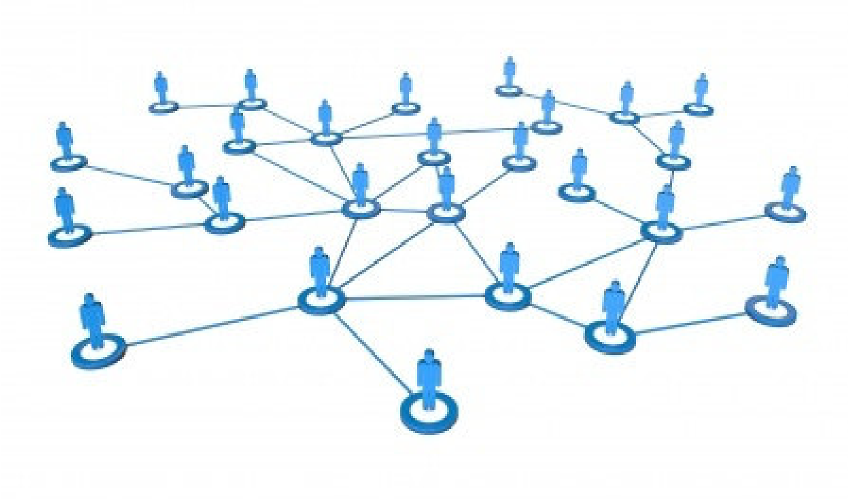
ÖZYEĞİN ÜNİVERSİTESİ

CS550-450: Distributed Systems and Cloud Computing

Professor: Dr Ismail Ari
TA: Muhammad Kashif



Lab 2

How to create, upload and invoke AWS Lambda function using Eclipse IDE

Learning Outcome

- To learn how to set up Tool Kit (Eclipse IDE) for creating and deploying AWS Lambda Function
- To learn how to setup AWS credential and used it with Eclipse IDE
- To learn how to create AWS Lambda project
- To learn how to create an Amazon S3 Bucket for Lambda function
- To learn how to invoke the Lambda function

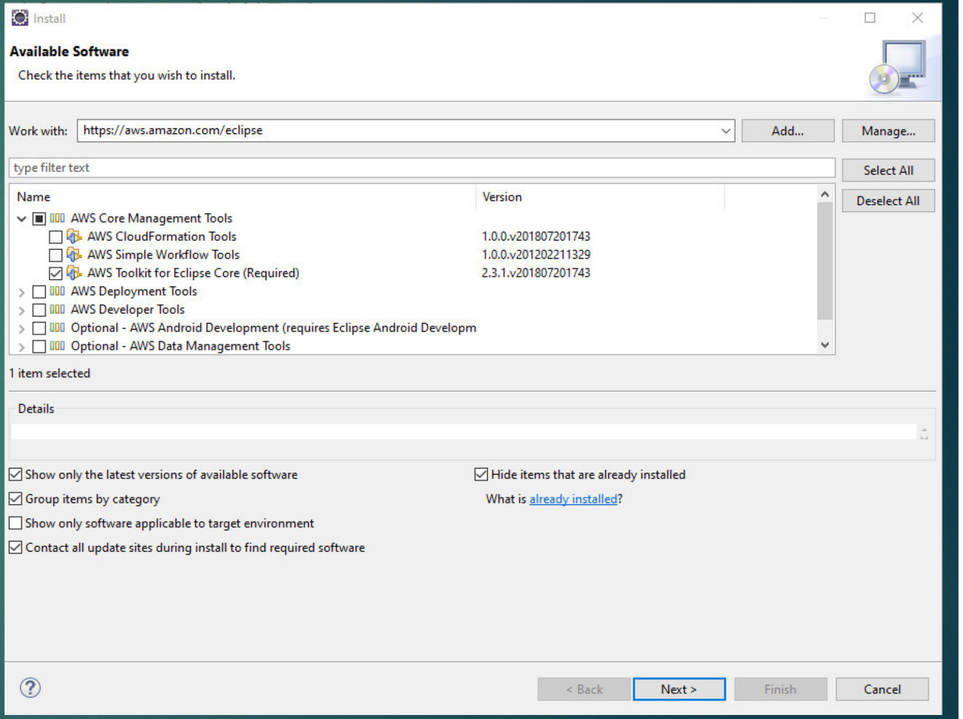
Getting Started

Step 1: Setup Tool Kit (Eclipse IDE)

- Set up the Toolkit (Eclipse IDE)
- Download Java 1.8 or latest
- Dlownlaod Eclipse IDE for Java Developers 4.2 or later

Step 2: To install the AWS Toolkit for Eclipse

- Within Eclipse, click **Help** and then click **Install New Software**.
- In the **Work with** box, type https://aws.amazon.com/eclipse and press Enter.
- Select **AWS Toolkit for Eclipse Core**, it is **required** and select **AWS Lambda Plugin** from **AWS Deployment Tools**. Other components are optional.
- Click Next or Finish to complete installation. (As shown in Fig)



Available Software

Check the items that you wish to install.

Work with: https://aws.amazon.com/eclipse

type filter text

Name	Version
■ AWS Core Management Tools	
□ AWS CloudFormation Tools	1.0.0.v201807201743
□ AWS Simple Workflow Tools	1.0.0.v201202211329
☑ AWS Toolkit for Eclipse Core (Required)	2.3.1.v201807201743
□ AWS Deployment Tools	
□ AWS Developer Tools	
□ Optional - AWS Android Development (requires Eclipse Android Developm	
□ Optional - AWS Data Management Tools	

