Lambdas En Java- Parte I

Por: Cristian David Dallos Bustos

Agenda

Contexto

Caso de Estudio

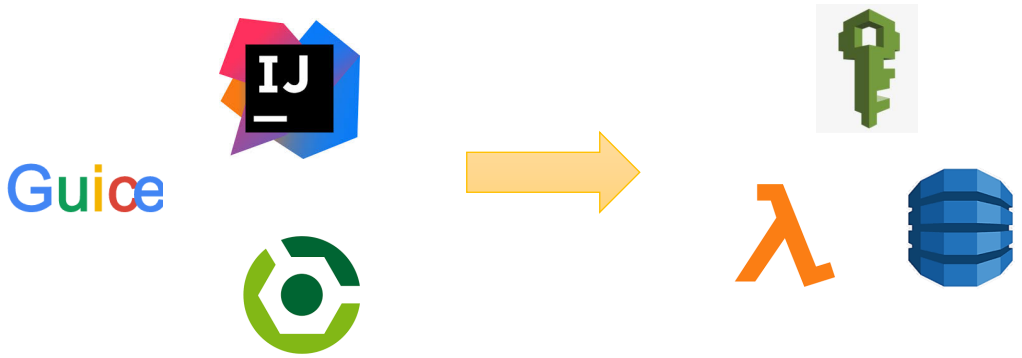
Demo

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:

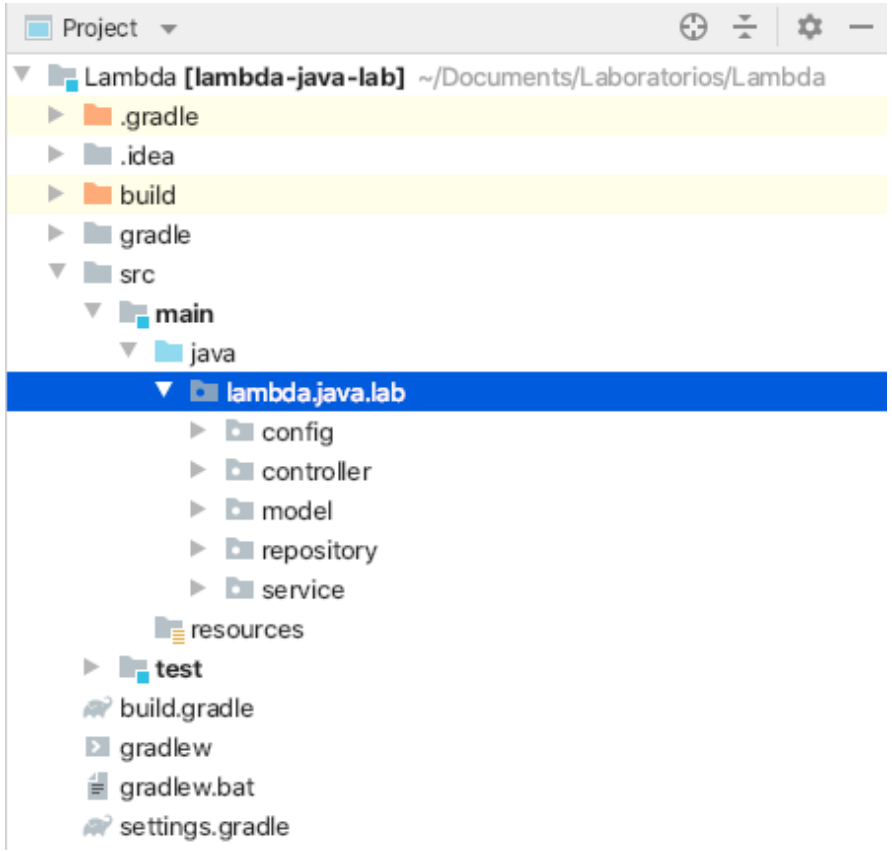
- Contrucción y empaquetamiento de una Lambas con Java y Gradle.
- Despliege de una lambda en AWS
- Creación de tabla de DynamoDB e inserción de datos.
- Creación de política en IAM - AWS



