

St. Francis Institute of Technology, Mumbai-400 103  
**Department Of Information Technology**

A.Y. 2024-2025

Class: TE-ITA/B, Semester: V

Subject: **Advanced DevOps Lab**

**Experiment – 09: To understand AWS Lambda, its workflow, and to create the first Lambda function using Python/Java.**

1. **Aim:** To write first Lambda function using Python/Java/Node.js.
2. **Objectives:** Aim of this experiment is that, the students will learn:
  - Serverless cloud concept and how to create Lambda function in various languages
  - Invoke Lambda function
  - Monitoring AWS Lambda
3. **Lab objective mapped : ITL504.6:** To demonstrate a composition of nano services using AWS Lambda and Create Functions with the Serverless Framework.
4. **Prerequisite:** Knowledge of Python/Java/Node.js , AWS console.
5. **Requirements:** AWS account, browser, Personal Computer, Windows operating system, Internet Connection, Google Doc.
6. **Pre-Experiment Exercise:**

Answer the following (write in hand)

Explain Serverless concept?

What are the applications of AWS Lambda?

**Brief Theory:**

AWS Lambda is a compute service that lets you run code without provisioning or managing servers.

Lambda runs your code on a high-availability compute infrastructure and performs all of the administration of the compute resources, including server and operating system maintenance, capacity provisioning and automatic scaling, and logging. With Lambda, all you need to do is supply your code in one of the language runtimes that Lambda supports.

**When to use Lambda**

Lambda is an ideal compute service for application scenarios that need to scale up rapidly, and scale down to zero when not in demand. For example, you can use Lambda for:

**File processing:** Use Amazon Simple Storage Service (Amazon S3) to trigger Lambda data processing in real time after an upload.

**Stream processing:** Use Lambda and Amazon Kinesis to process real-time streaming data for application activity tracking, transaction order processing, clickstream analysis, data cleansing, log filtering, indexing, social media analysis, Internet of Things (IoT) device data telemetry, and metering.

**Web applications:** Combine Lambda with other AWS services to build powerful web applications that automatically scale up and down and run in a highly available configuration across multiple data centers.

**IoT backends:** Build serverless backends using Lambda to handle web, mobile, IoT, and third-party API requests.

**Mobile backends:** Build backends using Lambda and Amazon API Gateway to authenticate and process API requests. Use AWS Amplify to easily integrate with your iOS, Android, Web, and React Native frontends.

## 7. Laboratory Exercise

### A. Procedure:

#### a. Perform following steps (attach screenshots)

- Enter Lambda service from console
- Select a Lambda Blueprint
- Configure and create Lambda function
- Invoke Lambda function and verify results
- AWS Lambda automatically monitors Lambda functions and reports metrics
- Clean up Lambda function

## 8. Post-Experiments Exercise

### A. Extended Theory:(attach SS)

Create the Lambda function using Node.js

### B. Questions:(write in hand)

- What is an AWS Lambda function?
- EC2 vs Lambda vs Elastic Beanstalk
- Technical Limitations of AWS Lambda :
  - a. The maximum time a function can run is \_\_\_\_ minutes.
  - b. The default timeout is \_\_\_\_.
  - c. Lambda is unsuitable for \_\_\_\_\_ (long-running workloads / short-running workloads).
  - d. The payload for each invocation of a Lambda function is limited to \_\_ MB.
  - e. \_\_\_\_\_ is limited to just under 3GB.

### C. Conclusion:(write in hand)

1. Write what was performed in the experiment
2. Mention a few applications of what was studied.
3. Write the significance of the studied topic

