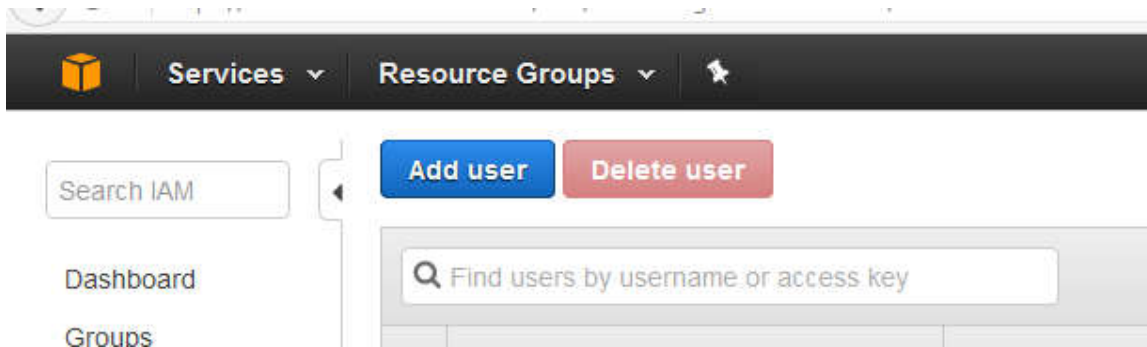
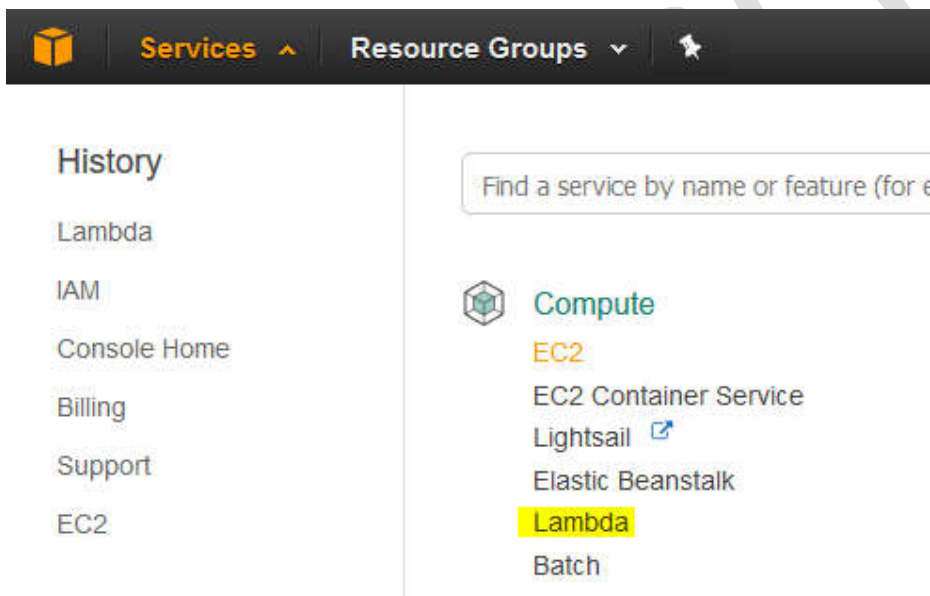


Step 1: Create a role by following below:



Step 2: Select the lambda service



Step 3: Blank function

After selecting lambda we will have to select one blueprint and here we select Blank Function

New function

blueprint

triggers

function

Select blueprint

Blueprints are sample configurations of event sources and Lambda functions. Choose a blueprint that best aligns with your desired scenario and customize as needed, or skip this step if you want to author a Lambda function and configure an event source separately. Except where otherwise noted, blueprints are licensed under [CC0](#).

Welcome to AWS Lambda! You can get started on creating your first Lambda function by choosing one of the blueprints below.

Select runtime

Filter

<<

< Viewing 1-9 of 93

>

>>

Blank Function

Configure your function from scratch. Define the trigger and deploy your code by stepping through our wizard.

custom

kinesis-firehose-syslog-to-json

An Amazon Kinesis Firehose stream processor that converts input records from RFC3164 Syslog format to JSON.

nodejs · kinesis-firehose

alexa-skill-kit-sdk-factskill

Demonstrate a basic fact skill built with the ASK NodeJS SDK

nodejs6.10 · alexa

batch-get-job-python27

batch-get-job-python27

kinesis-firehose-apachelog-to-j...

kinesis-firehose-apachelog-to-j...

cloudfront-modify-response-he...

cloudfront-modify-response-he...