Caso de Estudio

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

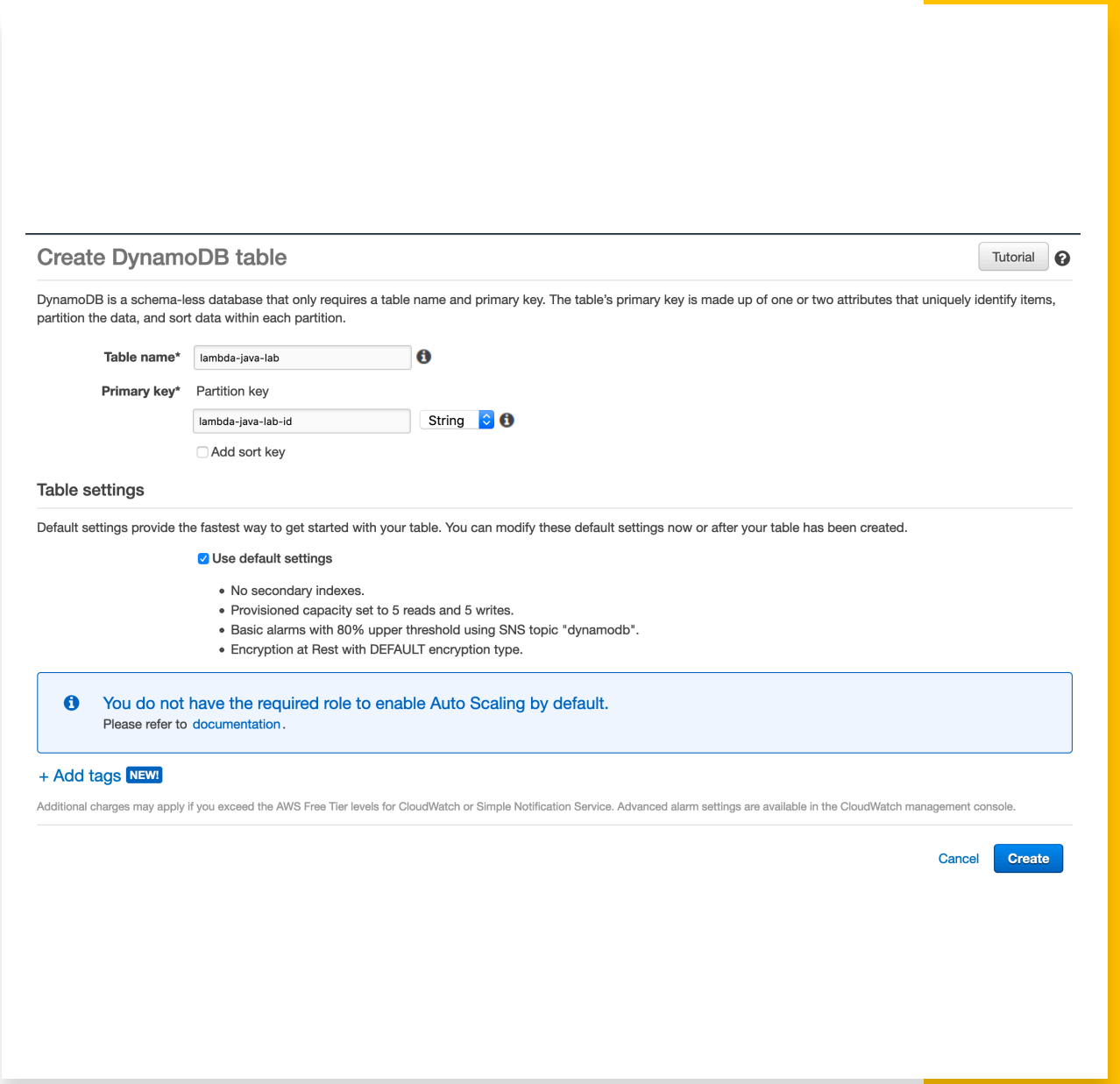
- Paquete Controlador: Clase o paquete encargado de exponer el método HandlerRquest a invocar.
- Configuración: Clase o Paquete contenedor se los componente a inyectar.
- Servicio: Clase o Paquete contendor del método de negocio ofrecido por la lambda.
- Repositorio: Clase o Paquete contenedor de la 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:

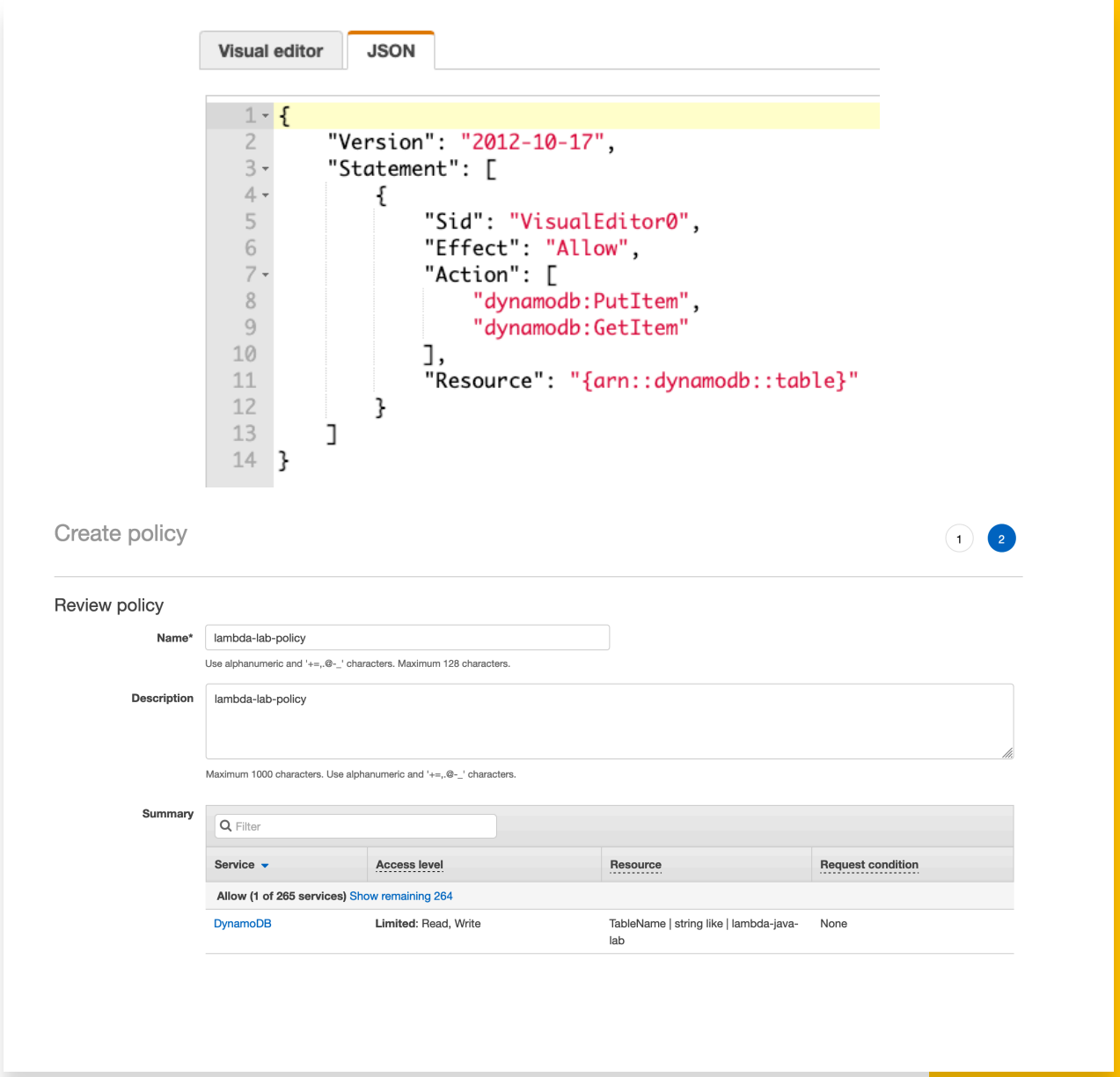
- Dar clic en la opcion “Create DynamoDB table” e ingresar el nombre “lambda-java-lab”.
- Agregar el “primary-key” “lambda-java-lab-id”
- Dejar el resto de caracteristicas por efecto y dar clic en el boton “create”



