AWS Demo Java Spring Cloud Function Serverless

If Java is your choice of programming language-Spring Cloud Function,Serverless Framework makes a great technology stack. It boosts developer productivity by decoupling from Vendor specific FaaS API, and deployment activities.

View on Github

Example Demonstrating Spring Cloud Function Framework + AWS Lambda + Serverless Framework

This example uses Serverless Framework to deploy a Spring Cloud Function on AWS Lambda Platform.

Quick introduction to Spring Cloud Function

Spring Cloud Function provides an uniform programming model to develop functions which can be run on any FaaS platforms like AWS Lambda. The same code can run as a web endpoint, a stream processor, or a task. Also, enables Spring Boot features. More information is availabe at Spring Cloud Function

Example project

Use-case

This is a scenario in which when we provide a Member ID from Health Insurance Card, the service should reply with the type of coverage that member has. In this fictional scenario it always returns a coverage type as MEDICAL for a given Member ID.

Function

{% highlight java %}

```java
@Bean
public Function<HealthFirstMemberRequest, HealthFirstMemberResponse> members() {
        return member -> {
                HealthFirstMemberResponse response = new HealthFirstMemberResponse();
                response.setMemberId(member.getMemberId());
                response.setCoverage(HealthFirstMemberResponse.Coverage.MEDICAL);
            return response;
    };
}
```

{% endhighlight %}

AWS Lambda Handler

A Handler class which just implements SpringBootRequestHandler is needed. This is what we are going to configure in serverless.yml file.

Maven POM file

The following dependencies in POM.xml file does all the magic of generating AWS specific Lambda code.

```xml
<dependency>
  <groupId>org.springframework.cloud</groupId>
  <artifactId>spring-cloud-function-adapter-aws</artifactId>
</dependency>
<dependency>
  <groupId>com.amazonaws</groupId>
  <artifactId>aws-lambda-java-core</artifactId>
```

```
    <version>${aws-lambda-core.version}</version>
    <scope>provided</scope>
  </dependency>

```

Setup

Generate deployable Artifacts

Run following command to generate deployable / uploadable .jar file.

```
$mvn clean package
```

This gnerates a JAR file **'member-function-0.0.1-SNAPSHOT-aws.jar'** under target folder. Serverless Framework uploads this JAR file to AWS Lambda.

serverless.yml

We define the serverless.yml file with all the settings needed.

```
service: sls-aws-spring-cloud-function
provider:
  name: aws
  runtime: java8
  timeout: 10
package:
  artifact: target/aws-java-spring-cloud-function-demo-0.0.1-SNAPSHOT-aws.jar
functions:
  members:
    handler: com.healthfirst.memberfunction.AwsLambdaHandler
    environment:
      FUNCTION_NAME: members
```

Deployment

Please run following command to deploy the project on AWS Lambda

```
$sls deploy -v --aws-profle <your AWS profile name>
```