Step 4: click next



Here the blank function will help us to get the custom lambda options and with the help of that we can create as per our configuration.

Go to next step

Configure triggers

You can choose to add a trigger that will invoke your function.

Welcome to AWS Lambda! You can get started on creating your first Lambda function by choosing one of the blueprints below.

  **Lambda** Remove

Cancel Previous Next

Step: configure the Lambda function

Next we need to configure the function name and platform as it shown below.

Configure function

A Lambda function consists of the custom code you want to execute. [Learn more](#) about Lambda functions.

Name*

Description

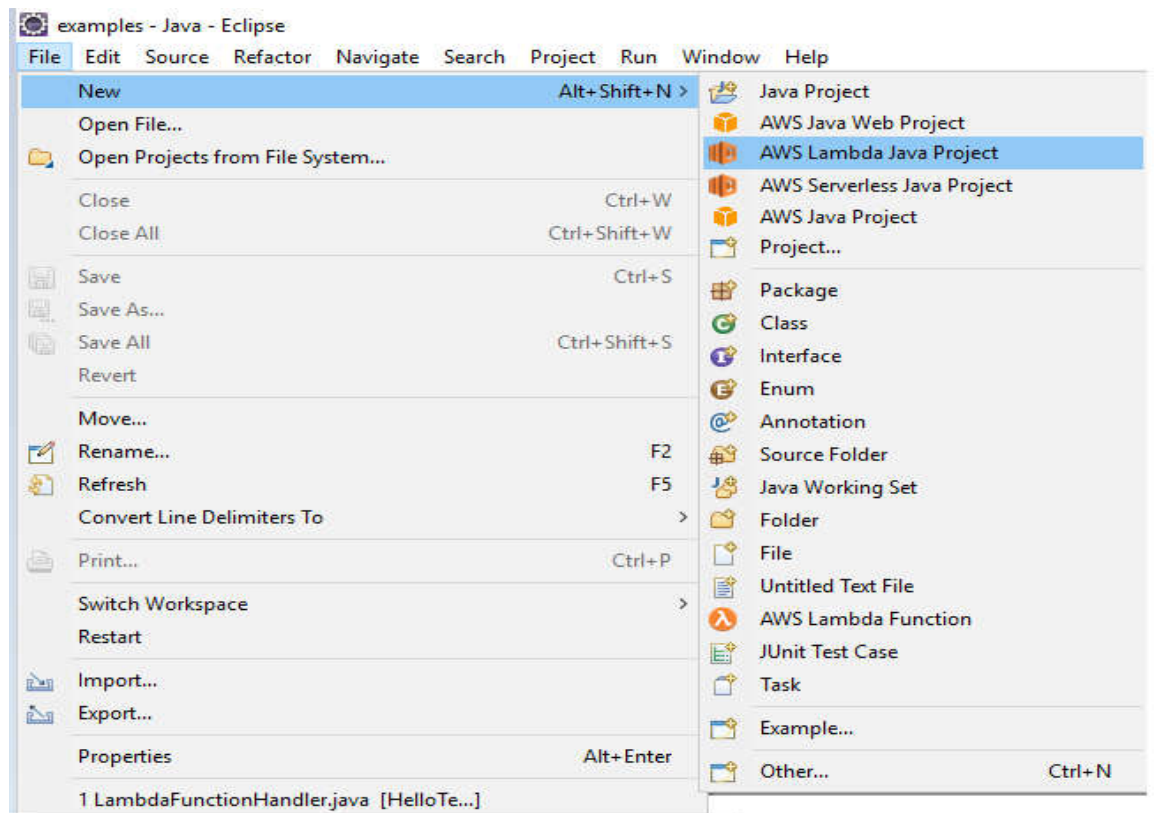
Runtime*

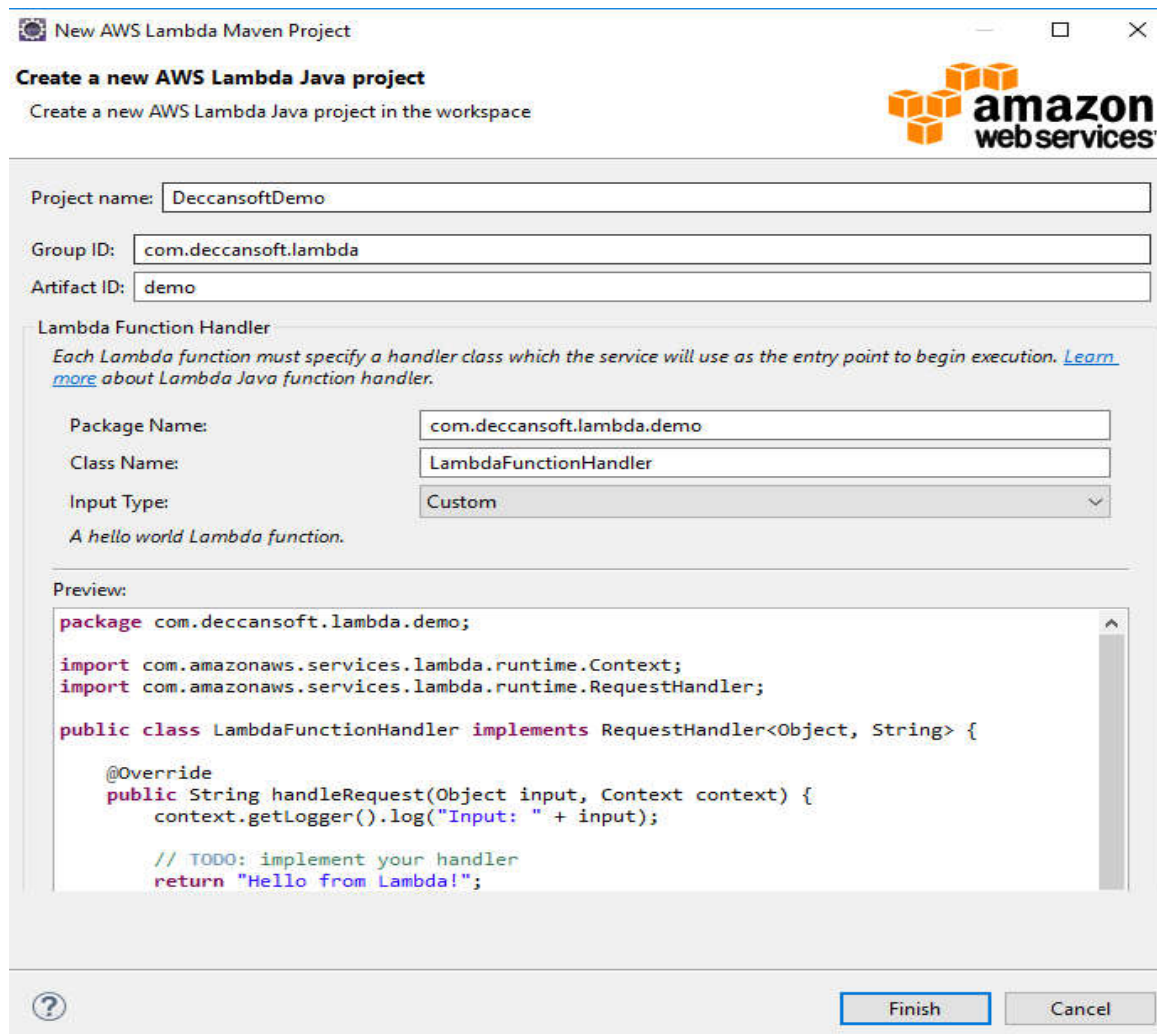
Lambda function code

Provide the code for your function. [Learn more](#) about deploying Lambda functions.

Code entry type

Now here we will understand the creation of AWS project in eclipse

Step 6: create new AWS Lambda Java Project

Step 7: follow as given below

Create a new AWS Lambda Java project
Create a new AWS Lambda Java project in the workspace

Project name:

Group ID:

Artifact ID:

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.

Package Name:

Class Name:

Input Type:

A hello world Lambda function.