Demo
Crear Politica

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

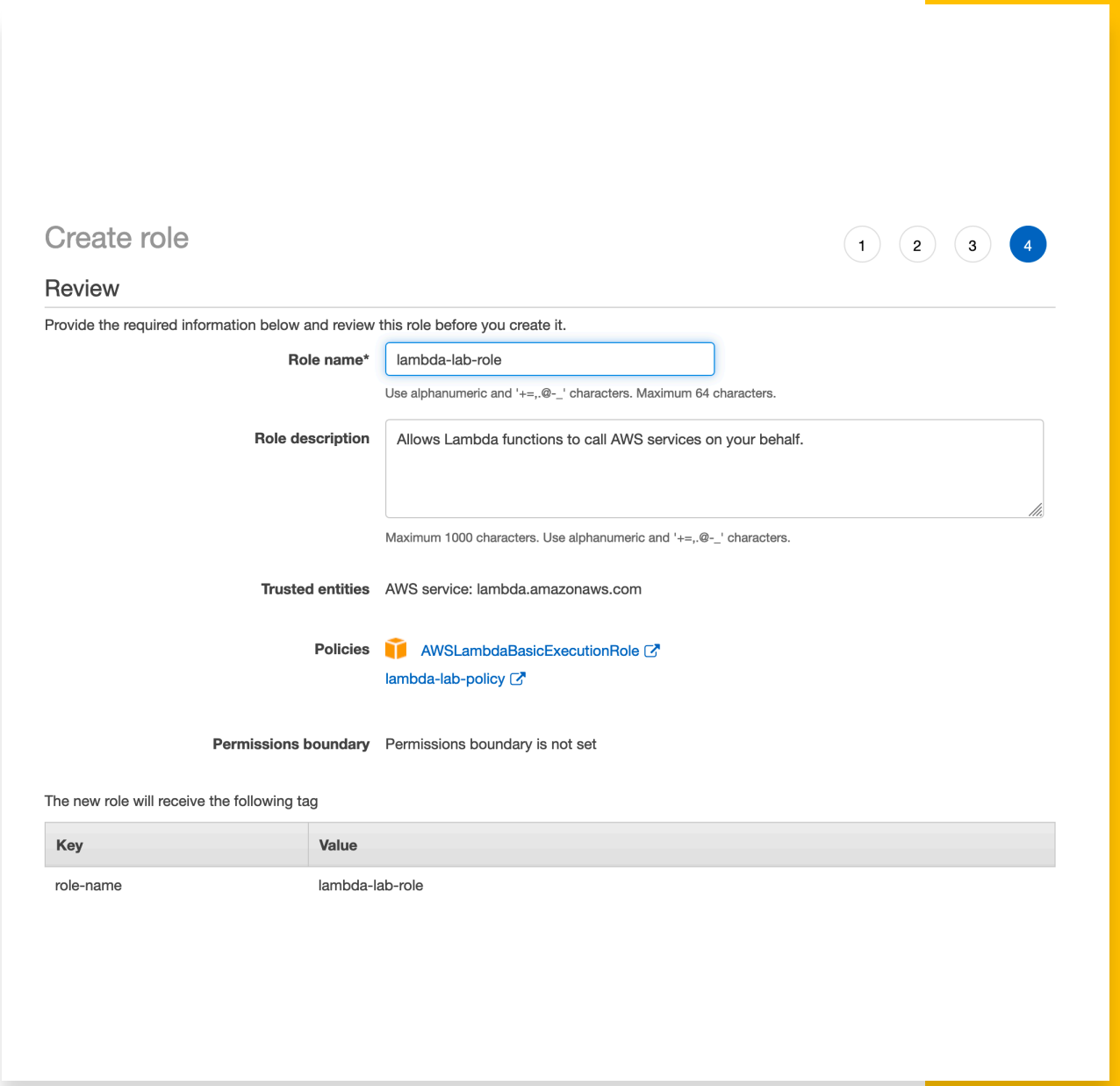
- Dar clic en la opcion “Policies” y dar clic en el boton “Create Policy”.
- Cambiar a la opcion “JSON” y agregar el siguiente snippet ubicada en el archivo policy.json del repo “lambda-java-demo-01”.
- Asignar el nombre “lambda-lab-policy”



