



Lambdas En Java- Parte I

Por: Cristian David Dallos Bustos

Agenda



Contexto



Caso de Estudio



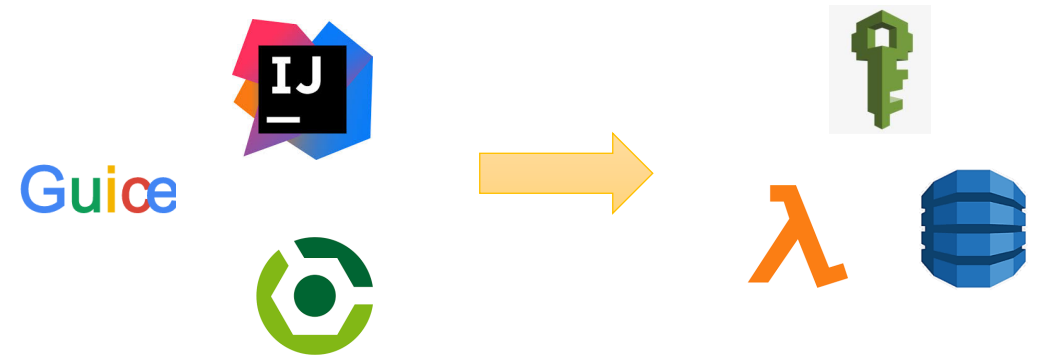
Demo

Contexto

En la presente sesión vamos a construir una lambda en lenguaje “Java”, basada en el framework de inyección de dependencias “Google Guice”.

Al finalizar la sesión habremos logrado:

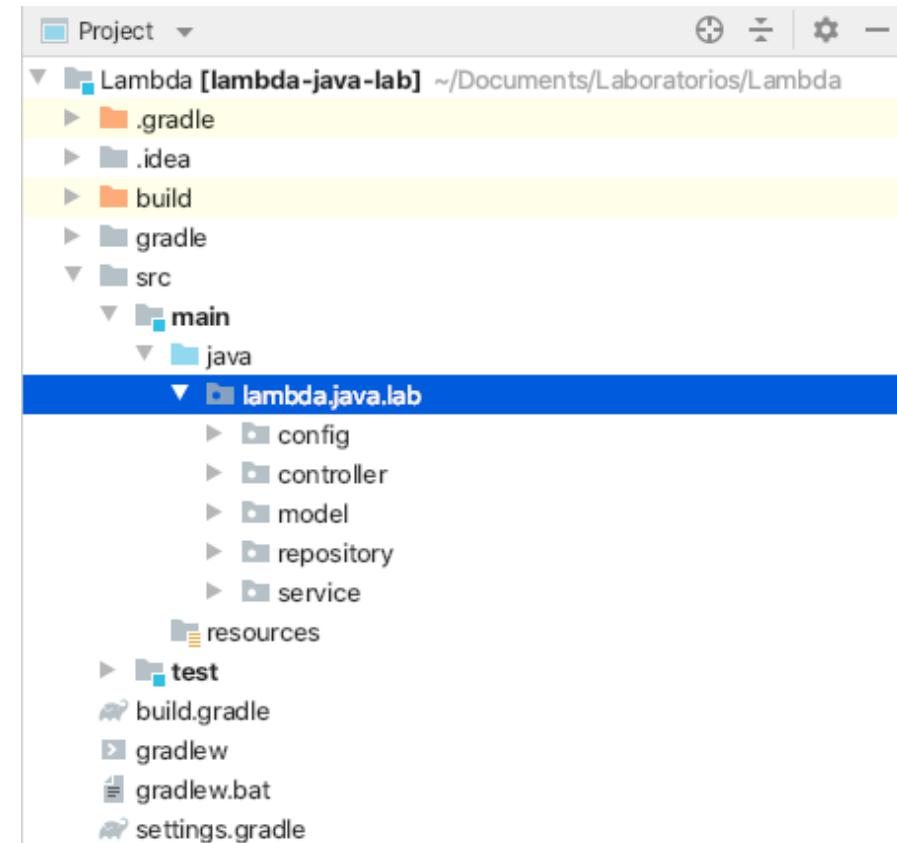
- Contrucción y empaquetamiento de una Lambas con Java y Gradle.
- Despliege de una lambda en AWS
- Creación de tabla de DynamoDB e inserción de datos.
- Creación de política en IAM - AWS



Caso de Estudio

A continuación vamos a contruir un proyecto en lenguaje Java con la siguiente anatomía:

- Paquete Controlador: Clase o paquete encargado de exponer el método HandlerRquest a invocar.
- Configuración: Clase o Paquete contenedor se los componente a inyectar.
- Servicio: Clase o Paquete contenedor del método de negocio ofrecido por la lambda.
- Repositorio: Clase o Paquete contenedor de la interfaz de acceso al servicio de DynamoDB.
- Modelo: Clase o Paquete contenedor de los modelos usados en el proceso de negocio.



