

Deploying AWS Lambda Step-by-Step

Create, Configure, and Test Your First Function

Author: Abhishek Gupta

What is AWS Lambda?

- AWS Lambda is a serverless compute service.
- – Runs code without provisioning servers
- – Triggers from events (e.g., S3, API Gateway)
- – Supports multiple languages (Python, Node.js, etc.)



Prerequisites

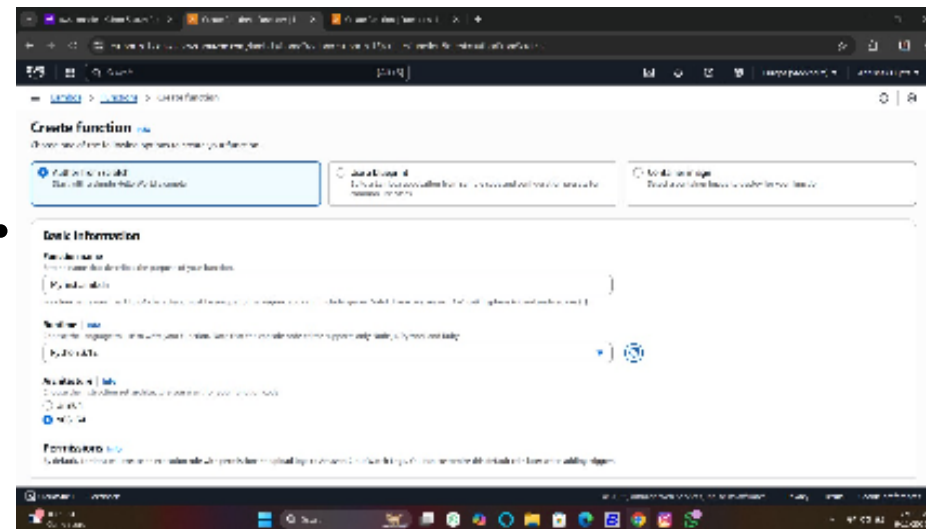
- - AWS Account
- - Basic knowledge of programming (e.g., Python, Node.js)
- - IAM Permissions to access Lambda and CloudWatch

Step 1 – Log in to AWS Console

- Go to: <https://console.aws.amazon.com>
- 1. Sign in with your AWS credentials
- 2. Navigate to AWS Lambda from the Services menu

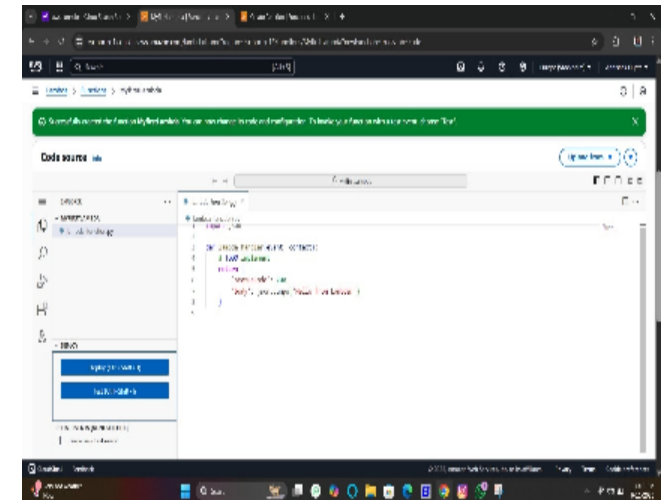
Step 2 – Create a Lambda Function

- Click Create Function
 - – Choose: Author from scratch
 - – Name: MyFirstLambda
 - – Runtime: e.g., Python 3.
 - – Execution Role:
 - – Choose existing or
 - – Create new role with basic Lambda permissions



Step 3 – Write Your Code

- Inline Editor:
- `def lambda_handler(event, context):`
- `return {`
- `'statusCode': 200,`
- `'body': 'Hello from Lambda!'`
- `}`
- – Click Deploy to save the code