Demo
Crear Role

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

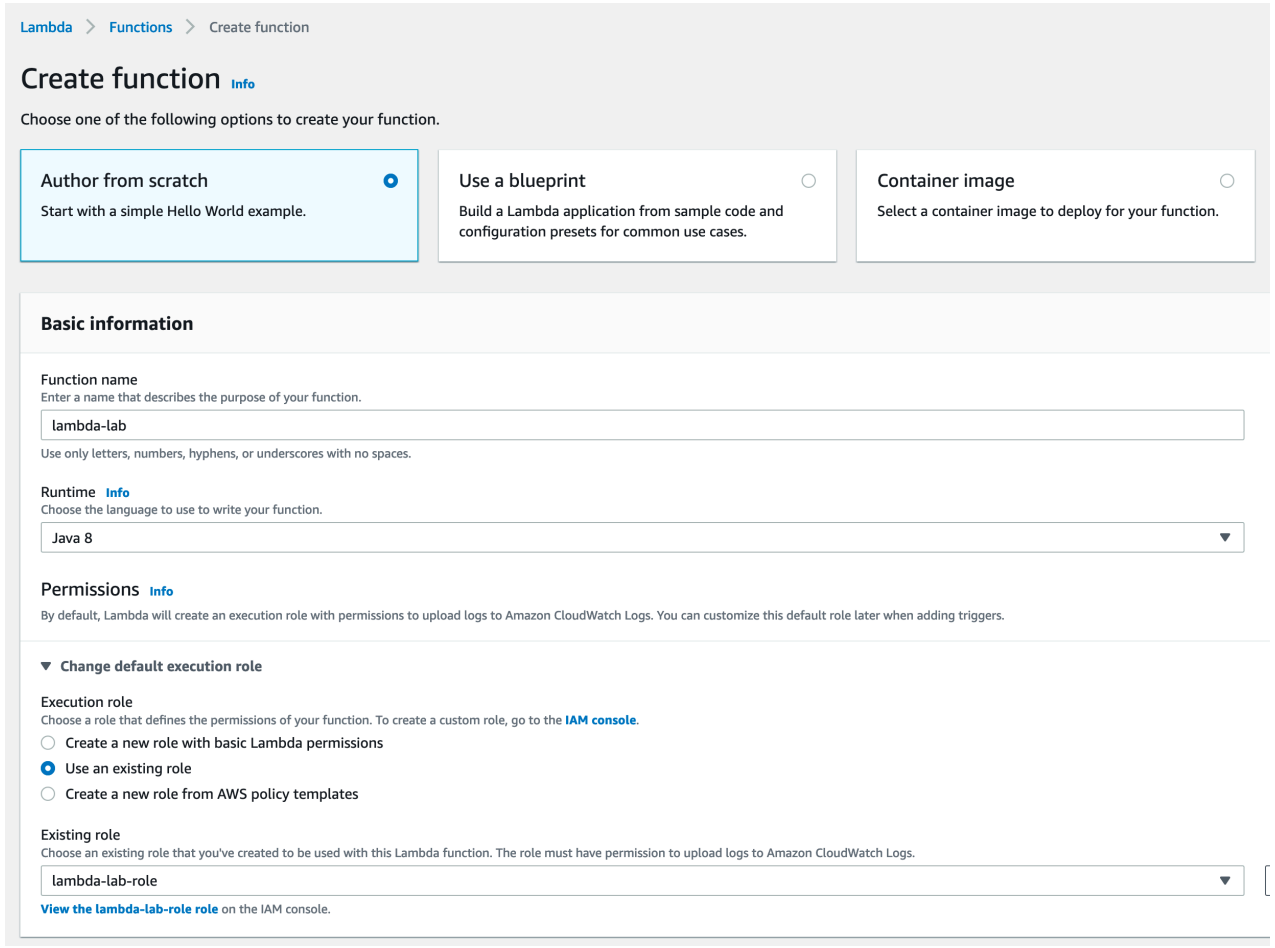
- Dar clic en la opcion “Roles” y dar clic en el boton “Create Role”.
- Buscar las políticas “AWSLambdaBasicExecutionRole” y “lambda-lab-policy”
- Asignar el nombre “lambda-lab-role”