Demo

Crear Tabla DyamoDB

Ingresar a la consola AWS e ir a la opcion DynamoDB:

- Dar clic en la opcion “Create DynamoDB table” e ingresar el nombre “lambda-java-lab”.
- Agregar el “primary-key” “lambda-java-lab-id”
- Dejar el resto de características por efecto y dar clic en el boton “create”

Create DynamoDB table

[Tutorial](#) 

DynamoDB is a schema-less database that only requires a table name and primary key. The table's primary key is made up of one or two attributes that uniquely identify items, partition the data, and sort data within each partition.

Table name* 

Primary key* Partition key
  

☐ Add sort key

Table settings

Default settings provide the fastest way to get started with your table. You can modify these default settings now or after your table has been created.

☒ Use default settings

- No secondary indexes.
- Provisioned capacity set to 5 reads and 5 writes.
- Basic alarms with 80% upper threshold using SNS topic "dynamodb".
- Encryption at Rest with DEFAULT encryption type.

 You do not have the required role to enable Auto Scaling by default.
Please refer to [documentation](#).

+ Add tags **NEW!**

Additional charges may apply if you exceed the AWS Free Tier levels for CloudWatch or Simple Notification Service. Advanced alarm settings are available in the CloudWatch management console.

[Cancel](#) [Create](#)

Demo

Crear Politica

Ingresar a la consola AWS e ir a la opcion IAM:

- Dar clic en la opcion “Policies” y dar clic en el boton “Create Policy”.
- Cambiar a la opcion “JSON” y agregar el siguiente snippet ubicada en el archivo policy.json del repo “lambda-java-demo-01”.
- Asignar el nombre “lambda-lab-policy”

```
Visual editor  JSON
1 {
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Sid": "VisualEditor0",
6       "Effect": "Allow",
7       "Action": [
8         "dynamodb:PutItem",
9         "dynamodb:GetItem"
10      ],
11      "Resource": "{arn::dynamodb::table}"
12    }
13  ]
14 }
```

Create policy

1 2

Review policy

Name* lambda-lab-policy

Use alphanumeric and '+,=,@-.' characters. Maximum 128 characters.

Description lambda-lab-policy

Maximum 1000 characters. Use alphanumeric and '+,=,@-.' characters.

Summary

Filter

Service	Access level	Resource	Request condition
Allow (1 of 265 services) Show remaining 264			
DynamoDB	Limited: Read, Write	TableName string like lambda-java-lab	None

Demo

Crear Role

Ingresar a la consola AWS e ir a la opción IAM:

- Dar clic en la opción “Roles” y dar clic en el botón “Create Role”.
- Buscar las políticas “AWSLambdaBasicExecutionRole” y “lambda-lab-policy”
- Asignar el nombre “lambda-lab-role”

Create role

1 2 3 4

Review

Provide the required information below and review this role before you create it.

Role name*

lambda-lab-role

Use alphanumeric and '+=, @-_' characters. Maximum 64 characters.

Role description

Allows Lambda functions to call AWS services on your behalf.

Maximum 1000 characters. Use alphanumeric and '+=, @-_' characters.

Trusted entities

AWS service: lambda.amazonaws.com

Policies

 [AWSLambdaBasicExecutionRole](#)

[lambda-lab-policy](#)

Permissions boundary

Permissions boundary is not set

The new role will receive the following tag

Key	Value
role-name	lambda-lab-role

Demo

Crear Lambda

Ingresa a la consola AWS e ir a la opción Lambda:

- Dar clic en el botón “Create Function”.
- Seleccionar la opción “Author from scratch”.
- Asignar el nombre “lambda-lab”
- Seleccionar Java 8
- Usar la opción “Use an existing role” y seleccionar el rol “lambda-lab-role”

Lambda > Functions > Create function

Create function [Info](#)

Choose one of the following options to create your function.

Author from scratch ☒
Start with a simple Hello World example.

Use a blueprint ☐
Build a Lambda application from sample code and configuration presets for common use cases.

Container image ☐
Select a container image to deploy for your function.

Basic information

Function name
Enter a name that describes the purpose of your function.

Use only letters, numbers, hyphens, or underscores with no spaces.

Runtime [Info](#)
Choose the language to use to write your function.

