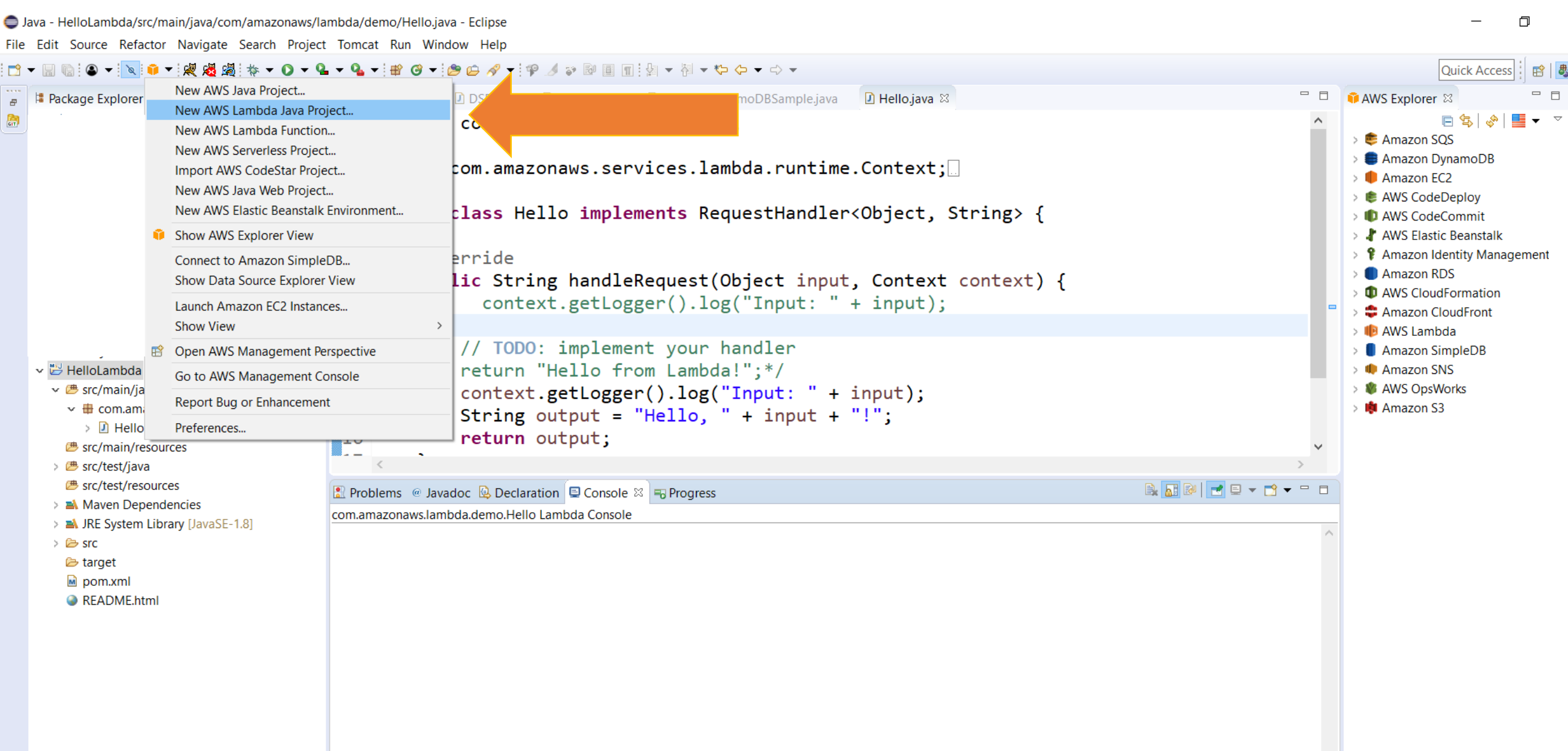


AWS-Lambda Programmatic Tutorial

Narges Mehran, MSc.

