**Assignment-2 Solutions**

**Writing the Python code for Transpotion of matrix:**

def transpose\_matrix(matrix):

    if not matrix:

        return []

    num\_rows = len(matrix)

    num\_cols = len(matrix[0])

    transposed\_matrix = [[0] \* num\_rows for \_ in range(num\_cols)]

    for i in range(num\_rows):

        for j in range(num\_cols):

            transposed\_matrix[j][i] = matrix[i][j]

    return transposed\_matrix

**Save and zip this file with the folder.**

**Creation of Layers in the Lambda console:**

* Clicked the create layer in the right corner of the console
* Provided the layer name as transpose and uploaded the zip file which had the python file in it
* Chose the runtime as Python 3.8 and created the layer

**Creation of lambda function and adding the created layer to the function**

* Open the AWS console
* In the lambda section, clicked on create function
* Gave the naming conventions as assignment2-lambda1 and chose the runtime as Python 3.8
* Successfully created the lambda function
* Choosing the add layer option to link the layer to the function
* In the custom layer, add the previously created layer and select the version
* Successfully added the layer to the function

import json

import transpose  # Import the transpose\_layer package

def lambda\_handler(event, context):

    # Example input matrix

    matrix = [[1, 2, 3],

              [4, 5, 6],

              [7, 8, 9]]

    # Transpose the matrix using the function from the layer

    transposed\_matrix = transpose.transpose\_matrix(matrix)

    # Convert the transposed matrix to JSON format

    transposed\_json = json.dumps(transposed\_matrix)

    # Print the transposed matrix

    print(transposed\_json)

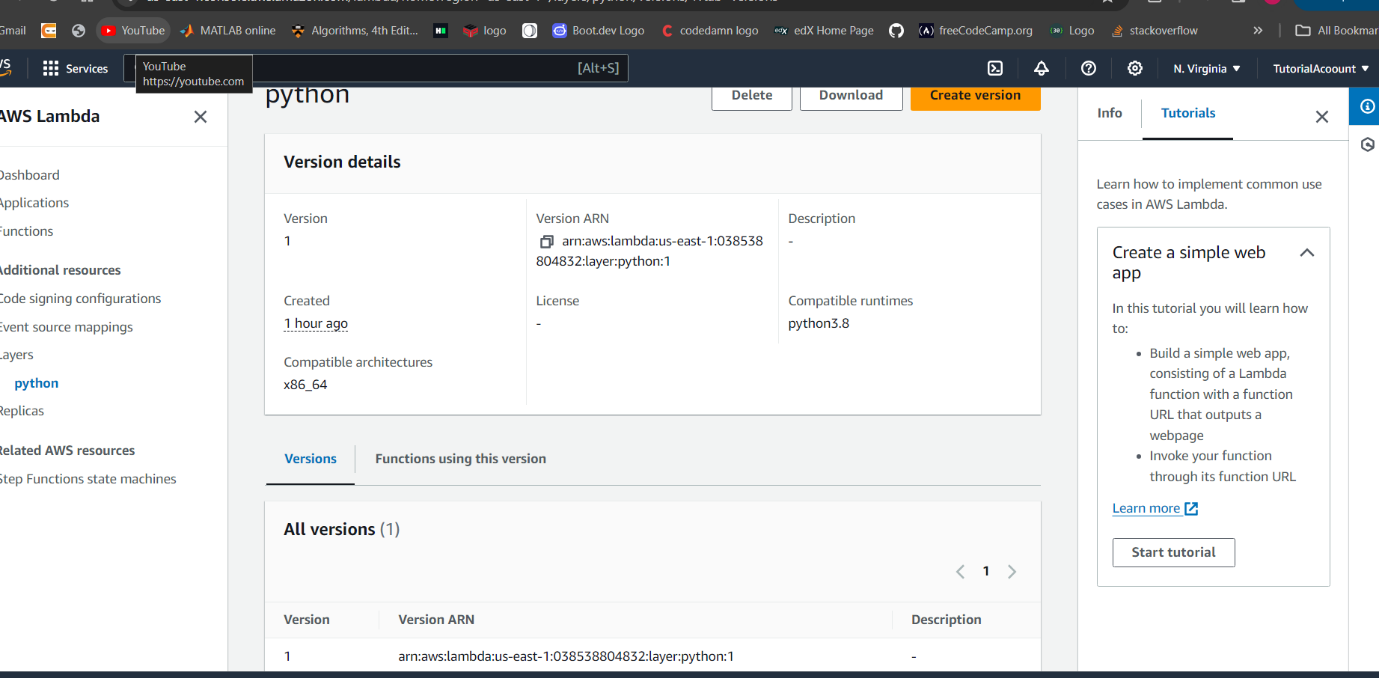
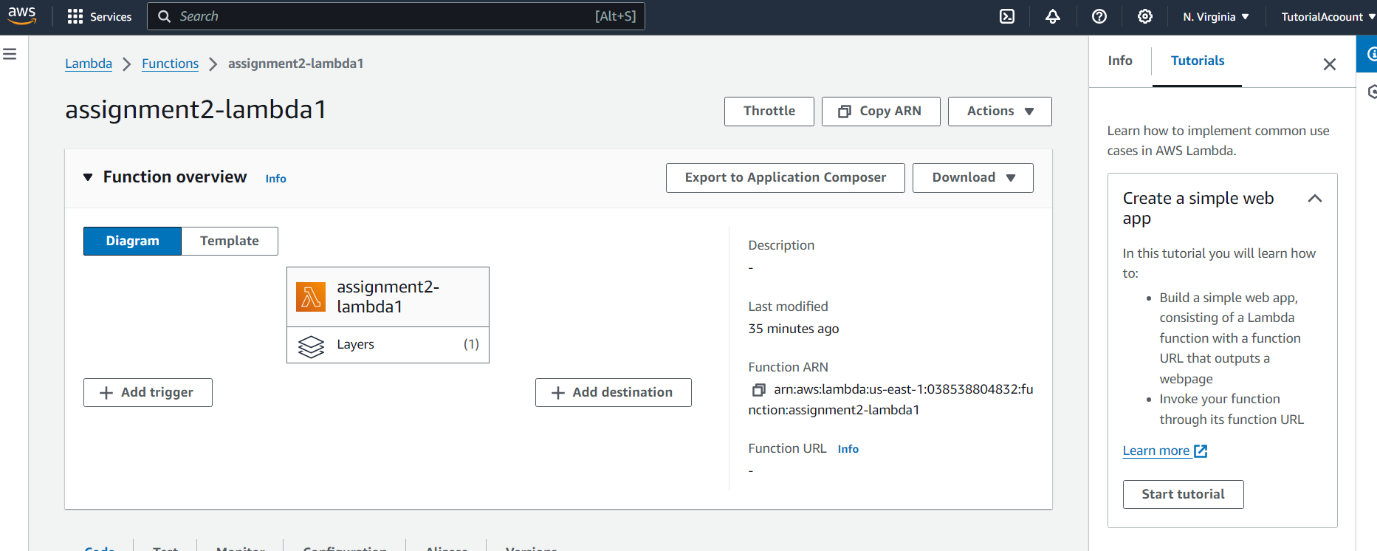
    return {

