AWS Lambda User Guide

Overview

This guide walks you through creating, testing, monitoring, and managing AWS Lambda functions using Python 3.13 runtime.

Prerequisites

- AWS Account with appropriate permissions
- Access to AWS Console

Table of Contents

Creating a Lambda Function

Step 1: Navigate to Lambda Service

1. Search for "Lambda" in the AWS Console search bar
2. Click on Lambda or navigate to: `https://eu-north-1.console.aws.amazon.com/lambda/home?region=eu-north-1#/functions`

Step 2: Create Function

1. Click **"Create function"** button
2. Direct link: `https://eu-north-1.console.aws.amazon.com/lambda/home?region=eu-north-1#/create/function?firstrun=true&intent=authorFromScratch`
3. Select **"Author from scratch"**
4. Configure function settings:

- **Function name**: `lambda-01`
- **Runtime**: Python 3.13

5. Click **"Create function"**

Step 3: Function Creation Confirmation

After creation, you'll see the function ARN:

```
arn:aws:lambda:eu-north-1:266833220666:function:lambda-01
```

Testing the Function

Initial Test

1. Navigate to the **"Test"** tab
2. The default "Hello World" Python code will be present
3. Click **"Test"** button
4. View the test results in the execution details:

```
json

{
  "statusCode": 200,
  "body": "\"Hello AWS from Lambda!\""
}
```

Creating a Test Event

1. **Event name**: `lambda-test-event-01`
2. Click **"Save"** (located next to the Test button)

> **Note**: Remember to click **"Deploy"** to save any code changes before testing.

Editing Lambda Code

Updated Lambda Function Code

Replace the default code with the following enhanced version:

python

```
import json

def lambda_handler(event, context):
    # TODO implement
    print("value1 = " + event['key1'])
    return {
        'statusCode': 200,
        'body': json.dumps({
            'message': 'Hello AWS from Lambda!',
            'event': event  # Add the event object to the returned JSON
        })
    }
```

Deploy Changes

1. After editing the code, click **"Deploy"** button
2. Wait for deployment confirmation

Creating Custom Test Events

Test Event 1: lambda-test-event-01

Default test event with standard key-value pairs:

json

```
{
  "key1": "value1",
  "key2": "value2",
  "key3": "value3"
}
```

Test Event 2: lambda-test-event-02

Custom test event:

```json
{
  "key1": "value4",
  "key2": "value5",
  "key3": "value6"
}
```

Test Results Comparison

Both test events will produce similar output structure:

lambda-test-event-01 Output:

```json
{
  "statusCode": 200,
  "body": "{\"message\": \"Hello AWS from Lambda!\", \"event\": {\"key1\": \"value1\",
}
```

lambda-test-event-02 Output:

```json
{
  "statusCode": 200,
  "body": "{\"message\": \"Hello AWS from Lambda!\", \"event\": {\"key1\": \"value4\",
}
```

Monitoring with CloudWatch

Accessing Monitor Tab

1. Navigate to the **"Monitor"** tab in your Lambda function
2. View CloudWatch metrics and performance graphs
3. Monitor function invocations, duration, and error rates

Understanding IAM Permissions

Accessing Function Permissions

1. Go to **"Configuration"** tab
2. Click **"Permissions"** on the left sidebar
3. Click on the **"Execution role"** (e.g., `lambda-01-role-4ei4tl8z`)

Default IAM Policy

The Lambda function automatically gets the following IAM policy:

Policy Name: `AWSLambdaBasicExecutionRole-4e8ce2e3-af08-4e54-94b2-f120baf1294f`

Policy JSON:

```
json

{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "logs:CreateLogGroup",
            "Resource": "arn:aws:logs:eu-north-1:266833220666:*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogStream",
                "logs:PutLogEvents"
            ],
            "Resource": [
                "arn:aws:logs:eu-north-1:266833220666:log-group:/aws/lambda/lambda-01:
            ]
        }
    ]
}
```

Policy Permissions Explained

- **logs:CreateLogGroup**: Allows creating CloudWatch log groups
- **logs:CreateLogStream**: Allows creating log streams within log groups
- **logs:PutLogEvents**: Allows writing log events to CloudWatch

---

Viewing CloudWatch Logs

Accessing CloudWatch

1. Search for "CloudWatch" in AWS Console
2. Navigate to: `https://us-east-1.console.aws.amazon.com/cloudwatch/home?region=us-east-1#home:`

Finding Lambda Logs

1. In the left menu, click **"Logs"**
2. Click **"Log groups"**
3. Find `/aws/lambda/lambda-01`
4. Direct link: `https://eu-north-1.console.aws.amazon.com/cloudwatch/home?region=eu-north-1#logsV2:log-groups/log-group/$252Faws$252Flambda$252Flambda-01`
5. Click on **"Log streams"** to view detailed execution logs

---

Cleanup and Deletion

Important Notes About Package Types

- **Package type: Zip** - Code is packaged as a ZIP file (default editor mode)
- **Package type: Image** - Uses container images (not Docker files, but Lambda-specific container images)

Deleting the Lambda Function

1. Go to Lambda function dashboard
2. Select your function
3. Click **"Actions"** → **"Delete function"**

> **Warning**: Deleting a function permanently removes the function code. Related logs, roles, test event schemas, and triggers are retained in your account.

Complete Cleanup Process

1. Delete CloudWatch Log Groups

1. Navigate to CloudWatch
2. Go to **"Log groups"**
3. Select `/aws/lambda/lambda-01`

4. Click **"Actions"** → **"Delete log group"**

2. Delete IAM Roles and Policies

1. Search for "IAM" in AWS Console
2. Navigate to **"Roles"**
3. Find and delete the Lambda execution role
4. Navigate to **"Policies"**
5. Find and delete associated Lambda policies

3. Check Billing and Free Tier Usage

1. Search for "Billing and Cost Management"
2. Navigate to **"Free tier"**
3. Direct link: `https://us-east-1.console.aws.amazon.com/billing/home?region=us-east-1#/freetier`
4. Review all billing information and usage

Best Practices

1. **Always deploy code changes** before testing
2. **Use descriptive names** for functions and test events
3. **Monitor CloudWatch logs** for debugging
4. **Clean up resources** to avoid unnecessary charges
5. **Review IAM permissions** regularly for security
6. **Test with multiple event scenarios** to ensure robustness

---

Troubleshooting

Common Issues

- **Function not updating**: Ensure you clicked "Deploy" after code changes
- **Test failures**: Check CloudWatch logs for detailed error messages
- **Permission errors**: Verify IAM roles and policies are correctly configured
- **Timeout errors**: Increase function timeout in Configuration settings

Getting Help

- Check CloudWatch logs for detailed error information
- Review AWS Lambda documentation
- Use AWS Support for complex issues