1 item selected

Details

☑ Show only the latest versions of available software

☑ Group items by category

□ Show only software applicable to target environment

☑ Contact all update sites during install to find required software

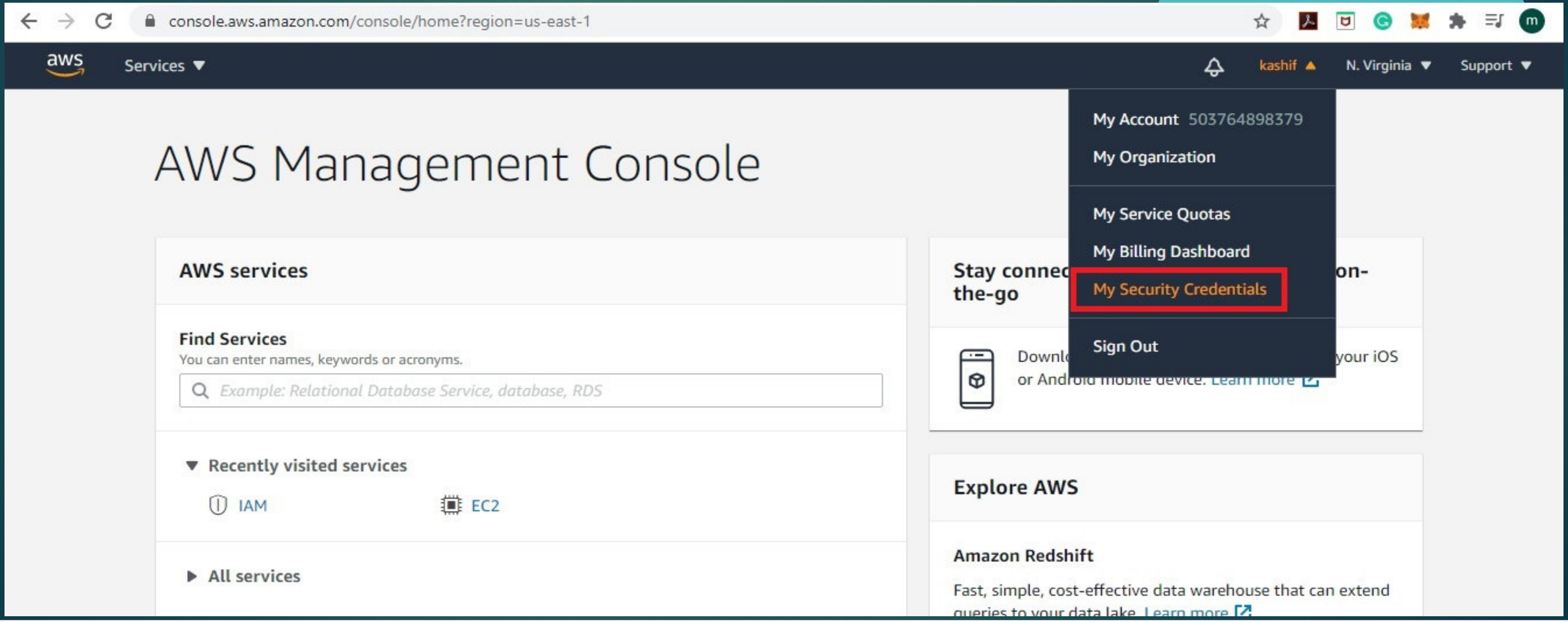
☑ Hide items that are already installed

What is already installed?

Getting Started

Step 3: Setup AWS Credentail to get access key ID and secret access key

- Open the IAM Console: https://console.aws.amazon.com/iam/home
- On the navigation menu, choose Users.
- Open the **Security credentials** tab, and then choose **Create access key**



AWS Management Console

AWS services

You can enter names, keywords or acronyms.

Stay connec... on-the-go

Downl... your iOS or Android mobile device. Learn more

Explore AWS

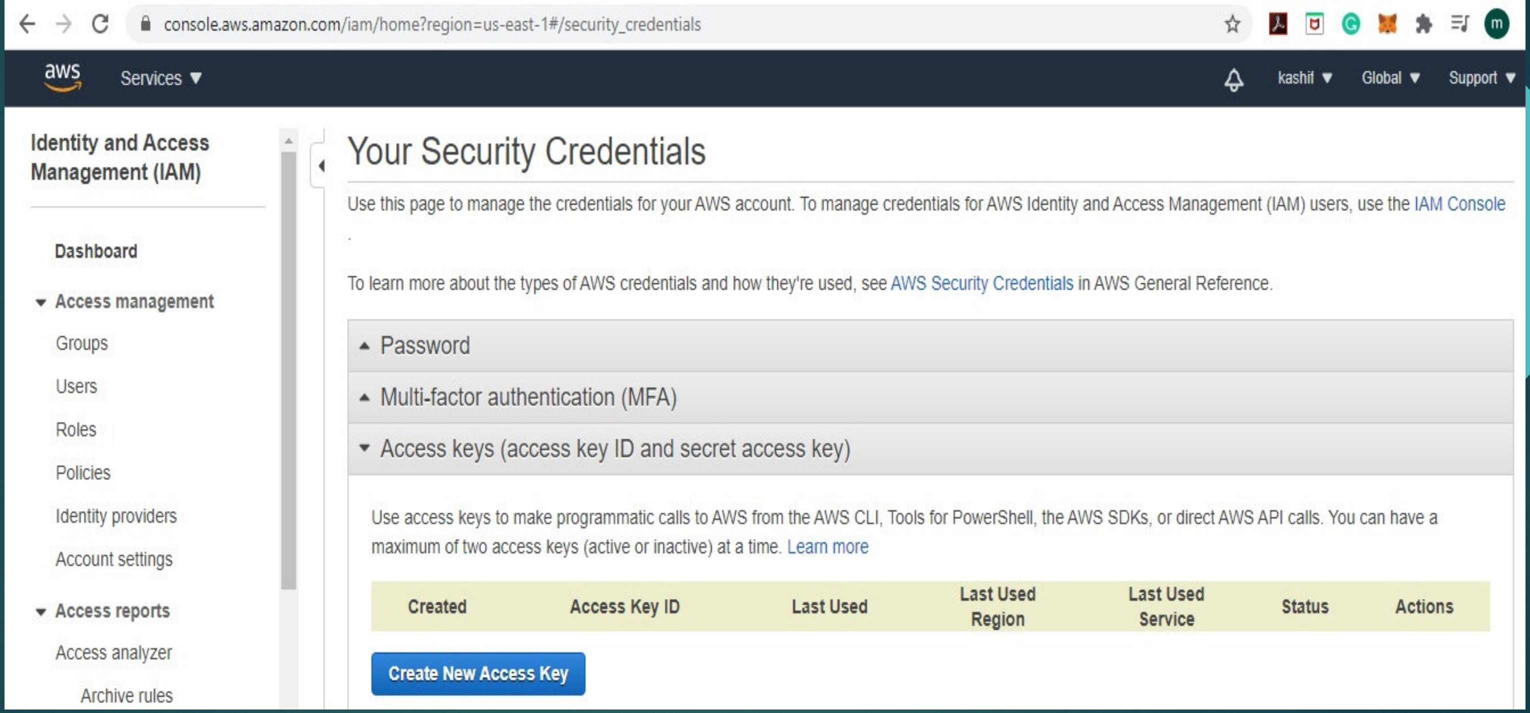
Amazon Redshift

Fast, simple, cost-effective data warehouse that can extend queries to your data lake. Learn more

Getting Started

Step 3: Setup AWS Credentail

- To create the new access key, choose **New Access key tab.**



Your Security Credentials

Use this page to manage the credentials for your AWS account. To manage credentials for AWS Identity and Access Management (IAM) users, use the IAM Console .

To learn more about the types of AWS credentials and how they're used, see AWS Security Credentials in AWS General Reference.