Demo
Crear Lambda

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

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



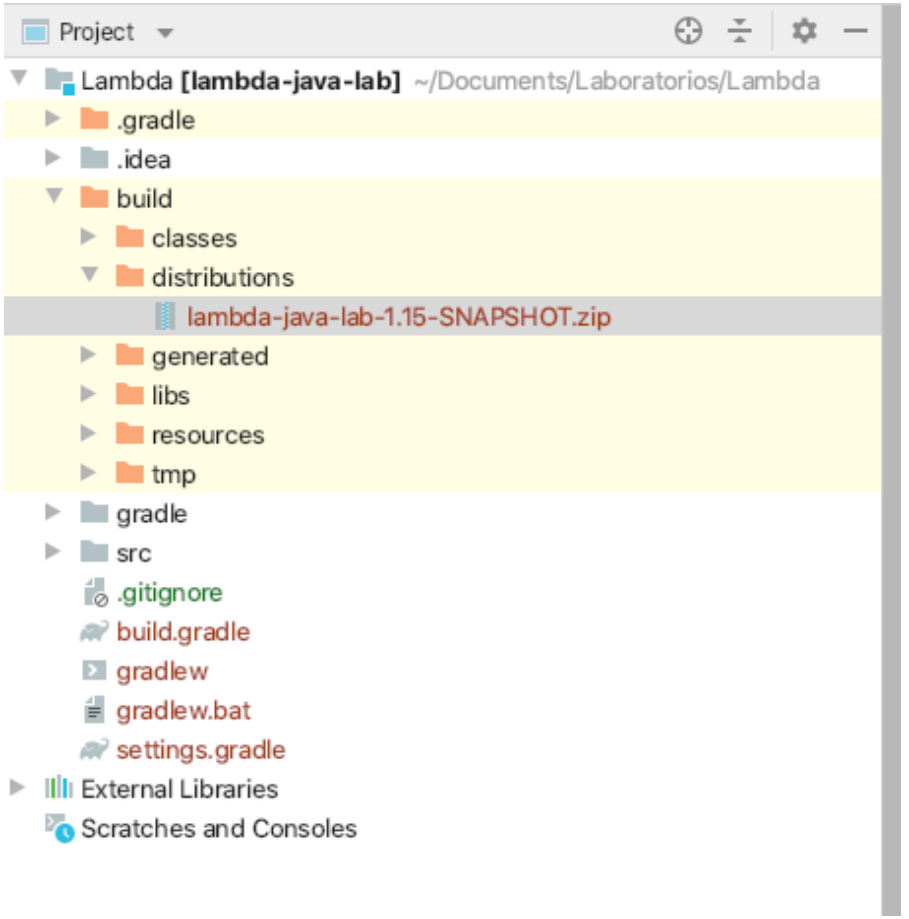
Demo
Compilar Lambda

Para este punto se requeire:

- IntelliJ
- Java 8
- Gradle
- Git

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

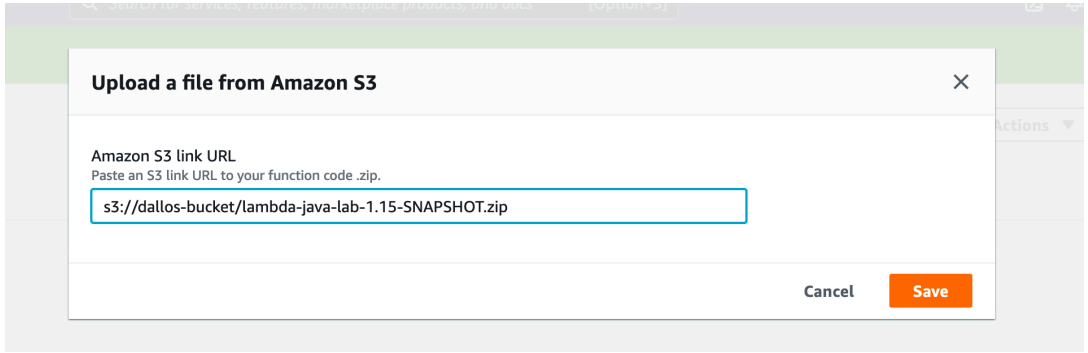
- clonar el proyecto lambda-java-demo-01.
- En el terminal del proyecto tirar el comando “gradle clean build”
- El resultado debera generar una carpeta .zip en la ruta build/distributions