Current Topics in Distributed Systems: Internet of
Things and Cloud Computing, SS2020

How to create AWS Lambda project



How to create a handler class with Java

- Change Input type to: Custom

New AWS Lambda Maven Project

Create a new AWS Lambda Java project

A project with this name already exists.

Project name: HelloLambda

Maven configuration

Group ID: com.amazonaws.lambda

Artifact ID: demo

Version: 1.0.0

Package name: com.amazonaws.lambda.demo

Lambda Function Handler

Each Lambda function must specify a handler class which the service will use as the entry point to begin execution. [Learn more](#) about Lambda Java function handler.

Class Name: LambdaFunctionHandler

Input Type: Custom

A hello world Lambda function.

Preview:

```
package com.amazonaws.lambda.demo;

import com.amazonaws.services.lambda.runtime.Context;
import com.amazonaws.services.lambda.runtime.RequestHandler;

public class LambdaFunctionHandler implements RequestHandler<Object, String> {

    @Override
    public String handleRequest(Object input, Context context) {
        context.getLogger().log("Input: " + input);

        // TODO: implement your handler
        return "Hello from Lambda!";
    }
}
```

Properties

Finish 3 Cancel

Implement the Handler Method

- Replace the contents of the `handleRequest` function with the following code.

```
@Override
public String handleRequest(String input, Context context)
{
    context.getLogger().log("Input: " + input);
    String output = "Hello, " + input + "!";
    return output;
}
```