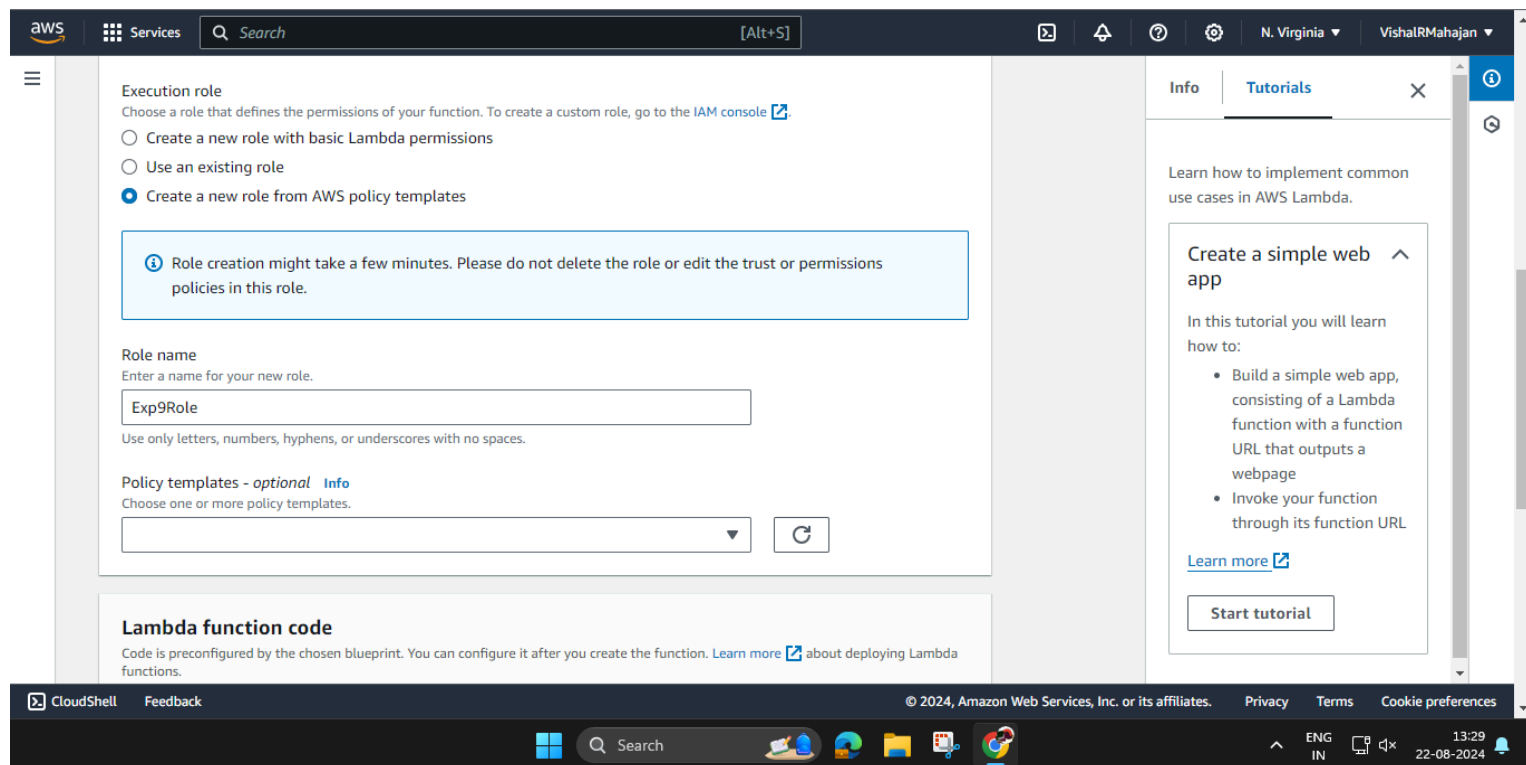
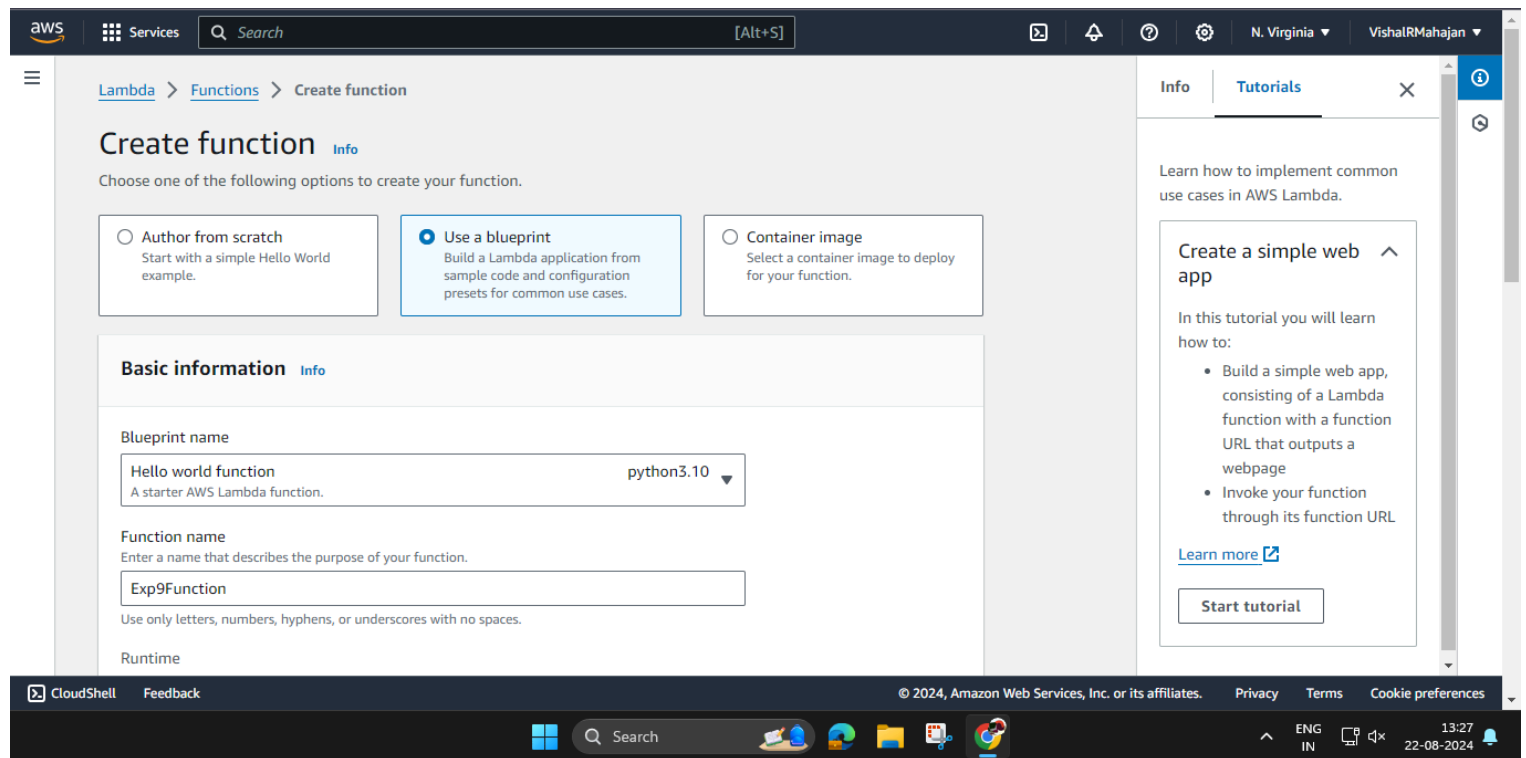
### D. References:

1. <https://aws.amazon.com/getting-started/hands-on>
  2. <https://www.scalyr.com/blog/aws-lambda-tutorial/>
  3. <https://docs.aws.amazon.com/lambda/latest/dg/welcome.html>
-

Name: Vishal Rajesh Mahajan  
Class: TE IT A

ADL EXP09  
Roll No: 62

## SCREENSHOTS:



**Lambda function code**

Code is preconfigured by the chosen blueprint. You can configure it after you create the function. [Learn more](#) about deploying Lambda functions.

This function contains external libraries.

```
1 import json
2 print('Loading function')
3
4
5
6 def lambda_handler(event, context):
7     #print('Received event: ' + json.dumps(event, indent=2))
8     print('value1 = ' + event['key1'])
9     print('value2 = ' + event['key2'])
10    print('value3 = ' + event['key3'])
11    return event['key1'] # Echo back the first key value
12    #raise Exception('Something went wrong')
13
```

Cancel Create function

**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

**Configure test event**

A test event is a JSON object that mocks the structure of requests emitted by AWS services to invoke a Lambda function. Use it to see the function's invocation result.

To invoke your function without saving an event, configure the JSON event, then choose Test.

Test event action

☒ Create new event ☐ Edit saved event

Event name

Exp9Event

Maximum of 25 characters consisting of letters, numbers, dots, hyphens and underscores.