- Password
- Multi-factor authentication (MFA)
- Access keys (access key ID and secret access key)

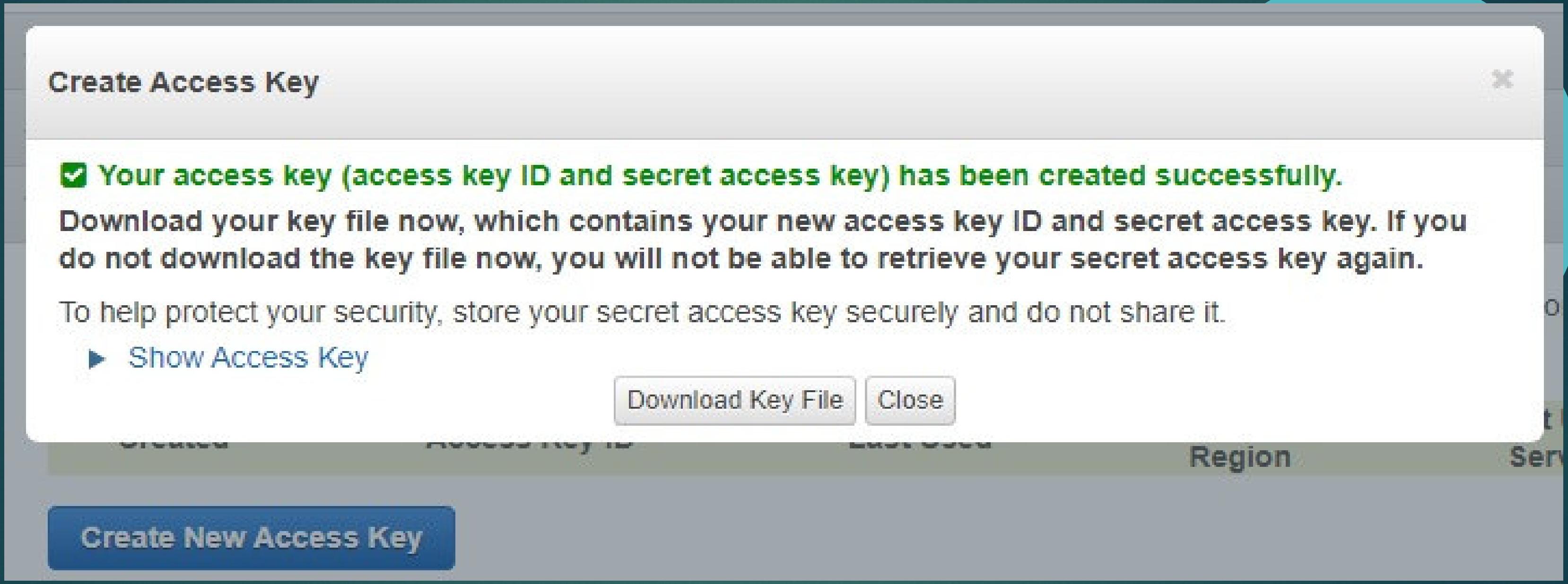
Use access keys to make programmatic calls to AWS from the AWS CLI, Tools for PowerShell, the AWS SDKs, or direct AWS API calls. You can have a maximum of two access keys (active or inactive) at a time. Learn more

Created	Access Key ID	Last Used	Last Used Region	Last Used Service	Status	Actions

Getting Started

Step 3: Setup AWS Credentail

To download the key pair, choose **Download .csv file**. Store the keys



Create Access Key

☑ Your access key (access key ID and secret access key) has been created successfully.

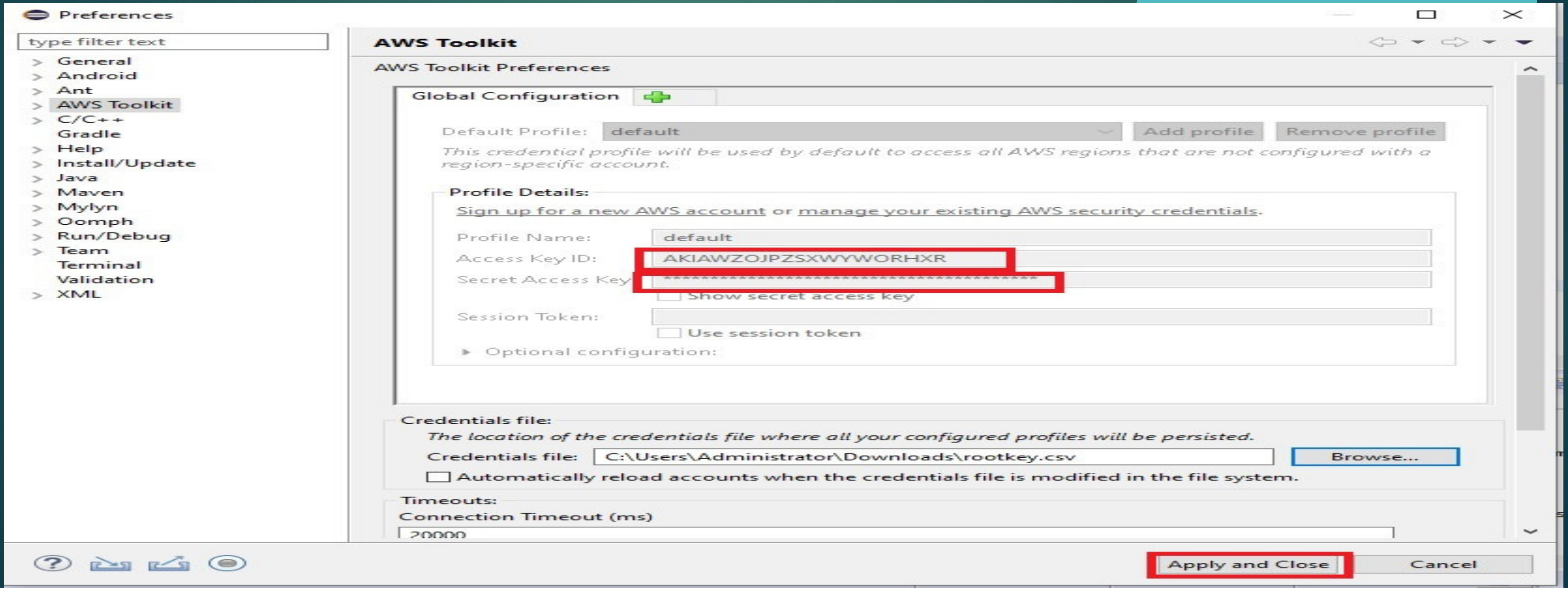
Download your key file now, which contains your new access key ID and secret access key. If you do not download the key file now, you will not be able to retrieve your secret access key again.