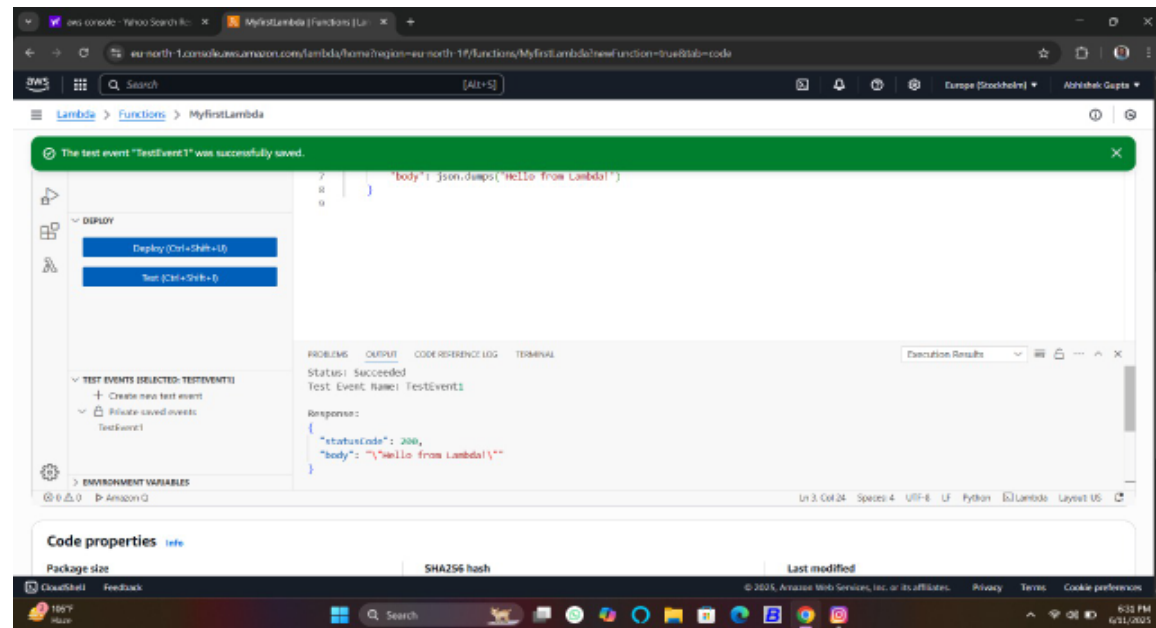


Step 4 – Configure a Test Event

- Click Test
 - – Create a new test event (use default or add sample JSON)
 - – Name: TestEvent1
 - – Click Test again to run your Lambda