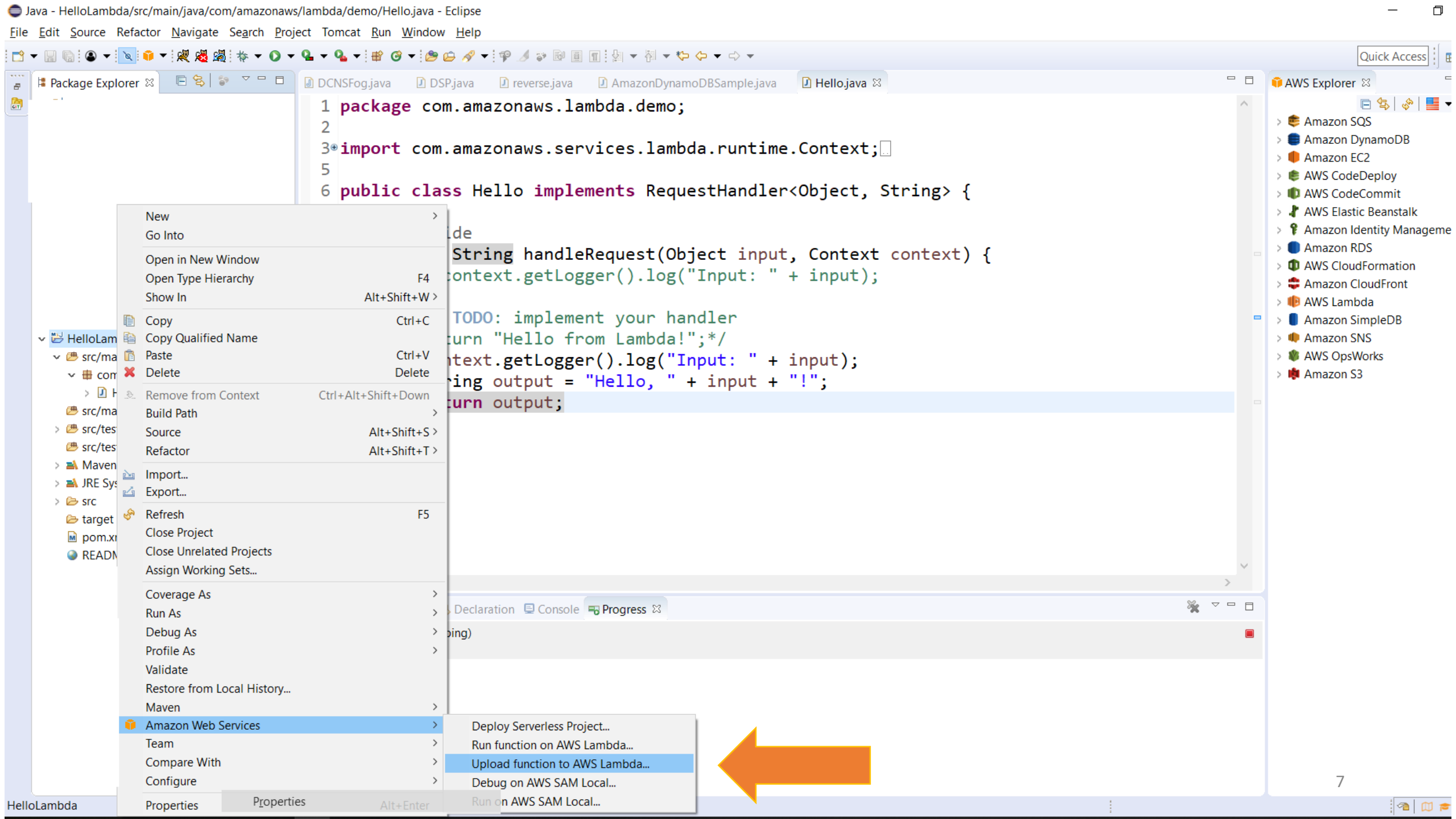
Allow Lambda to Assume an IAM Role

To create an IAM role for Lambda

- Sign in to the AWS Management Console.
- From the Services menu, open the IAM console.
- In the Navigation pane, choose Roles, and then choose Create role.
- For “Select type of trusted entity”, choose AWS service, and then choose Lambda for the service that will use this role. Then choose Next: Permissions.
- For Attach permissions policy, choose *AWSLambdaBasicExecutionRole*, *AWSCodeDeployRoleForLambda*, *AmazonS3FullAccess*, *AWSLambdaInvocation-DynamoDB*. Then choose Next: Review.
- Add a name for your role, such as hello-lambda-role, and a description for the role. Then choose Create role to finish creating the IAM role.