Demo
Cargar Lambda

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

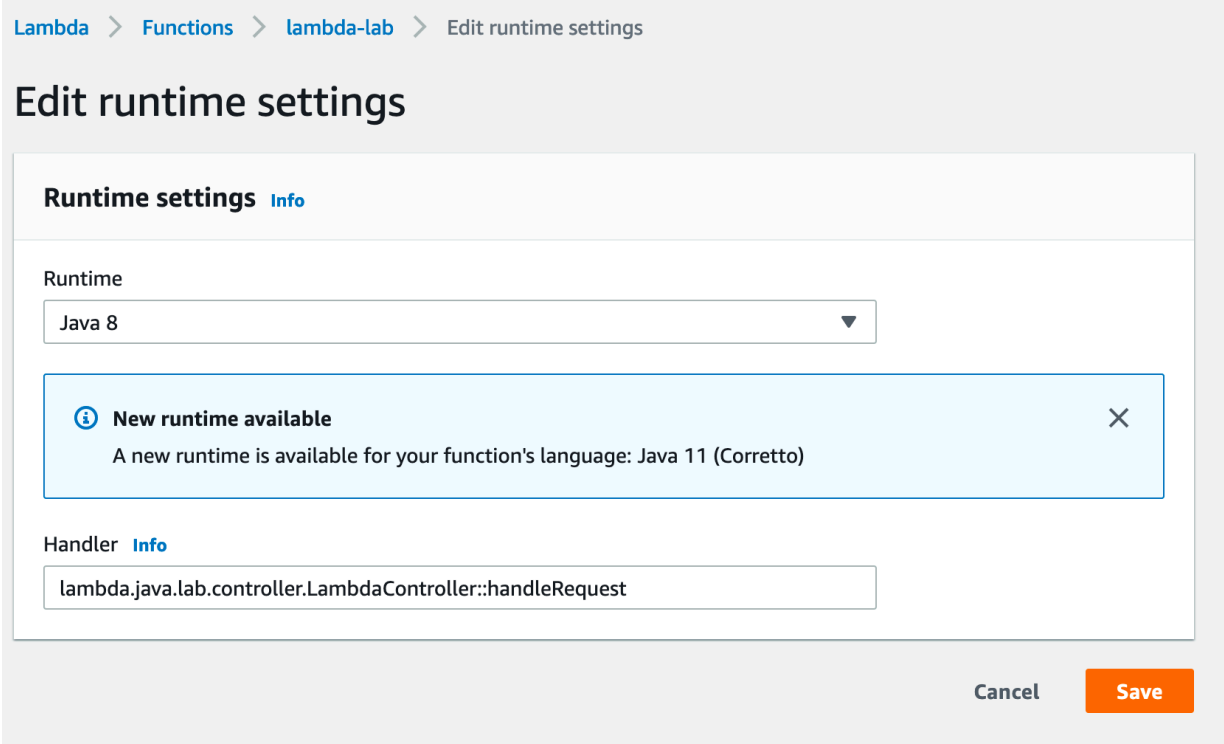
- Buscar la lambda “lambda-lab”.
- En la seccion “Function Code”, Dar clic en la opcion “Upload zip or jar file” y seleccionar el zip creado en el paso anterior.