To help protect your security, store your secret access key securely and do not share it.

Getting Started

Step 4: To add your access keys to the AWS Toolkit for Eclipse

- Open Eclipse's **Window > Preferences** dialog box and click **AWS Toolkit** in the sidebar.
- Type or paste your AWS access key ID in the **Access Key ID** box.
- Type or paste your AWS secret access key in the **Secret Access Key** box.
- Click **Apply** or **OK** to store your access key information



AWS Toolkit

AWS Toolkit Preferences

Global Configuration

Default Profile: default

This credential profile will be used by default to access all AWS regions that are not configured with a region-specific account.

Profile Details:

Sign up for a new AWS account or manage your existing AWS security credentials.

Profile Name: default

Access Key ID: AKIAWZOJPZSXWYWORHXR

Secret Access Key: **

□ Show secret access key

Session Token:

□ Use session token

Optional configuration:

Credentials file:

The location of the credentials file where all your configured profiles will be persisted.

Credentials file: C:\Users\Administrator\Downloads\rootkey.csv

□ Automatically reload accounts when the credentials file is modified in the file system.

Timeouts:

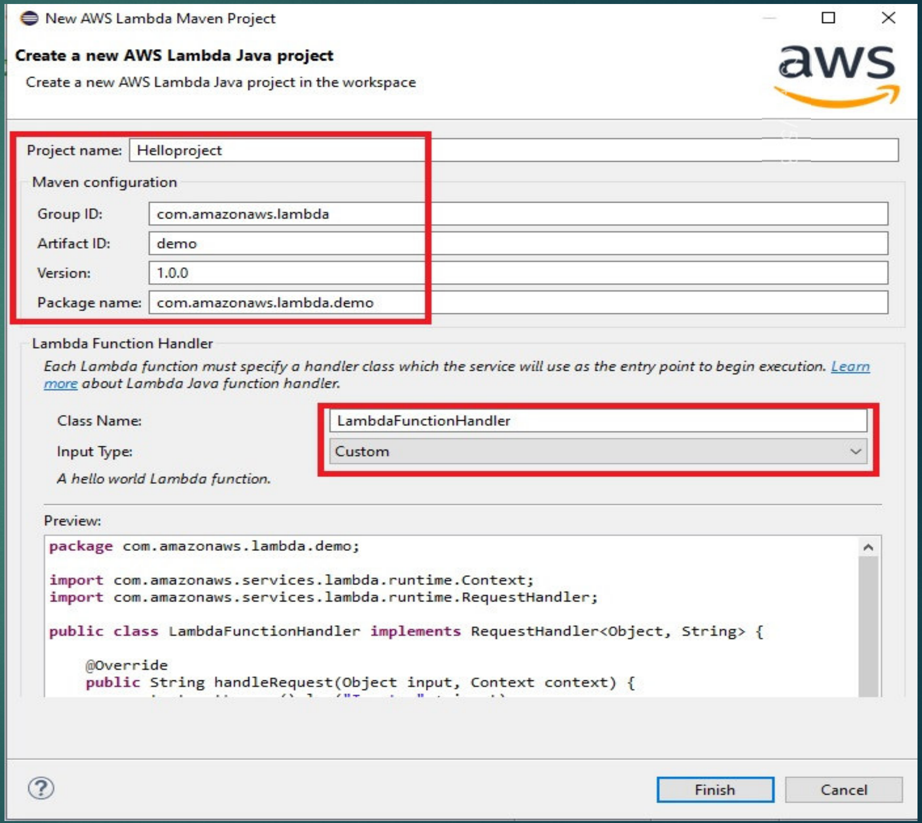
Connection Timeout (ms)

20000

Create AWS Lambda Project

Step 5: To create an AWS lambda project in Eclipse

- On the Eclipse toolbar, open the Amazon Web Services menu (identified by the AWS homepage icon), and then choose **New AWS Lambda Java project**. Or on the Eclipse menu bar, choose **File**, **New**, **AWS Lambda Java Project**.
- Add a *Project name*, *Group ID*, *Artifact ID*, and *class name* in the associated input boxes.
- After you choose **Finish**, your project's directory and source files are generated in your Eclipse workspace



Create a new AWS Lambda Java project

Create a new AWS Lambda Java project in the workspace

Project name: Helloproject

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.

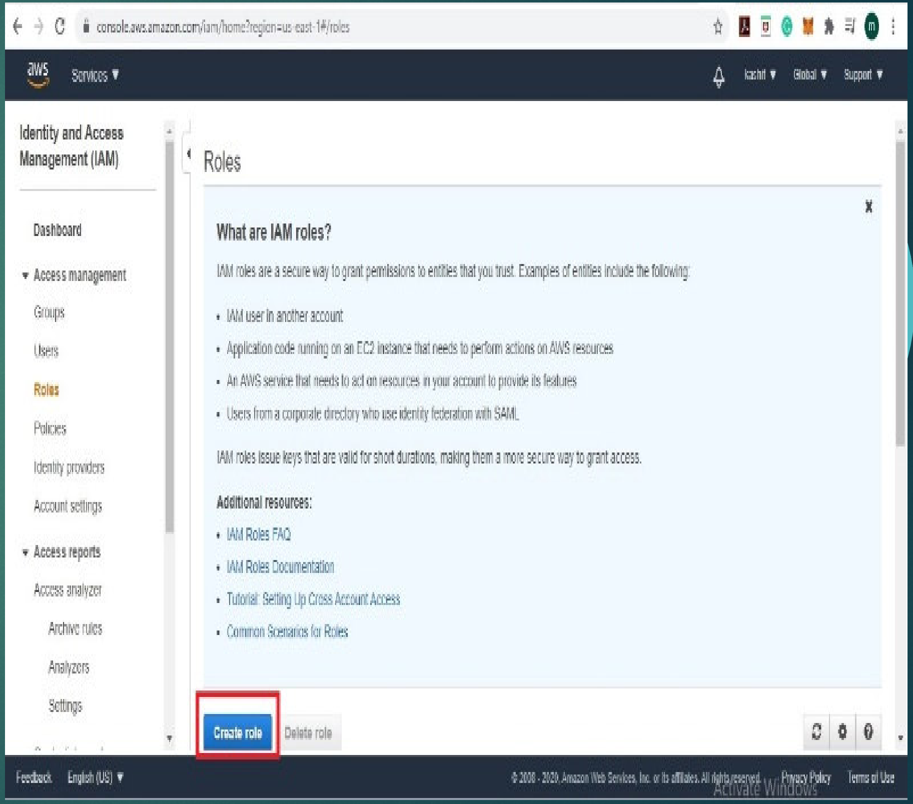
Preview:

```
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) {
```

Create AWS Lambda Project

Step 6: **To create an IAM role for Lambda on AWS console**

- 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 **AWSLambda BasicExecution Role**. Then choose **Next: Review**.