Event sharing settings

☒ Private

This event is only available in the Lambda console and to the event creator. You can configure a total of 10. [Learn more](#)

☐ Shareable

This event is available to IAM users within the same account who have permissions to access and use shareable events. [Learn more](#)

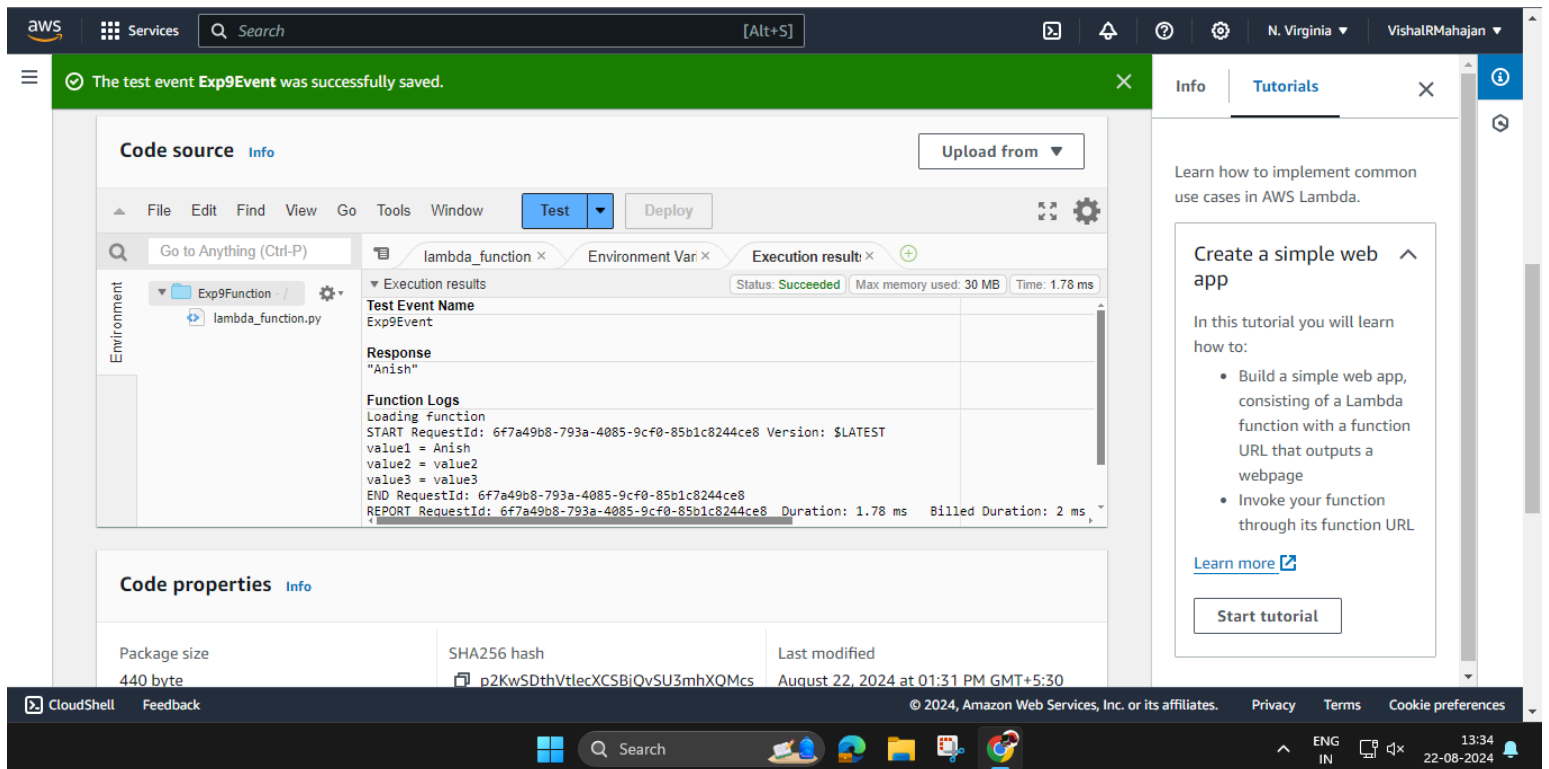
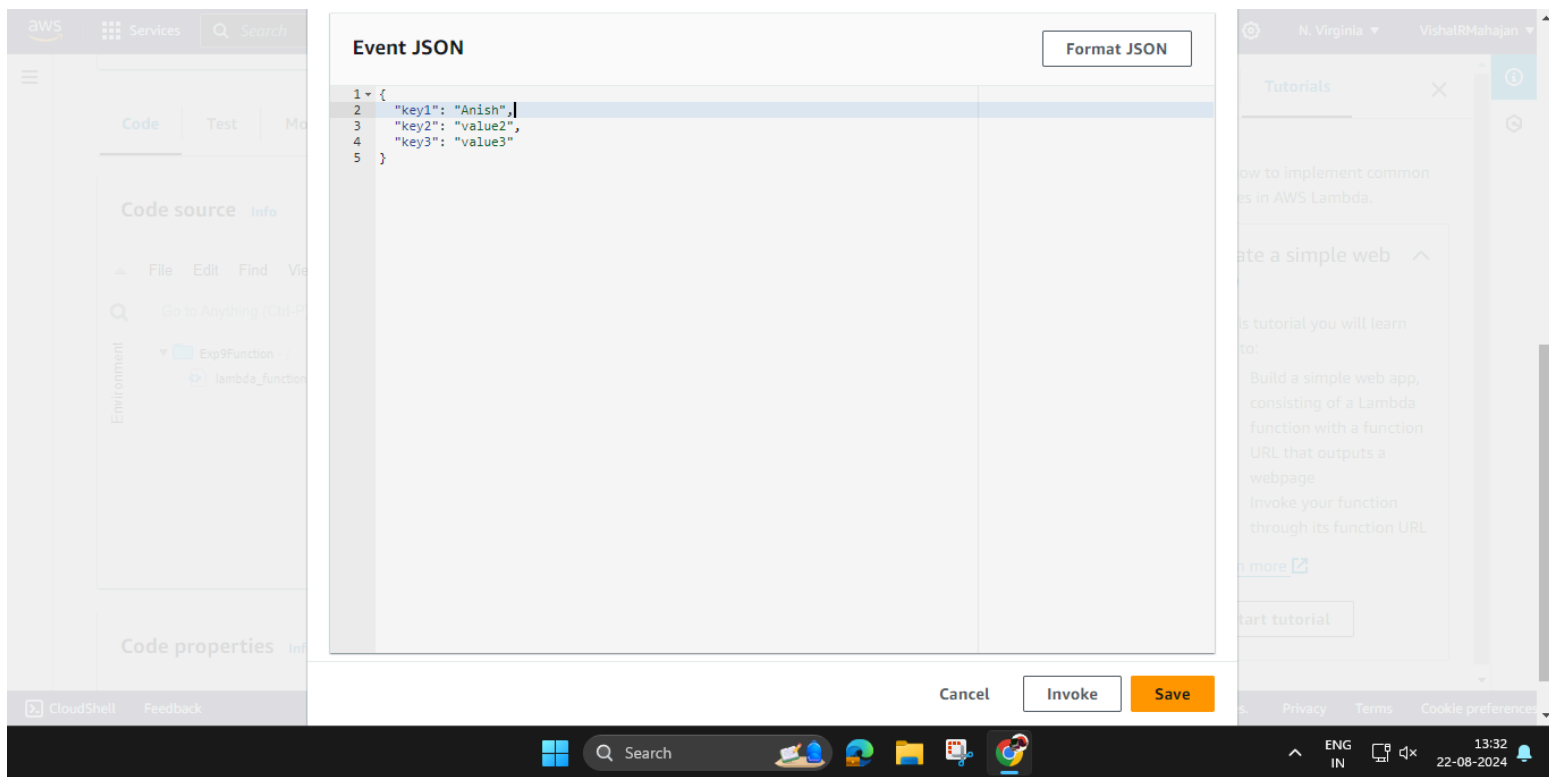
Template - optional

