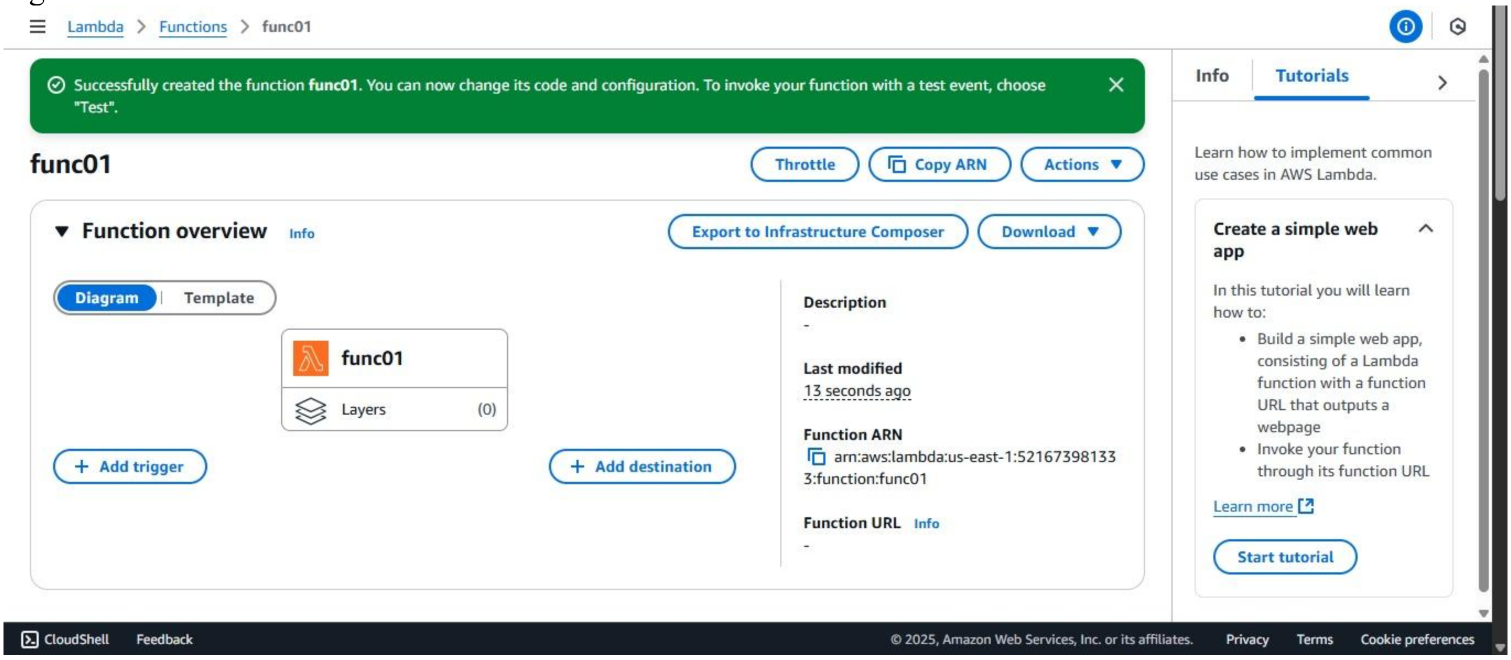
To create and deploy a simple AWS Lambda function that prints a custom welcome message — demonstrating serverless computing on AWS.

🚀 Part 1: Creating the Lambda Function

Step 1: Open Lambda Service

1. Log in to your AWS Console: https://aws.amazon.com/console/
2. In the **Search bar**, type **Lambda** and click on it.

Explanation:
AWS Lambda lets you run code without managing servers. You only focus on writing the function logic.



Lambda > Functions > func01

Successfully created the function **func01**. You can now change its code and configuration. To invoke your function with a test event, choose "Test".

func01

Function overview Info

Description
-

Last modified
13 seconds ago

Function ARN
arn:aws:lambda:us-east-1:52167398133 3:function:func01

Function URL Info
-

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

© 2025, Amazon Web Services, Inc. or its affiliates.

Step 2: Create the Function

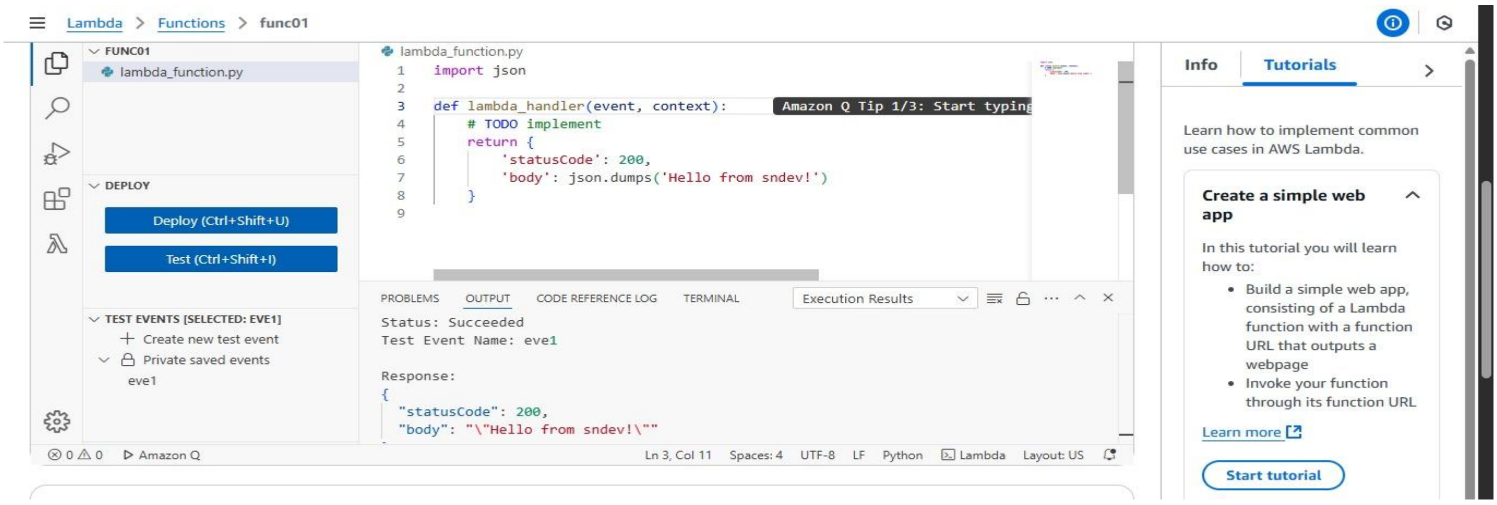
1. Click on the **“Create function”** button.
2. Select **“Author from scratch.”**

Step 3: Set Function Details

- **Function name**: e.g., func_x1
- **Runtime**: Choose **Python 3.9** or any preferred runtime (Node.js, etc.)

Tip: The runtime determines what programming language your Lambda function will use.

3. Scroll down and leave all other settings as **default**.
4. Click **Create function**.



Lambda > Functions > func01

lambda_function.py

```
import json

def lambda_handler(event, context):
    # TODO implement
    return {
        'statusCode': 200,
        'body': json.dumps('Hello from sndev!')
    }
```

```
Status: Succeeded
Test Event Name: eve1

Response:
{
  "statusCode": 200,
  "body": "\"Hello from sndev!\""
```

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