Demo
Configurar HandleRequest

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

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

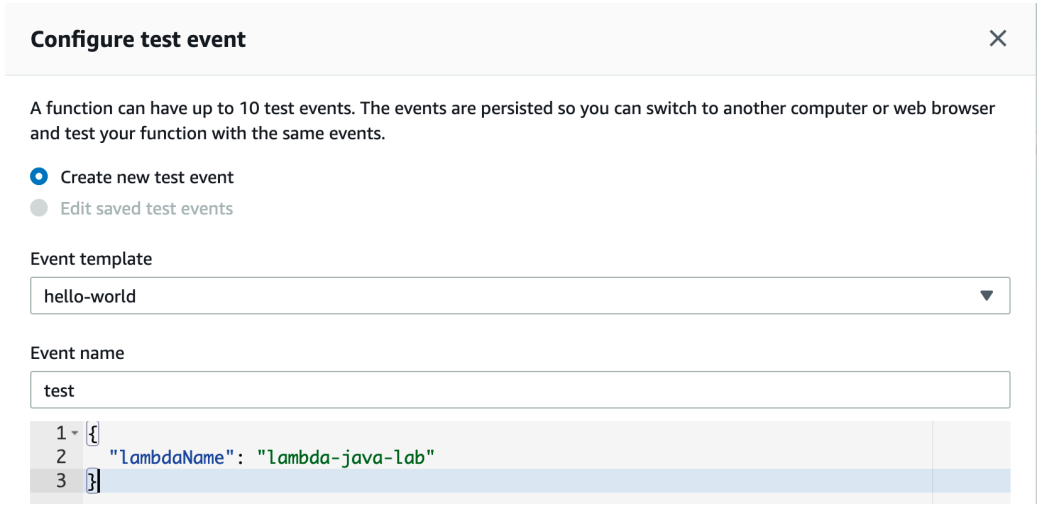


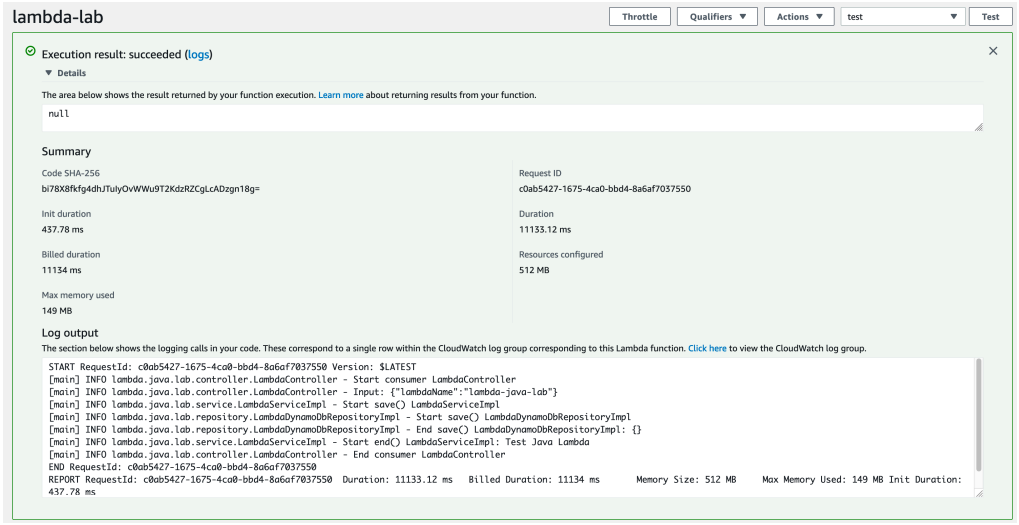
Demo
Probar Lambda

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

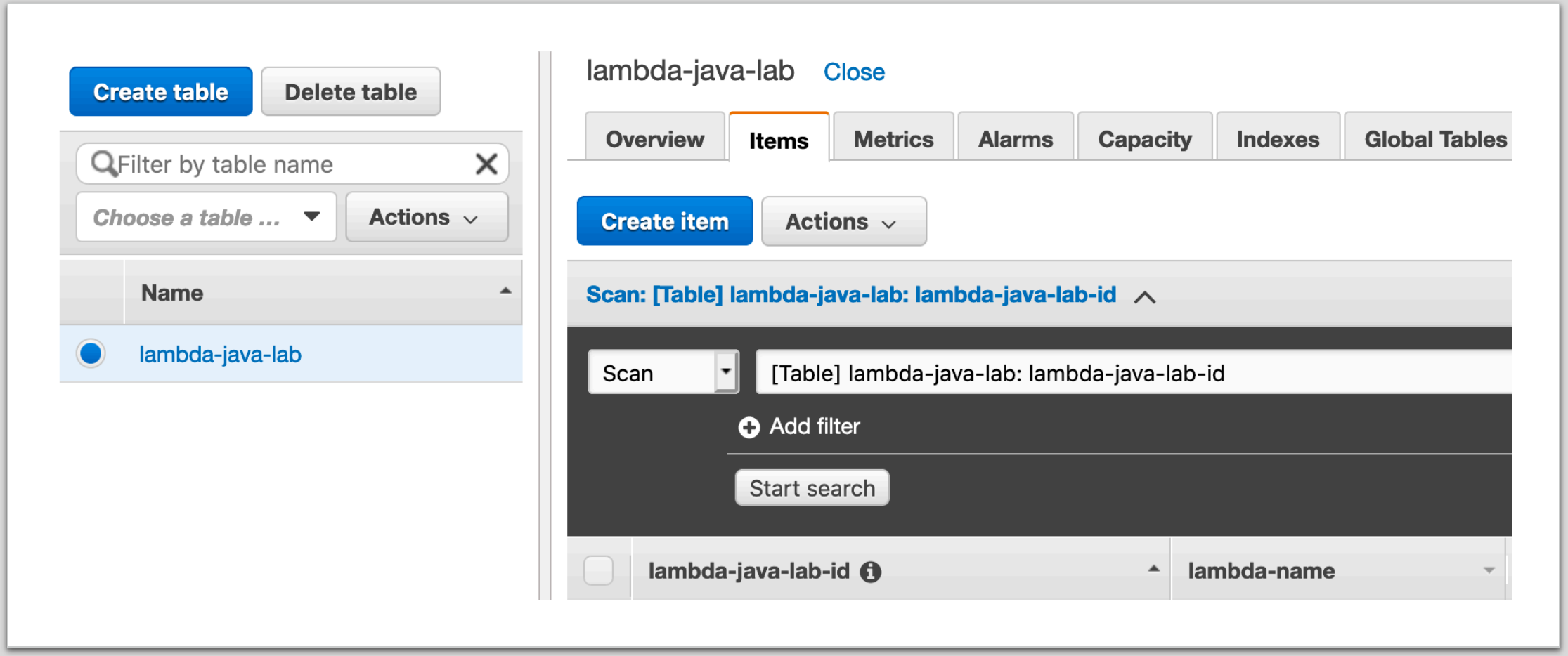
- Buscar la lambda “lambda-lab”.
- En la seccion “Test” agregar el fragmento de código:

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





Demo
Validar Resultado



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