hello-world

Cancel Invoke Save

Name: Vishal Rajesh Mahajan  
Class: TE IT A

ADL EXP09  
Roll No: 62



Name: Vishal Rajesh Mahajan  
Class: TE IT A

ADL EXP09  
Roll No: 62

The screenshot displays the AWS Lambda console interface. At the top, a green notification bar states "The test event Exp9Event was successfully saved." Below this, the "Code source" tab is active, showing the "lambda\_function.py" file. The "Test" button is highlighted. The "Execution results" tab is also visible, showing a successful execution with a status of "Succeeded", max memory used of 30 MB, and a time of 31.16 ms. The "Response" is "Anish". The "Function Logs" section shows the following output:

```
START RequestId: 12f50c32-ca61-47be-901e-8294561a10fe Version: $LATEST
value1 = Anish
value2 = Hello World!
value3 = value3
END RequestId: 12f50c32-ca61-47be-901e-8294561a10fe
REPORT RequestId: 12f50c32-ca61-47be-901e-8294561a10fe  Duration: 31.16 ms  Billed Duration: 32 ms
```

The "Code properties" tab is also visible, showing the package size as 440 bytes and the SHA256 hash as p2KwSDthVtlecXCSBjQvSU3mhXQMcs. The last modified date is August 22, 2024 at 01:31 PM GMT+5:30. On the right side, there is a "Tutorials" panel with a section titled "Create a simple web app" and a "Start tutorial" button.

The screenshot displays the "Configure test event" dialog in the AWS Lambda console. The dialog provides information about test events and allows for configuration. It states: "A test event is a JSON object that mocks the structure of requests emitted by AWS services to invoke a Lambda function. Use it to see the function's invocation result." It also explains: "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 changes."

The "Test event action" section has two options: "Create new event" (unselected) and "Edit saved event" (selected). The "Event name" is set to "Exp9Event". There are "Invoke" and "Delete" buttons.

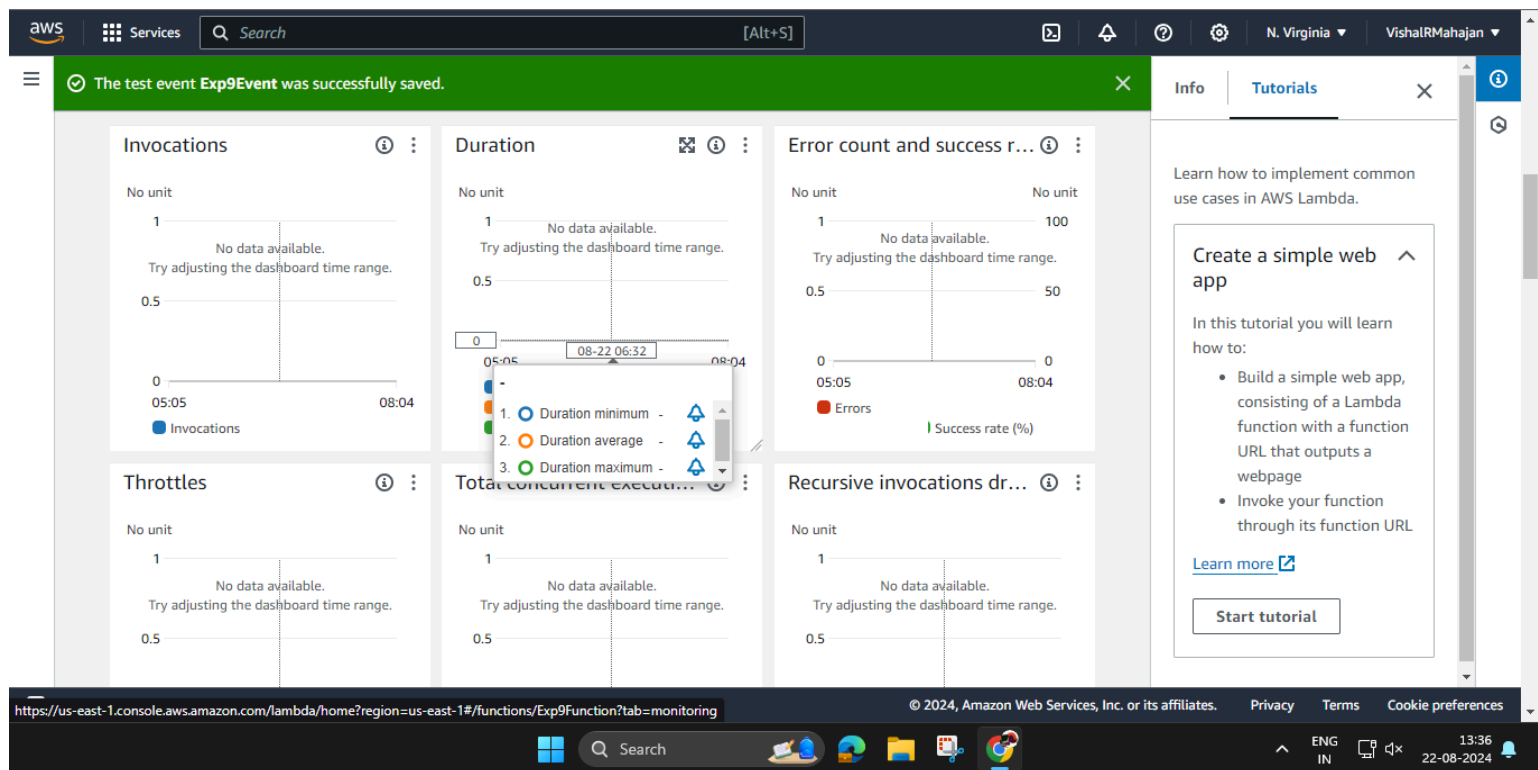
The "Event JSON" section shows the following JSON structure:

```
{
  "key1": "Anish",
  "key2": "Hello World!",
  "key3": "value3"
}
```