Create an Amazon S3 Bucket for Lambda Code

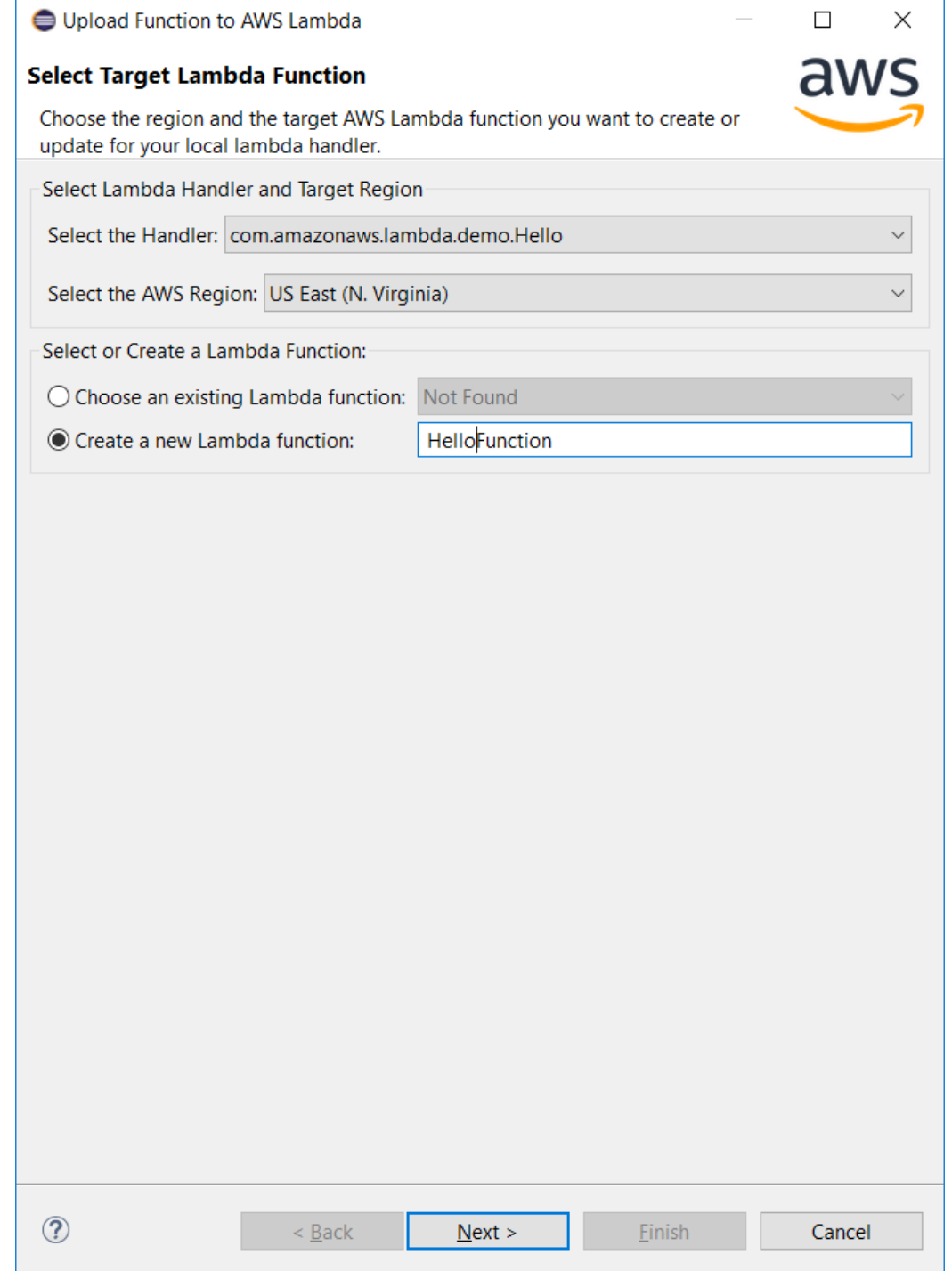
- AWS Lambda requires an Amazon S3 bucket to store your Java project when you upload it. You can either use a bucket that already exists in the AWS Region in which you'll run your code, or you can create a new one specifically for Lambda to use (recommended).
- You can create an Amazon S3 bucket in two ways, either through the AWS Management Console or by using the AWS Toolkit for Eclipse. This section describes how to create an Amazon S3 bucket in the console. See [Upload the Code](#) to create one using the AWS Toolkit for Eclipse.
- **To create an Amazon S3 bucket for use with Lambda**
 1. Sign in to the [AWS Management Console](#).
 2. From the **Services** menu, open the [S3 console](#).
 3. Choose **Create bucket**.
 4. Enter a bucket name, and then choose a region for your bucket. This region should be the same one in which you intend to run your Lambda function. For a list of regions supported by Lambda see [Regions and Endpoints](#) in the Amazon Web Services General Reference.
 5. Choose **Create** to finish creating your bucket.



How to upload the function code

Upload your function to AWS Lambda by:

- Choose *Create a new Lambda function*, and then type a name for your function (for example, *HelloFunction*).
- Click *Next* to proceed to function configuration



The screenshot shows the 'Upload Function to AWS Lambda' wizard in the AWS Management Console. The title bar at the top reads 'Upload Function to AWS Lambda'. The main heading is 'Select Target Lambda Function'. Below this, a sub-header says 'Choose the region and the target AWS Lambda function you want to create or update for your local lambda handler.' The AWS logo is in the top right corner. The form contains two main sections. The first section, 'Select Lambda Handler and Target Region', has two dropdown menus: 'Select the Handler:' with the value 'com.amazonaws.lambda.demo.Hello' and 'Select the AWS Region:' with the value 'US East (N. Virginia)'. The second section, 'Select or Create a Lambda Function:', has two radio buttons. The first is 'Choose an existing Lambda function:' with a dropdown menu showing 'Not Found'. The second is 'Create a new Lambda function:', which is selected, and has a text input field containing 'HelloFunction'. At the bottom of the form, there are four buttons: a help button with a question mark, a '< Back' button, a 'Next >' button (which is highlighted with a blue border), and a 'Finish' button. To the right of the 'Finish' button is a 'Cancel' button.