Permissions [Info](#)
By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.

▼ **Change default execution role**

Execution role
Choose a role that defines the permissions of your function. To create a custom role, go to the [IAM console](#).

☐ Create a new role with basic Lambda permissions
☒ Use an existing role
☐ Create a new role from AWS policy templates

Existing role
Choose an existing role that you've created to be used with this Lambda function. The role must have permission to upload logs to Amazon CloudWatch Logs.

[View the lambda-lab-role role](#) on the IAM console.

Demo

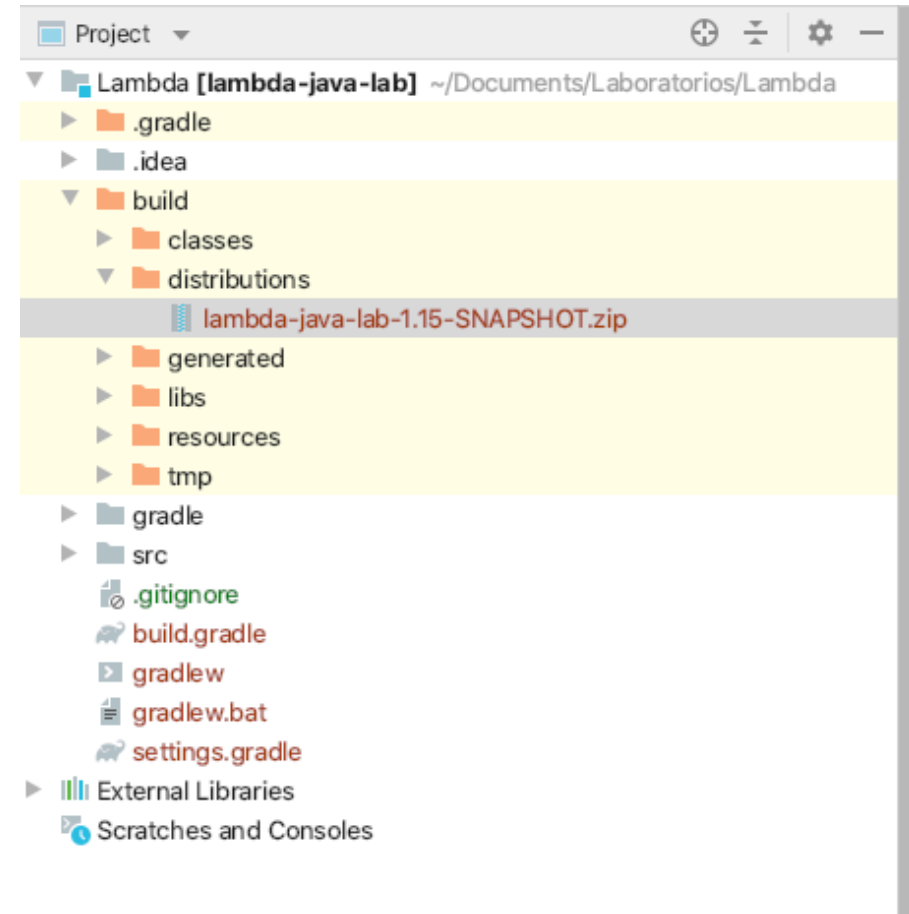
Compilar Lambda

Para este punto se requiere:

- IntelliJ
- Java 8
- Gradle
- Git

Ingresa al repositorio <https://github.com/Tutorial-Labs/lambda-java-demo-01> y:

- clonar el proyecto lambda-java-demo-01.
- En el terminal del proyecto tirar el comando “gradle clean build”
- El resultado debera generar una carpeta .zip en la ruta build/distributions

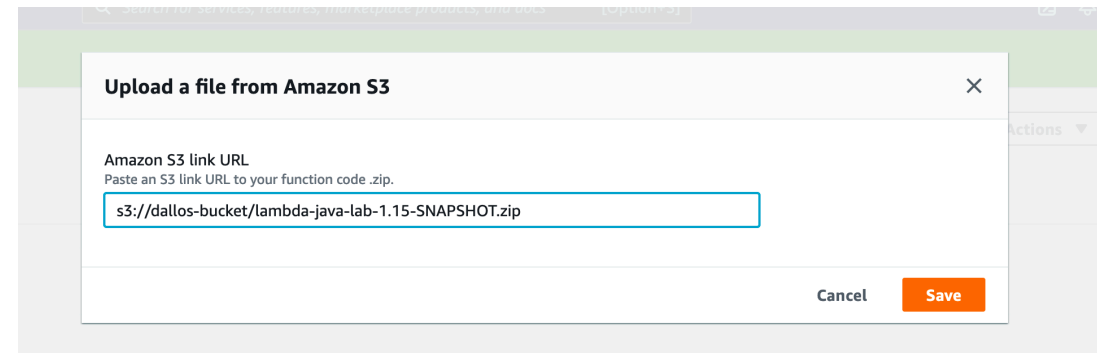


Demo

Cargar Lambda

Ingresa a la consola AWS e ir a la opción Lambda:

- Buscar la lambda “lambda-lab”.
- En la sección “Function Code”, dar clic en la opción “Upload zip or jar file” y seleccionar el zip creado en el paso anterior.



Demo

Configurar HandleRequest

Ingresa a la consola AWS e ir a la opción Lambda:

- Buscar la lambda “lambda-lab”.
- En la sección “Runtime Settings”, editar el nombre y asignar “lambda.java.lab.controller.LambdaController::handleRequest”.


Lambda > Functions > lambda-lab > Edit runtime settings

Edit runtime settings

Runtime settings [Info](#)

Runtime

Java 8 ▼

 **New runtime available**

A new runtime is available for your function's language: Java 11 (Corretto)

×

Handler [Info](#)

lambda.java.lab.controller.LambdaController::handleRequest

Cancel Save

Demo

Probar Lambda

Ingresar a la consola AWS e ir a la opcion Lambda:

- Buscar la lambda “lambda-lab”.
- En la seccion “Test” agregar el fragmento de código:

```
{  
  "lambdaName": "adl-lambda-java-lab"  
}
```

Configure test event

A function can have up to 10 test events. The events are persisted so you can switch to another computer or web browser and test your function with the same events.

- ☒ Create new test event
- ☐ Edit saved test events

Event template

hello-world

Event name

test

```
1 {  
2   "lambdaName": "lambda-java-lab"  
3 }
```

lambda-lab

Execution result: succeeded (logs)

Details

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

null

Summary

Code SHA-256 b78X8fkg4dhJtulyOvWWu9T2KdzRZCgtCADzgn18g=	Request ID c0ab5427-1675-4ca0-bbd4-8a6af7037550
Init duration 437.78 ms	Duration 11133.12 ms
Billed duration 11134 ms	Resources configured 512 MB
Max memory used 149 MB	

Log output

The section 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.

```
START RequestId: c0ab5427-1675-4ca0-bbd4-8a6af7037550 Version: $LATEST  
[main] INFO lambda.java.lab.controller.LambdaController - Start consumer LambdaController  
[main] INFO lambda.java.lab.controller.LambdaController - Input: {"lambdaName": "lambda-java-lab"}  
[main] INFO lambda.java.lab.service.LambdaServiceImpl - Start save() LambdaServiceImpl  
[main] INFO lambda.java.lab.repository.LambdaDynamoDbRepositoryImpl - Start save() LambdaDynamoDbRepositoryImpl  
[main] INFO lambda.java.lab.repository.LambdaDynamoDbRepositoryImpl - End save() LambdaDynamoDbRepositoryImpl: {}  
[main] INFO lambda.java.lab.service.LambdaServiceImpl - Start end() LambdaServiceImpl: Test Java Lambda  
[main] INFO lambda.java.lab.controller.LambdaController - End consumer LambdaController  
END RequestId: c0ab5427-1675-4ca0-bbd4-8a6af7037550  
REPORT RequestId: c0ab5427-1675-4ca0-bbd4-8a6af7037550 Duration: 11133.12 ms Billed Duration: 11134 ms Memory Size: 512 MB Max Memory Used: 149 MB Init Duration: 437.78 ms
```

Demo

Validar Resultado

[Create table](#) [Delete table](#)

✕

Choose a table ... ▼

Actions ▼

Name
<input checked="" type="radio"/> lambda-java-lab

lambda-java-lab [Close](#)

Overview

Items

Metrics

Alarms

Capacity

Indexes

Global Tables

[Create item](#)

Actions ▼

Scan: [Table] lambda-java-lab: lambda-java-lab-id ^

Scan

[Table] lambda-java-lab: lambda-java-lab-id

+ Add filter

Start search

☐

lambda-java-lab-id ⓘ

▲

lambda-name ▼



Referencias

- <https://google.github.io/guice/api-docs/4.2/javadoc/index.html>
 - <https://spring.io/projects/spring-cloud>
 - <https://quarkus.io>
 - <https://github.com/Tutorial-Labs>
 - <https://github.com/Java-Techie-jt/springboot-aws-lambda>
-