

Practical: Serverless with AWS Lambda

Table of Contents

- [Activity: connect AWS Lambda with AWS S3 using Node.js](#)
 - [Walk-through video](#)
 - [Step 1: Create an S3 Bucket](#)
 - [Step 2: Create an AWS Lambda Function](#)
 - [Step 3: Add S3 Trigger to Lambda](#)
 - [Step 4: Write the Lambda Function](#)
 - [Step 5: Test the Lambda Function](#)
- [Use in your project](#)
 - [Connecting to resources in the VPC](#)

Activity: connect AWS Lambda with AWS S3 using Node.js

In this activity, you will learn how to trigger an AWS Lambda function when a file is uploaded to an S3 bucket.

This is a simple demonstration of reacting to an event that, perhaps, a user initiated.

Walk-through video

CAB432 Lambda Practical


 Search

[Hide transcript](#)

Hello. Welcome to the practical for AWS Lambda, which is the AWS service for serverless functions. So what we're going to do is demonstrate the basics of Lambda by creating a small function, which will be triggered whenever an item is added to an S3 bucket and then it will just print some information into the logs. So it's pretty

Step 1: Create an S3 Bucket

Create an S3 bucket, if you do not have one already.

If you need help, go back to the practical where we create an S3 bucket for the first time:

Practical: S3 blob storage service (<https://canvas.qut.edu.au/courses/20367/pages/practical-s3-blob-storage-service-javascript>)

Step 2: Create an AWS Lambda Function

1. Navigate to Lambda in the AWS Management Console.
2. Click Create function and choose the following:
 - Author from scratch.
 - Function name: `n123456789-prac10-01`.
 - Runtime: Node.js 22.x
 - **IMPORTANT:**
 - Under *Change default execution* select *Use and existing role*

- Under *Existing role* select **CAB432-Lambda-RoLe**
3. Add tags:
 - Under *Additional configurations*:
 - Under *Tags* tick *Enable*
 - Add tags for **qut-username** set to your QUT username like **n1234567@qut.edu.au** and key **purpose** set to **practical**
 4. Click Create function.
 5. Dismiss the dialogue.

Step 3: Add S3 Trigger to Lambda

1. Navigate to the Lambda function you just created.
2. Scroll down to the Function overview section.
3. Click Add trigger and:
 - Choose S3 as the trigger source.
 - Select the bucket you created earlier.
 - For Event type, choose **All object create events**.
4. Read the note about *Recursive invocation* and tick to acknowledge.
5. Click Add.

Step 4: Write the Lambda Function

1. Download the sample code: [lambda_practical.zip](https://canvas.qut.edu.au/courses/20367/files/7022404/download)
(<https://canvas.qut.edu.au/courses/20367/files/7022404/download>).
2. Go to the Code tab of your Lambda function.
3. In the top right of the code editor, click *Upload from*, then *.zip file*
4. Select the zip file that you downloaded previously.
5. Click *Save*
6. On the left, click *Deploy*

You should now see the sample code in the code editor.

Alternatively, you can copy/paste the code into the editor.

Step 5: Test the Lambda Function

1. Upload a file to your S3 bucket via the AWS Console or AWS CLI:
 - Navigate to the S3 bucket you created.
 - Click Upload and choose a file (e.g., a simple text file).
2. After uploading, check the CloudWatch Logs for your Lambda function:
 - Navigate to the Monitor tab of your function
 - Click View CloudWatch logs

Use in your project

In this activity, we reacted to a file being uploaded to S3.

How could you build off this? You could

- Enqueue a task to resize an image upon upload
- Check the contents of a file upon upload
- Remove associated metadata from a database when a file is deleted
- and many other possibilities

Lambdas can also be used with other services:

- Trigger by an API gateway route
- Trigger by various other events
- Create a custom metric in CloudWatch to use as an auto-scaling target metric

The `CAB432-Lambda-Role` has permissions to make calls to these AWS services:

- CloudWatch: read and write custom metric data
- DynamoDB: read and write
- S3: put, get, list and delete objects
- SQS: send, receive and delete messages
- Parameter Store: read parameters

Please take note that Lambdas are relatively expensive compared to EC2 instances or ECS containers for CPU intensive tasks or tasks which are run frequently enough to keep an EC2 instance busy. Hence they are not suitable for the CPU-intensive task associated with your project.

Connecting to resources in the VPC

Lambdas can connect to AWS services like S3 and DynamoDB without special configuration. However, they cannot connect to resources like RDS, EC2 instances or ECS containers without configuring networking for them.

When creating your Lambda, under *Additional Configurations*

1. Under *VPC*, tick the *Enable* box
2. Select the only VPC available.
3. Under *Subnets* choose a private subnet. This is sufficient for connecting to other resources on the VPC.
4. Under *Security Groups* select `CAB432SG`.

All other configuration can remain as before.