What are IAM roles?

IAM roles are a secure way to grant permissions to entities that you trust. Examples of entities include the following:

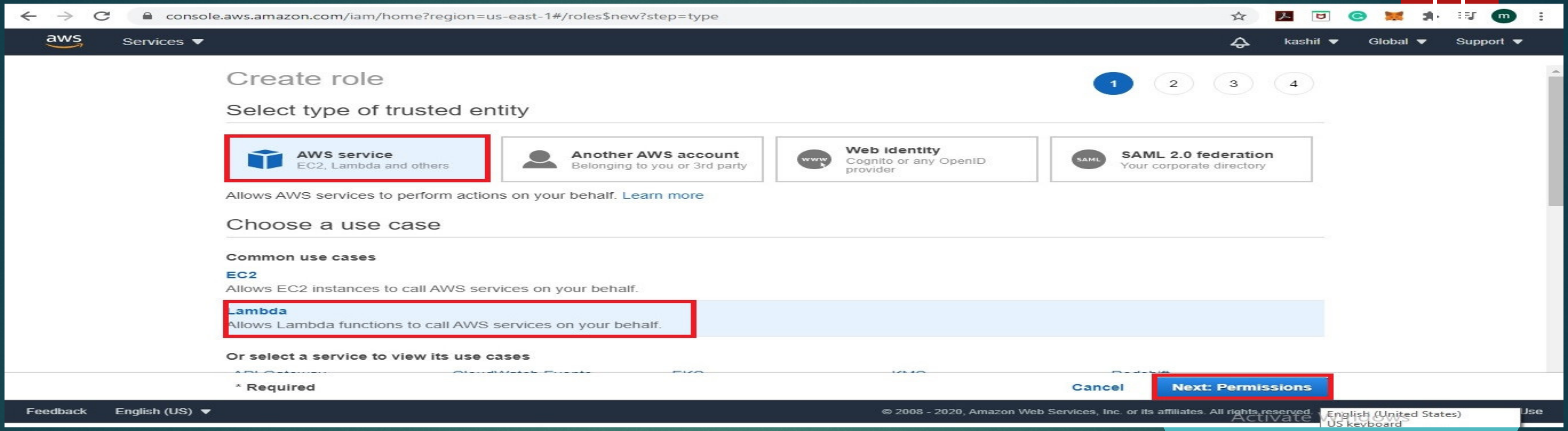
- IAM user in another account
- Application code running on an EC2 instance that needs to perform actions on AWS resources
- An AWS service that needs to act on resources in your account to provide its features
- Users from a corporate directory who use identity federation with SAML

IAM roles issue keys that are valid for short durations, making them a more secure way to grant access.

Additional resources:

- IAM Roles FAQ
- IAM Roles Documentation
- Tutorial: Setting Up Cross Account Access
- Common Scenarios for Roles

Create AWS Lambda Project



Create role

Select type of trusted entity

AWS service EC2, Lambda and others	Another AWS account Belonging to you or 3rd party	Web identity Cognito or any OpenID provider	SAML 2.0 federation Your corporate directory

Allows AWS services to perform actions on your behalf. Learn more

Choose a use case

Common use cases

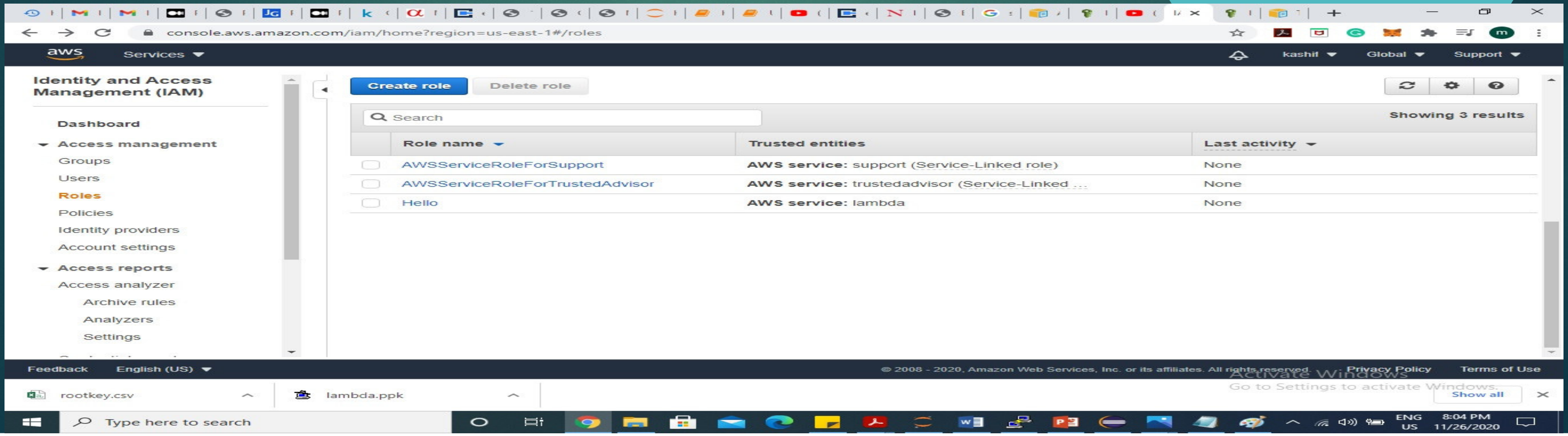
EC2
Allows EC2 instances to call AWS services on your behalf.

Lambda
Allows Lambda functions to call AWS services on your behalf.

Or select a service to view its use cases

* Required

© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.