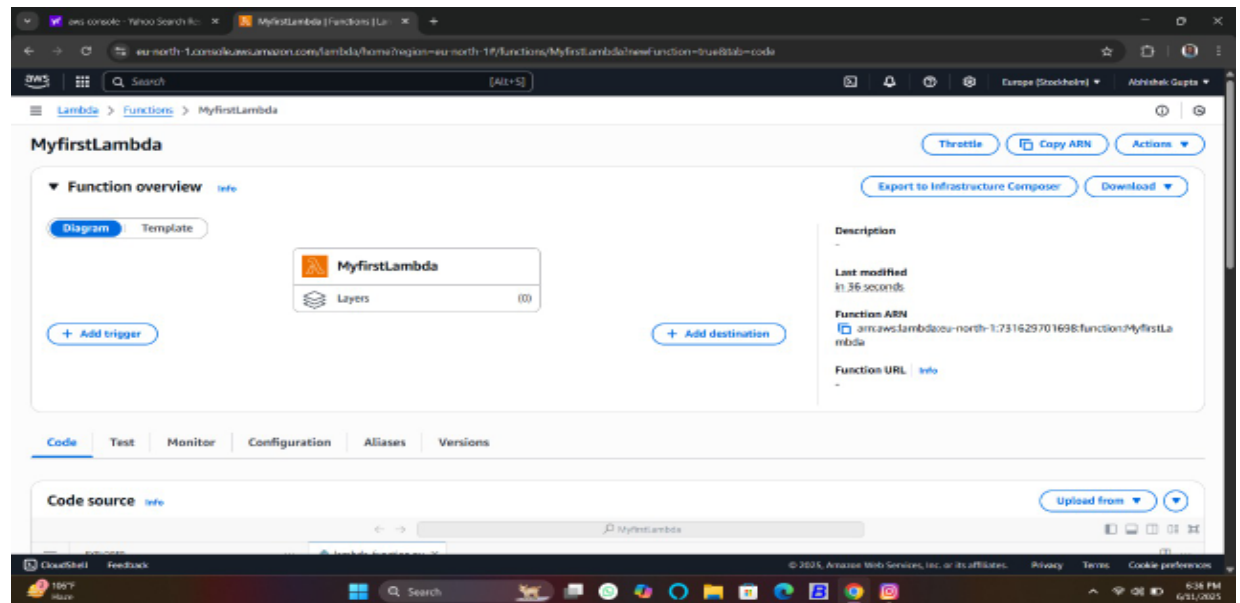
Step 5 – Check the Output

- View:
 - – Execution result
 - – Log output (CloudWatch)
 - – Response, duration, memory used



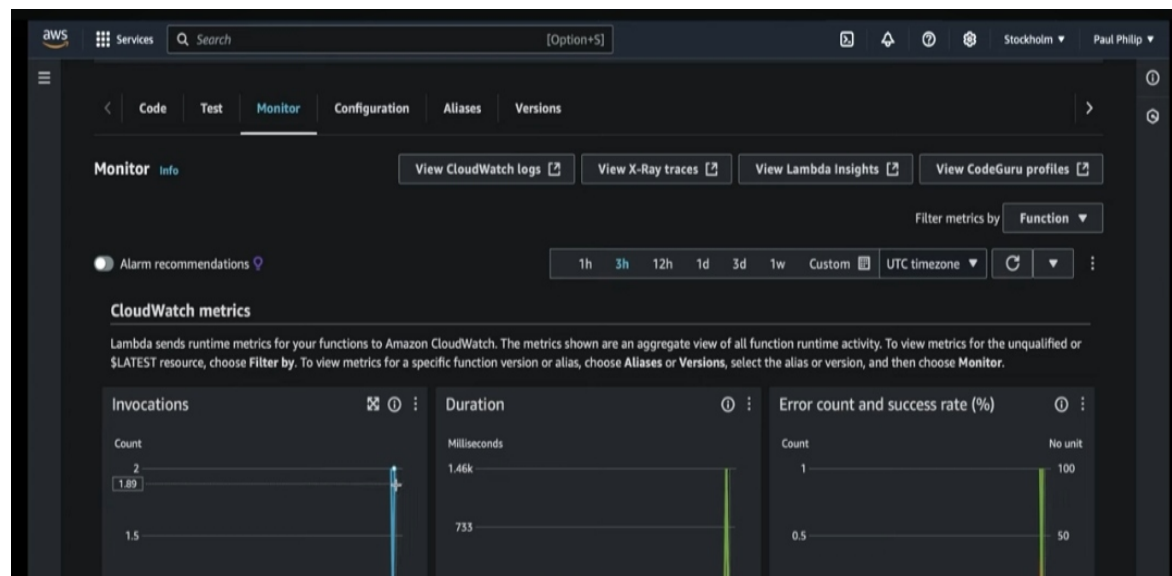
Step 6 – Trigger Setup (Optional)

- Add Trigger (e.g., S3, API Gateway)
- – Go to the Configuration > Triggers
- – Click Add Trigger
- – Choose a service and configure details



Step 7 – Monitor with CloudWatch

- Logs are auto-sent to CloudWatch
- – Go to Monitor tab
- – Or open CloudWatch > Logs
- – View function logs and metrics



Summary

- - Created Lambda function
- - Deployed simple code
- - Tested and viewed logs
- - (Optional) Set up triggers

Next Steps

- Try real use cases:
 - - Integrate with API Gateway
 - - Connect to DynamoDB or S3
 - - Add environment variables, layers