Upload Function to AWS Lambda

Select Target Lambda Function

Choose the region and the target AWS Lambda function you want to create or update for your local lambda handler.

Select Lambda Handler and Target Region

Select the Handler: com.amazonaws.lambda.demo.Hello

Select the AWS Region: US East (N. Virginia)

Select or Create a Lambda Function:

☐ Choose an existing Lambda function: Not Found

☒ Create a new Lambda function: HelloFunction

? < Back Next > Finish Cancel

Function Configuration

- Enter description of your function, select *IAM role* if you don't have student account.
- Select the *S3 bucket* to upload the function into that.
- Note: if needed change the advance resource settings

Upload Function to AWS Lambda

Function Configuration

Configure this Lambda function and upload to AWS.


Basic Settings

Name: HelloFunction

Description: Say Hello to someone

Function Role

Select the IAM role that AWS Lambda can assume to execute the function on your behalf. [Learn more](#) about Lambda execution roles.

IAM Role: hello-lambda-role  Create

Function Versioning and Alias

You can publish a new immutable version and an alias to that version whenever you have a new revision of the Lambda function. [Learn more](#) about Lambda function versioning and aliases.


☐ Publish new version

☐ Provide an alias to this new version

☐ Choose an existing function alias: Not Found

☒ Create a new function alias: beta

S3 Bucket for Function Code

S3 Bucket: example-bucket-1one  Create

Upload Lambda zip file with encryption to protect data at rest by using Amazon S3 master-key or by using AWS KMS master-key. [Learn more](#) about Amazon S3 encryption.


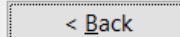
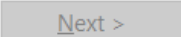
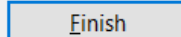
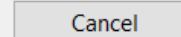
☒ None ☐ Amazon S3 master-key ☐ AWS KMS master-key