Preview:

```
package com.deccansoft.lambda.demo;

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

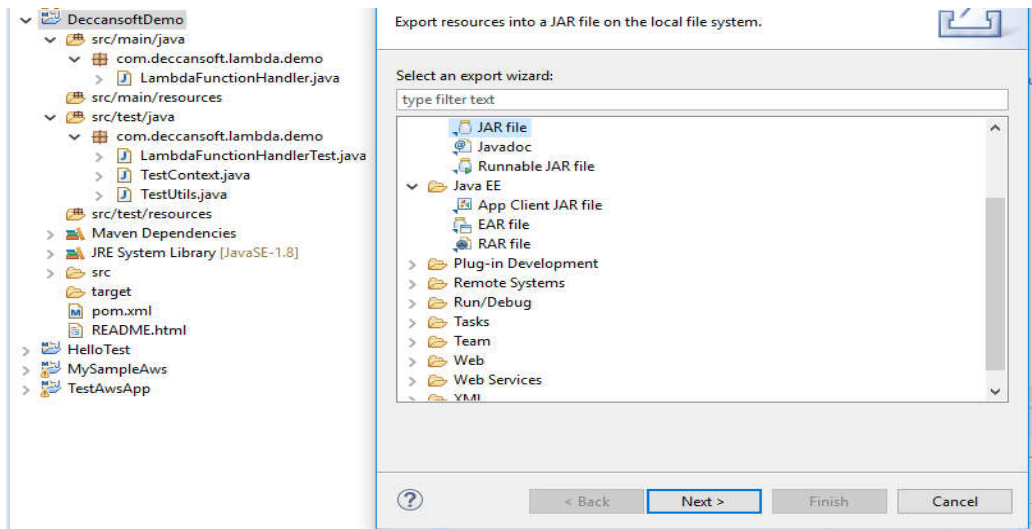
public class LambdaFunctionHandler implements RequestHandler<Object, String> {

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

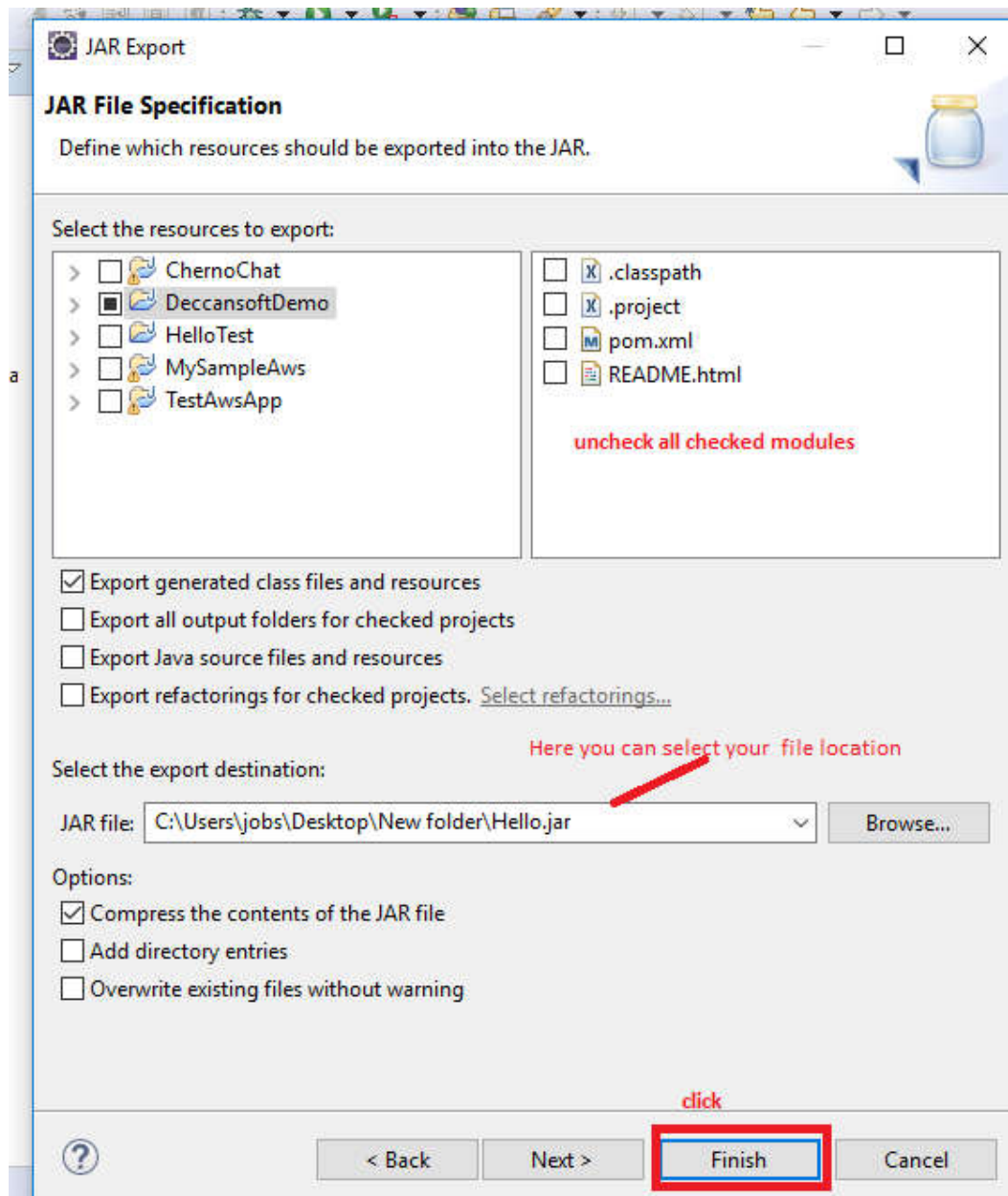
        // TODO: implement your handler
        return "Hello from Lambda!";
    }
}
```