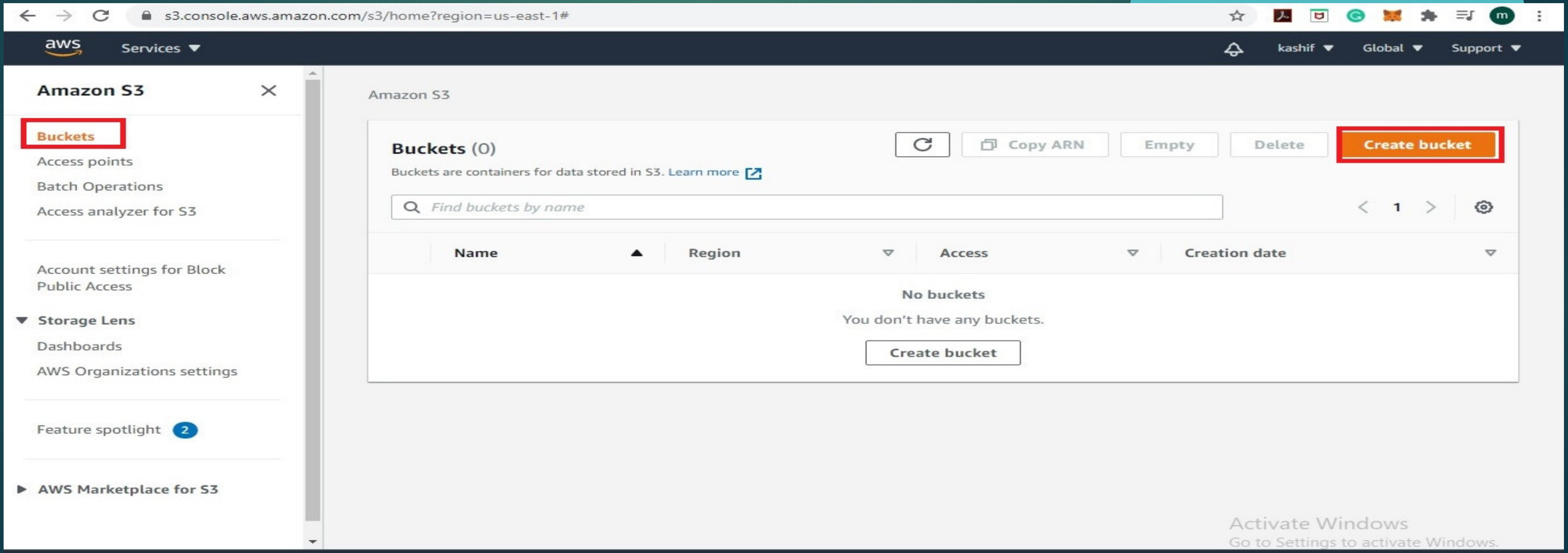
Showing 3 results

	Role name	Trusted entities	Last activity
□	AWSServiceRoleForSupport	AWS service: support (Service-Linked role)	None
□	AWSServiceRoleForTrustedAdvisor	AWS service: trustedadvisor (Service-Linked ...	None
□	Hello	AWS service: lambda	None

© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Create AWS S3 Bucket

- Step 7: To create an AWS S3 Bucket
  - Sign in to the AWS Management Console.
  - From the **Services** menu, open the S3 console.
  - Choose **Create bucket**.



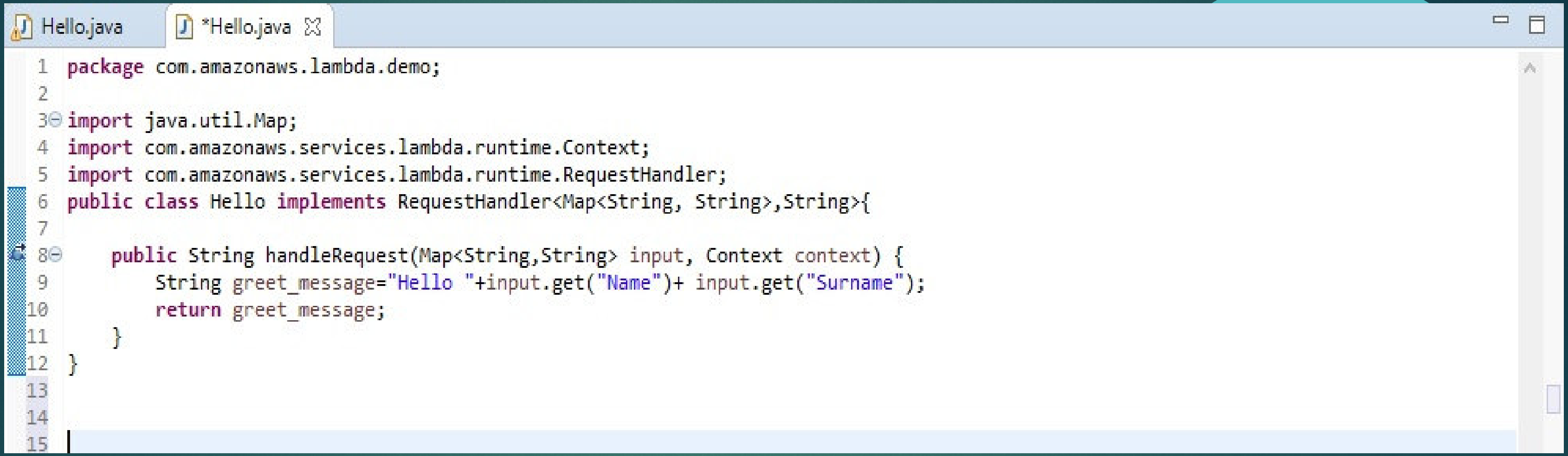
Buckets are containers for data stored in S3. Learn more

No buckets

You don't have any buckets.