KMS Key: Error Create

Advanced Settings

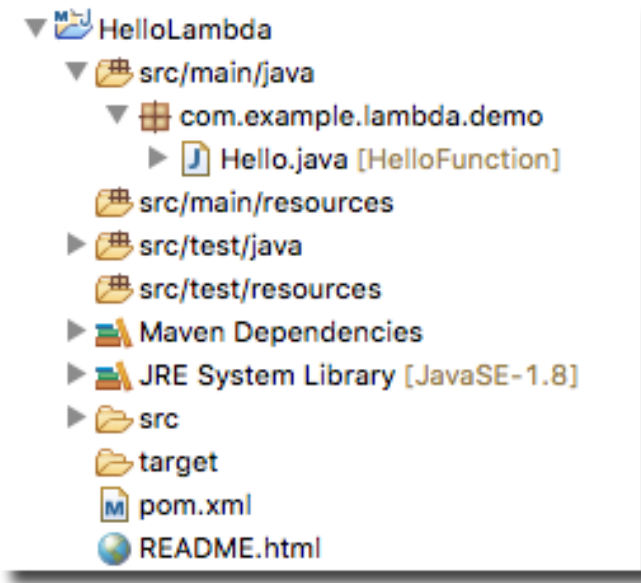
Memory (MB): 512

Timeout (s): 15

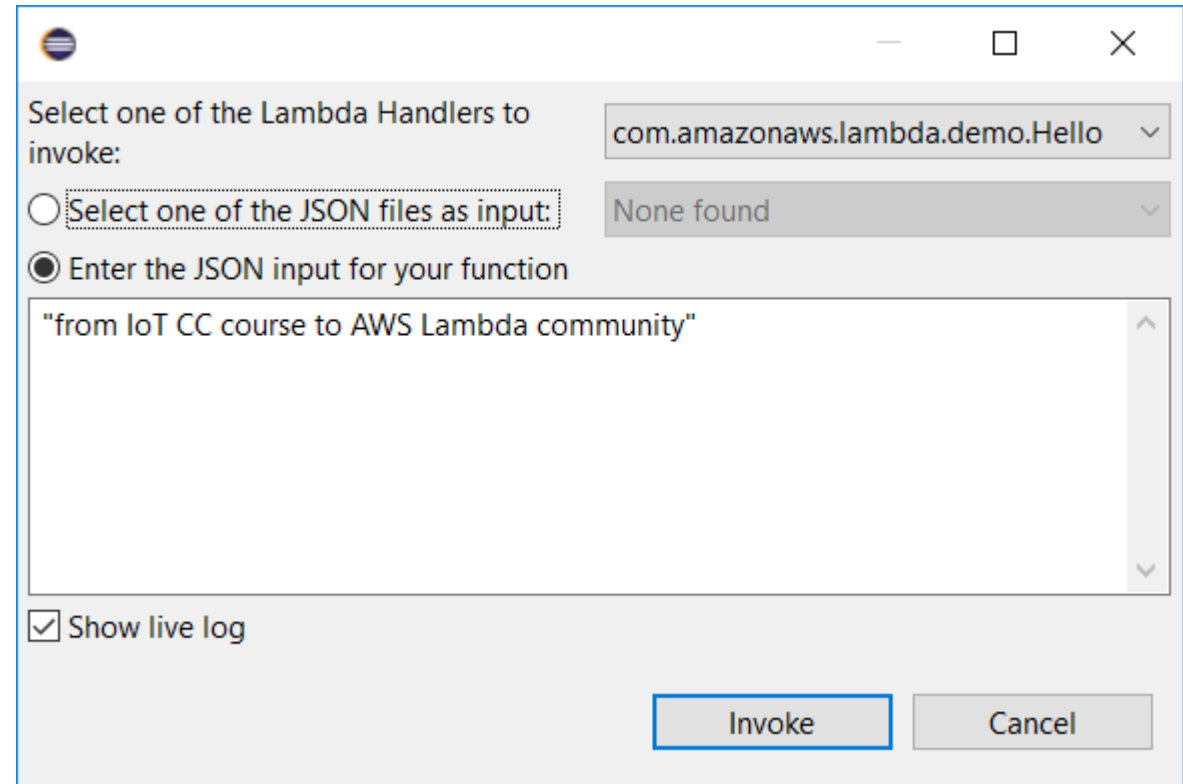
What if the upload succeeds?

- If the upload succeeds, you will see the Lambda function name that you chose appear next to your Java handler class in the **Project Explorer** view.



Invoke the Lambda Function

- Right-click in the Eclipse code window, choose *Amazon Web Services*, and then choose *Run Function on AWS Lambda*.
- Choose the handler class you want to invoke.
- Then type a valid JSON string, such as “IoT-CC course AWS Lambda community”.



Lambda function output in the Eclipse console

Invoking function...

===== FUNCTION OUTPUT =====

"Hello, from IoT CC course to AWS Lambda community!"

===== FUNCTION LOG OUTPUT =====

START RequestId: 28058fa0-582c-4c3a-8b46-70aa873e98b8 Version: \$LATEST

Input: from IoT CC course to AWS Lambda community

END RequestId: 28058fa0-582c-4c3a-8b46-70aa873e98b8

REPORT RequestId: 28058fa0-582c-4c3a-8b46-70aa873e98b8Duration: 0.62 ms

Billed Duration: 100 ms

Memory Size: 512 MB

Max Memory Used: 76 MB

Thanks for attention