A "Format JSON" button is available next to the JSON editor. At the bottom of the dialog, there are "Cancel", "Invoke", and "Save" buttons.

Name: Vishal Rajesh Mahajan  
Class: TE IT A

ADL EXP09  
Roll No: 62



The screenshot shows the 'Configure test event' dialog box in the AWS Lambda console. It provides information about test events and offers options to create or edit them. The 'Event name' is set to 'Exp9Event'. The 'Event JSON' field contains the following content:

```
1 {  
2   "key1": "Anish",  
3   "key2": "Hello World!",  
4   "key3": "value3"  
5 }
```

At the bottom of the dialog, there are 'Cancel', 'Invoke', and 'Save' buttons. The background shows a blurred view of the Lambda console interface, including the 'Code source' and 'Code properties' tabs.

Name: Vishal Rajesh Mahajan  
Class: TE IT A

ADL EXP09  
Roll No: 62

The screenshot displays the AWS Lambda console interface. The top navigation bar includes the AWS logo, 'Services' menu, a search bar, and user information for 'N. Virginia' and 'VishalRMahajan'. The main content area is divided into two panels. The left panel, titled 'Code source', shows a file explorer with 'Exp9Function' and 'lambda\_function.py'. Below it, the 'Code properties' section lists 'Package size' as 440 bytes, 'SHA256 hash' as p2KwSDthVtlecXCSBjQvSU3mhXQMcsjEPSyevRn5fml=, and 'Last modified' as August 22, 2024 at 01:31 PM GMT+5:30. The right panel, titled 'Execution results', shows a failed execution with a status of 'Failed', 'Max memory used: 31 MB', and 'Time: 17.91 ms'. The 'Response' section displays a JSON object with an error message: 'key1', error type: 'KeyError', request ID: 'd081af06-55c9-4056-b5e5-4d9ef178b243', and a stack trace indicating the error occurred in 'lambda\_handler' at line 8. A 'Function Logs' section is also visible. On the far right, a 'Tutorials' sidebar offers a guide on 'Create a simple web app'.

This screenshot shows the AWS Lambda console with the 'CloudWatch metrics' section active. The top navigation bar is consistent with the previous image. The main content area features a 'Filter metrics by' dropdown set to 'Function'. Below this, there are several charts: 'Invocations' (a line chart showing a count of 2), 'Duration' (a line chart showing milliseconds with a tooltip for '2024-08-22 08:10 UTC' displaying '1.44'), 'Error count and s...' (a line chart showing a count of 2), 'Throttles' (a line chart showing a count of 0), and 'Total concurrent executi...' (a line chart showing a count of 0). A 'Rec' (Recent) section is also visible. On the right, the same 'Tutorials' sidebar is present. The bottom of the screen shows a Windows taskbar with a search bar and various application icons.