Invoke Lambda Project

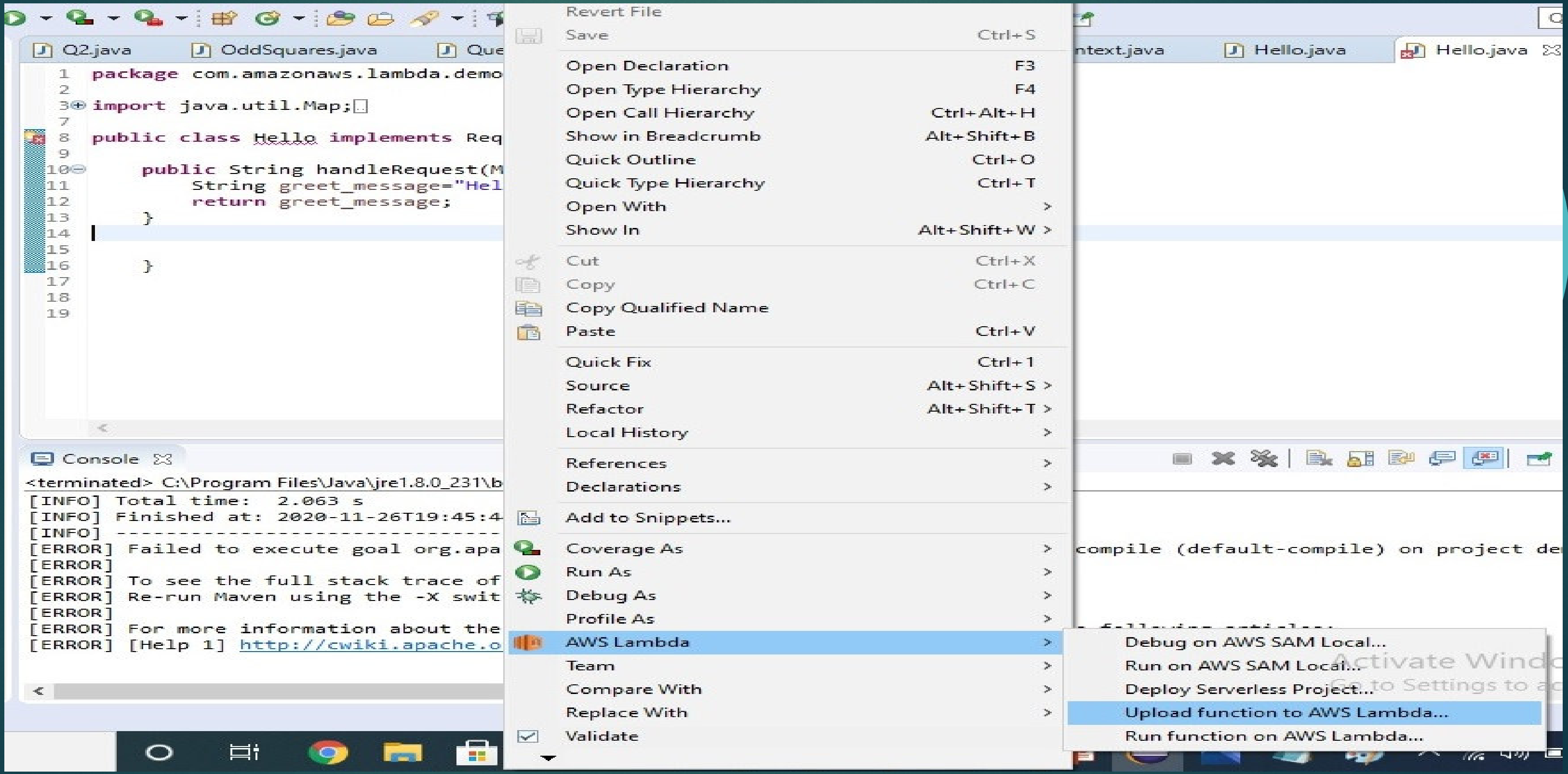
- Step 8: After creating the Lambda Java Project, you need to make following changes to **RequestHandler** and **handleRequest** methods to be able to send multiple inputs.



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

import java.util.Map;
import com.amazonaws.services.lambda.runtime.Context;
import com.amazonaws.services.lambda.runtime.RequestHandler;
public class Hello implements RequestHandler<Map<String, String>,String>{

    public String handleRequest(Map<String,String> input, Context context) {
        String greet_message="Hello "+input.get("Name")+ input.get("Surname");
        return greet_message;
    }
}
```

Invoke Lambda Project

- Step 9: First upload the java project to AWS console (Right click in the code space)

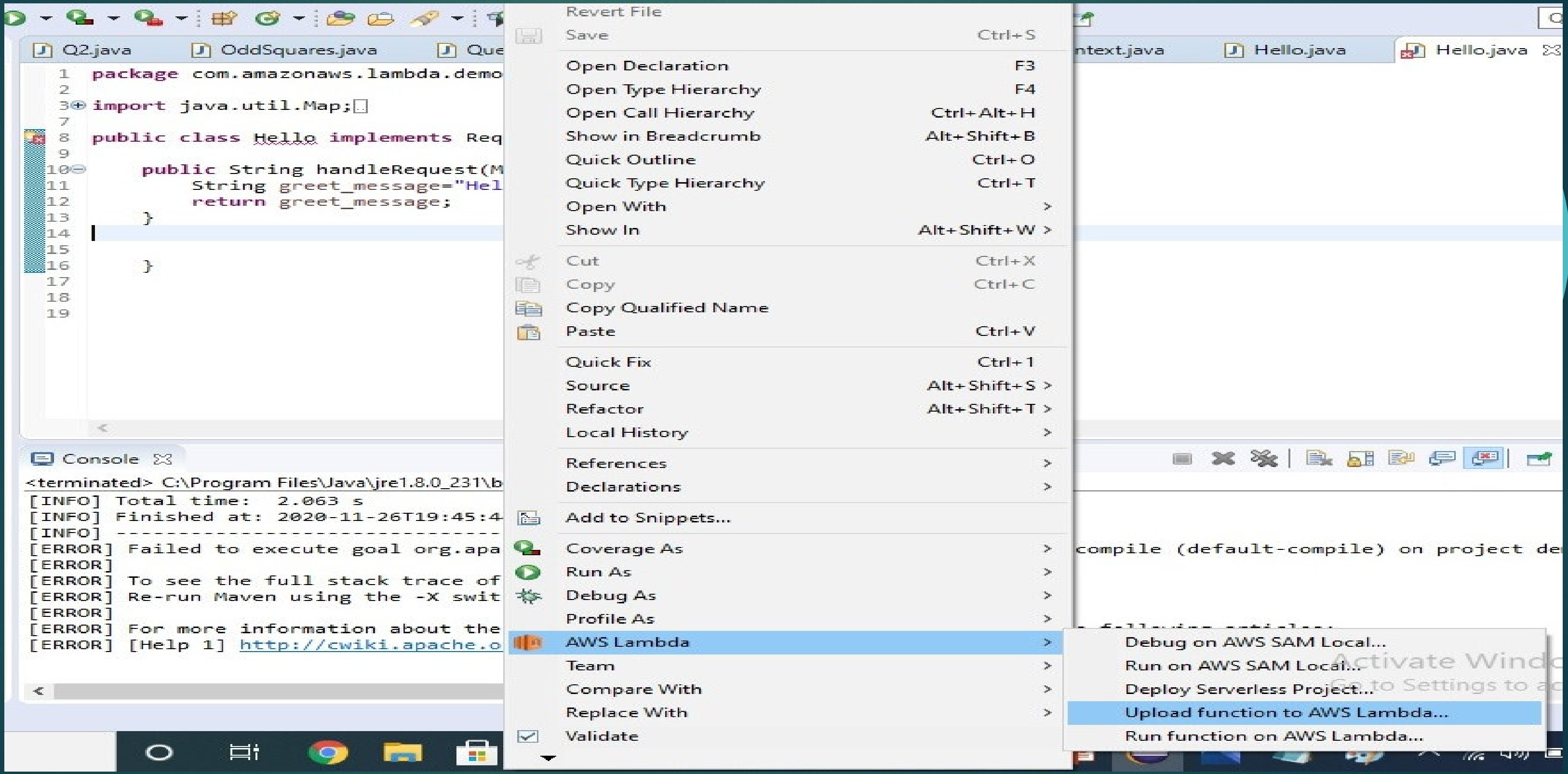