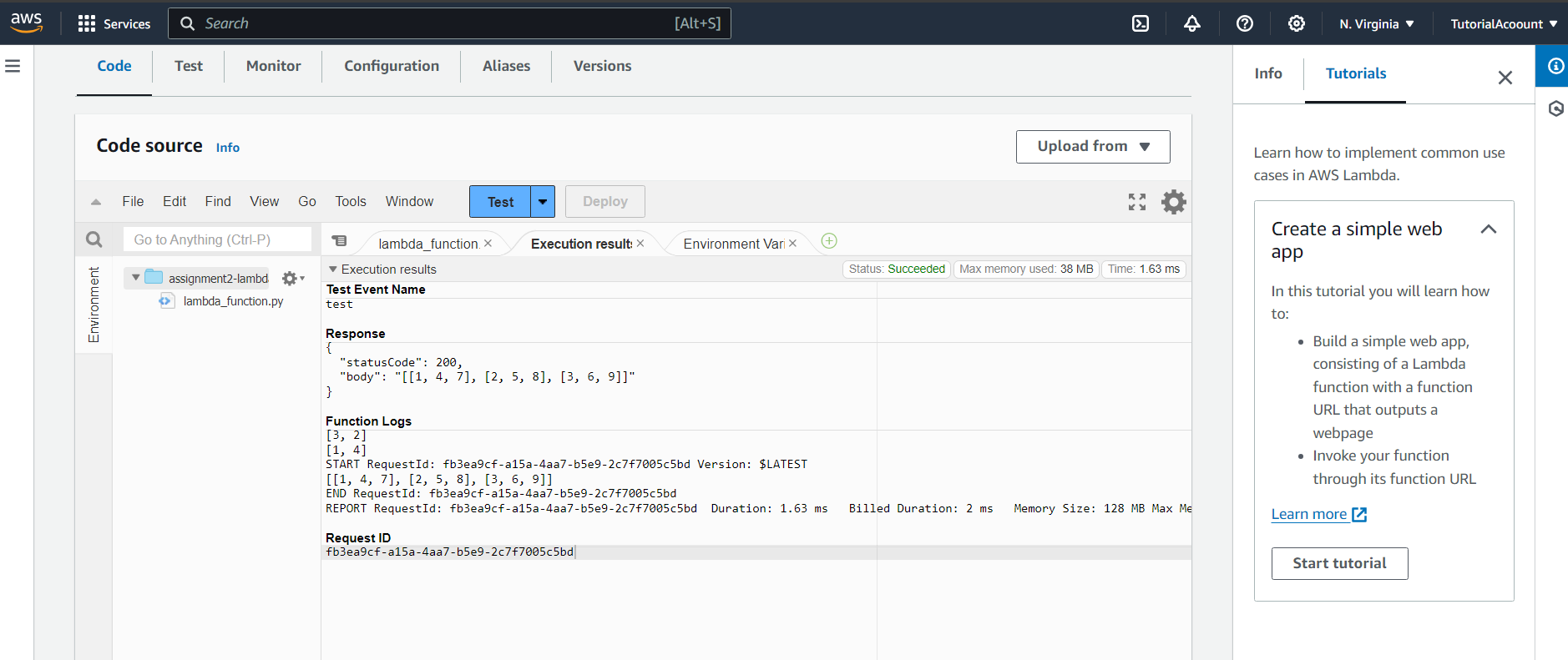
        'statusCode': 200,

        'body': transposed\_json

    }

The above code should be present in the lambda function.

**Snapshots:**

****