You would see something similar to this

```
Serverless: Packaging service...
Serverless: Creating Stack...
Serverless: Checking Stack create progress...
CloudFormation - CREATE_IN_PROGRESS - AWS::CloudFormation::Stack - sls-aws-java-spring-cloud-function-demo-d
CloudFormation - CREATE_IN_PROGRESS - AWS::S3::Bucket - ServerlessDeploymentBucket
CloudFormation - CREATE_IN_PROGRESS - AWS::S3::Bucket - ServerlessDeploymentBucket
CloudFormation - CREATE_COMPLETE - AWS::S3::Bucket - ServerlessDeploymentBucket
CloudFormation - CREATE_COMPLETE - AWS::CloudFormation::Stack - sls-aws-java-spring-cloud-function-demo-dev
Serverless: Stack create finished...
Serverless: Uploading CloudFormation file to S3...
Serverless: Uploading artifacts...
Serverless: Uploading service .zip file to S3 (12.91 MB)...
Serverless: Validating template...
Serverless: Updating Stack...
Serverless: Checking Stack update progress...
CloudFormation - UPDATE_IN_PROGRESS - AWS::CloudFormation::Stack - sls-aws-java-spring-cloud-function-demo-d
CloudFormation - CREATE_IN_PROGRESS - AWS::Logs::LogGroup - MembersLogGroup
CloudFormation - CREATE_IN_PROGRESS - AWS::IAM::Role - IamRoleLambdaExecution
CloudFormation - CREATE_IN_PROGRESS - AWS::Logs::LogGroup - MembersLogGroup
CloudFormation - CREATE_IN_PROGRESS - AWS::IAM::Role - IamRoleLambdaExecution
CloudFormation - CREATE_COMPLETE - AWS::Logs::LogGroup - MembersLogGroup
CloudFormation - CREATE_COMPLETE - AWS::IAM::Role - IamRoleLambdaExecution
CloudFormation - CREATE_IN_PROGRESS - AWS::Lambda::Function - MembersLambdaFunction
CloudFormation - CREATE_IN_PROGRESS - AWS::Lambda::Function - MembersLambdaFunction
CloudFormation - CREATE_COMPLETE - AWS::Lambda::Function - MembersLambdaFunction
CloudFormation - CREATE_IN_PROGRESS - AWS::Lambda::Version - MembersLambdaVersionN8TC7eR31v5vRUjauJUxtjhrj6n
CloudFormation - CREATE_IN_PROGRESS - AWS::Lambda::Version - MembersLambdaVersionN8TC7eR31v5vRUjauJUxtjhrj6n
CloudFormation - CREATE_COMPLETE - AWS::Lambda::Version - MembersLambdaVersionN8TC7eR31v5vRUjauJUxtjhrj6n1C1
CloudFormation - UPDATE_COMPLETE_CLEANUP_IN_PROGRESS - AWS::CloudFormation::Stack - sls-aws-java-spring-clou
CloudFormation - UPDATE_COMPLETE - AWS::CloudFormation::Stack - sls-aws-java-spring-cloud-function-demo-dev
Serverless: Stack update finished...
Service Information
service: sls-aws-java-spring-cloud-function-demo
stage: dev
region: us-east-1
stack: sls-aws-java-spring-cloud-function-demo-dev
api keys:
  None
endpoints:
  None
```

```
functions:
  members: sls-aws-java-spring-cloud-function-demo-dev-members

Stack Outputs
MembersLambdaFunctionQualifiedArn: arn:aws:lambda:us-east-1:899022498951:function:sls-aws-java-spring-cloud-
ServerlessDeploymentBucketName: sls-aws-java-spring-clou-serverlessdeploymentbuck-1qadcge7s5r27


```

Please run following command to invoke the function

```
$ sls invoke -f members -l --aws-profile <your AWS profile name> --data '{"memberId":"1234567890"}'
```

This will invoke the function by passing the Member ID. You would see console output with response **MEDICAL** as below.

```
{
    "memberId": "1234567890",
    "coverage": "MEDICAL"
}
--------------------------------------------------------------------
START RequestId: ae39247a-7c6d-11e8-b022-eb1234c7df4f Version: $LATEST
13:58:36.347 [main] INFO org.springframework.cloud.function.adapter.aws.SpringFunctionInitializer - Initiali

  .   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/
 :: Spring Boot ::

2018-06-30 13:58:39.013  INFO 1 --- [           main] lambdainternal.LambdaRTEntry             : Starting La
2018-06-30 13:58:39.016  INFO 1 --- [           main] lambdainternal.LambdaRTEntry             : No active p
2018-06-30 13:58:39.297  INFO 1 --- [           main] s.c.a.AnnotationConfigApplicationContext : Refreshing 
2018-06-30 13:58:44.069  INFO 1 --- [           main] o.s.j.e.a.AnnotationMBeanExporter        : Registering
2018-06-30 13:58:44.152  INFO 1 --- [           main] lambdainternal.LambdaRTEntry             : Started Lam
END RequestId: ae39247a-7c6d-11e8-b022-eb1234c7df4f
REPORT RequestId: ae39247a-7c6d-11e8-b022-eb1234c7df4f  Duration: 8002.57 ms    Billed Duration: 8100 ms

```

Please run the following command to uninstall the project from AWS

```
$ sls remove --aws-profile <your AWS profile name>
```

You would see console output something similar as

```
Serverless: Getting all objects in S3 bucket...
Serverless: Removing objects in S3 bucket...
Serverless: Removing Stack...
Serverless: Checking Stack removal progress...
........
Serverless: Stack removal finished...
```

Summary

If Java is your choice of programming language - Spring Cloud Function + Serverless Framework makes a great technology stack. It boosts developer productivity by decoupling from Vendor specific FaaS API, and deployment activities.