```
package com.amazonaws.lambda.demo

import java.util.Map;

public class Hello implements Req

    public String handleRequest(M
        String greet_message="Hel
        return greet_message;
    }


    }
```

```
<terminated> C:\Program Files\Java\jre1.8.0_231\b
[INFO] Total time:  2.063 s
[INFO] Finished at: 2020-11-26T19:45:4
[INFO] ------------------------------
[ERROR] Failed to execute goal org.apa                          compile (default-compile) on project de
[ERROR]
[ERROR] To see the full stack trace of
[ERROR] Re-run Maven using the -X swit
[ERROR]
[ERROR] For more information about the                          following articles:
[ERROR] [Help 1] http://cwiki.apache.o
```

Invoke Lambda Project

- Step 10: After uploading choose Run function on AWS Lambda

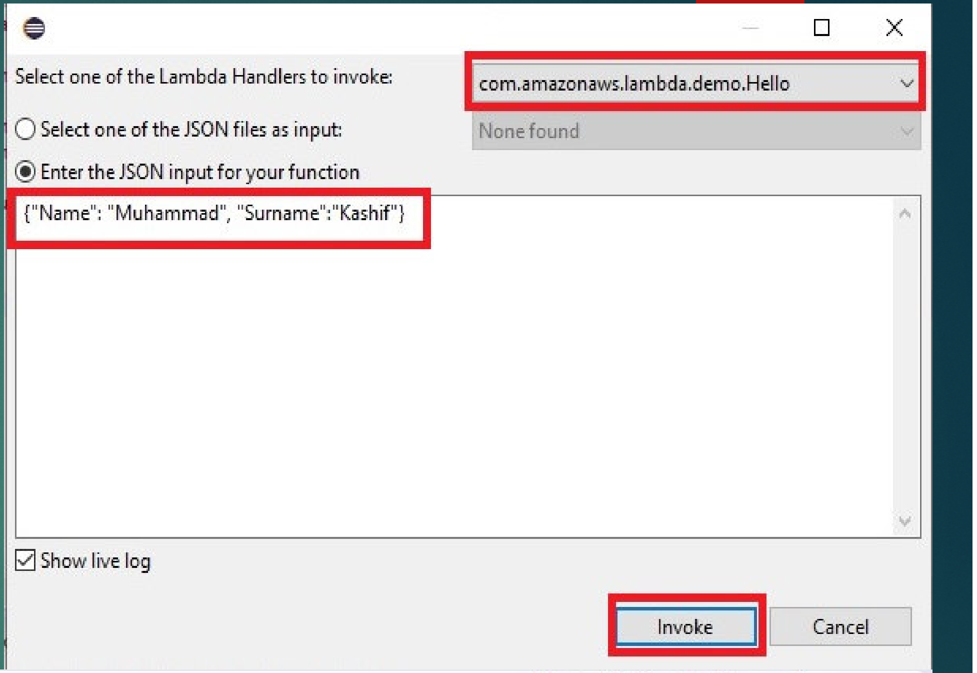


```
package com.amazonaws.lambda.demo

import java.util.Map;

public class Hello implements Req

    public String handleRequest(M
        String greet_message="Hel
        return greet_message;
    }


    }
```

```
<terminated> C:\Program Files\Java\jre1.8.0_231\b
[INFO] Total time:  2.063 s
[INFO] Finished at: 2020-11-26T19:45:4
[INFO] ------------------------------------
[ERROR] Failed to execute goal org.apa          compile (default-compile) on project de
[ERROR]
[ERROR] To see the full stack trace of
[ERROR] Re-run Maven using the -X swit
[ERROR]
[ERROR] For more information about the          following articles:
[ERROR] [Help 1] http://cwiki.apache.o
```

Invoke Lambda Project

- Step 11: Invoke Lambda project
- It will give the console output



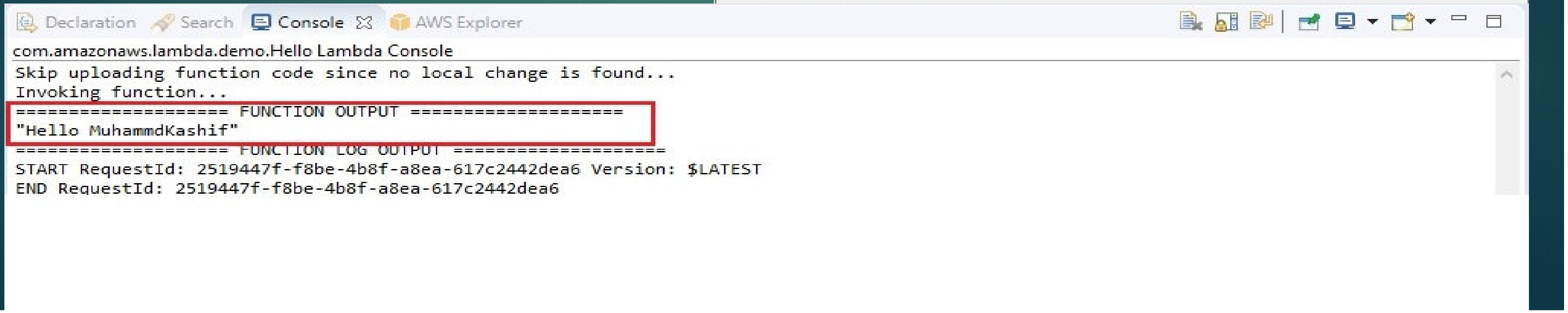
Select one of the Lambda Handlers to invoke: com.amazonaws.lambda.demo.Hello

○ Select one of the JSON files as input: None found

● Enter the JSON input for your function

```
{"Name": "Muhammad", "Surname":"Kashif"}
```

☑ Show live log



com.amazonaws.lambda.demo.Hello Lambda Console

```
Skip uploading function code since no local change is found...
Invoking function...
==================== FUNCTION OUTPUT ====================
"Hello MuhammdKashif"
==================== FUNCTION LOG OUTPUT ====================
START RequestId: 2519447f-f8be-4b8f-a8ea-617c2442dea6 Version: $LATEST
END RequestId: 2519447f-f8be-4b8f-a8ea-617c2442dea6
```

Releated Material

- Setting up Eclipse IDE (Prerequisite): https://docs.aws.amazon.com/en_us/toolkit-for-eclipse/v1/user-guide/setup-install.html

- Setting up AWS Credentials (Prerequisite): https://docs.aws.amazon.com/en_us/toolkit-for-eclipse/v1/user-guide/setup-credentials.html

- Create, Upload, and Invoke an AWS Lambda Function: https://docs.aws.amazon.com/en_us/toolkit-for-eclipse/v1/user-guide/lambda-tutorial.html
- If you have any questions, plese email me.

Muhammad.kashif@ozu.edu.tr