Note: In the above module, change the input type as Custom

Step 8: create jar file for the particular application and then that jar file we need to add in step 5.



Step 9: select all checked fields and create jar file by following below image



Step 10: upload the jar file as it was showing the step 5

Provide the code for your function. [Learn more](#) about deploying Lambda functions.

Code entry type Upload a .ZIP or JAR file

Function package* Upload upload the created jar of the particular project
For files larger than 10 MB, consider uploading via S3.

You can define Environment Variables as key-value pairs that are accessible from your function code. These are useful to store configuration settings without the need to change function code. [Learn more](#). For storing sensitive information, we recommend encrypting values using KMS and the console's encryption helpers.

Step 11: After uploading the jar we will find the below view.

Function package* Upload Hello.jar (24.7 kB)
For files larger than 10 MB, consider uploading via S3.

You can define Environment Variables as key-value pairs that are accessible from your function code. These are useful to store configuration settings without the need to change function code. [Learn more](#). For storing sensitive information, we recommend encrypting values using KMS and the console's encryption helpers.

Enable encryption helpers ☐ leave it as it is not required to fill anything here

Environment variables

Key	Value
-----	-------

Lambda function handler and role

Handler* soft.lambda.demo.LambdaFunctionHandler i
Role* Choose an existing role i
Here I have selected the existing role which can be taken as new also,
Existing role* RahulSir i
provide the fully qualified name of your lambda function class

► Tags

Step 12: we can finally check and edit if anything went wrong.

Lambda function E

Name Greeting

Description welcome message

Runtime Java 8

Environment variables

Handler com.deccansoft.lambda.demo.LambdaFunctionHandler

Existing role* RahulSir

Tags

DLQ Resource

Memory (MB) 512

Step 13:--Lambda function is created.

Lambda > Functions > Greeting click here to test **ARN** - arn:aws:lambda:us-east-2:205125362654:function:Greeting

Qualifiers **Test** Actions

Congratulations! Your Lambda function "Greeting" has been successfully created. You can now click on the "Test" button to input a test event and test your function. finally Lambda function is created and we can test that.

This function contains external libraries. Uploading a new file will override these libraries.

Code Configuration Triggers Tags Monitoring

Your Lambda function "Greeting" cannot be edited inline since the file name specified in the handler does not match a file name in your deployment package.

Code entry type Upload a .ZIP or JAR file

Function package* **Upload**

For files larger than 10 MB, consider uploading via S3.

Step 14: Here we are passing the parameter.

Use the editor below to enter an event to test your function with. You can edit the event again by choosing **Configure test event** in the Actions list. Note that changes to the event will only be saved locally.

Sample event template Hello World ▾

1	"We gives you the best service"
---	---------------------------------

Final output will be given as it is visible

Lambda > Functions > Greeting ARN - arn:aws:lambda:us-east-2:205125362654:function:Greeting

Qualifiers ▾ **Test** Actions ▾

Code Configuration Triggers Tags Monitoring ?

Your Lambda function "Greeting" cannot be edited inline since the file name specified in the handler does not match a file name in your deployment package.

Code entry type Upload a .ZIP or JAR file ▾

Function package* Upload

✓ Execution result: succeeded ([logs](#)) ✕ □ □

The area below shows the result returned by your function execution. [Learn more](#) about returning results from your function.

"welcome to deccansoft Lambda example:www.deccansoft.com"

Summary Log output

Code SHA-256 `vnJ3B+PQ/0xM1NMLqRiyavW8AYqwf1m14CcR4aRR74s=` The area below shows the logging calls in your code. These correspond to a single row within the CloudWatch log group corresponding to this Lambda function. [Click here](#) to view the CloudWatch log group.