**Note:**

if you are facing a permission while invoking the cloud function from your local machine through a url then try to use the below command in google cloud sdk:

gcloud functions add-invoker-policy-binding function-bucket(name of cloud function) \

--region="australia-southeast2"(region where cloud function is deployed) \

--member="MEMBER\_NAME" -->(allUsers)

**Http-Driven Function Call (it should not be a spring boot project, it will have only src folder and in outside it should pom.xml file only and in pom.xml remove <parent> tag ):**

1) Deploy the function manually through cloud console, make sure to while making click allow unauthenticated option, then also you need to provide cloud invoker permission for allUsers profile to make it run.

2) To run the function locally and manually use the command as :

Add the below plugin in pom.xml to run the function in local:

<build>

<plugins>

<plugin>

<groupId>com.google.cloud.functions</groupId>

<artifactId>function-maven-plugin</artifactId>

<version>0.11.0</version>

<configuration>

<functionTarget>com.httpcloudfunction.httpcloudfunction.HttpcloudfunctionApplication(fully qualified function name -> packagename.className)</functionTarget>

</configuration>

</plugin>

</plugins>

</build>

Run Command:

mvn function:run -Drun.functionTarget=Fully Qualified name of a function(package.className) (example: com.httpcloudfunction.httpcloudfunction.HttpcloudfunctionApplication)

You can see the output in localhost:8080 site and remember by default it is a POST method.

if you want to run the function in different port such as 8081 use the below command.

mvn function:run -Drun.functionTarget=Fully Qualified name of a function(package.className) (example: com.httpcloudfunction.httpcloudfunction.HttpcloudfunctionApplication) -Drun.port=8081

3) We can manually deploy the function in the cloud console by following site https://cloud.google.com/functions/docs/deploy#console\_1 or you can do it manually also by clicking on create cloud function in google cloud console.

We can use below command to deploy the function from command line:

gcloud functions deploy my-http-function(cloud function name) \

--gen2 \

--region=us-central1(region) \

--runtime=nodejs20(programming language) \

--source=. (directory where pom.xml is present)\

--entry-point=myHttpFunction (fully qualified function name -> packagename.className) \

--trigger-http (type of trigger)

4) We can invoke the cloud function deployed in Google cloud console by using command:

gcloud functions call function-1(function-name in cloud) --region=us-central1(region in cloud function is deployed) --gen2 --data "{\"name\":\"Kalani\"}" --> --data parameter is optional here it is generally used to send some POST Request body.

or you can invoke it by using url present in Trigger section where Url method type is POST method.

**Event-Driven Function Call(Pub/Sub Trigger) (it should not be a spring boot project, it will have only src folder and in outside it should pom.xml file only and in pom.xml remove <parent> tag):**

1) Deploy the function manually through Google cloud console and in that click on Create cloud function and in the options we can specify the pub/sub topic name through which the cloud functions gets triggered and then deploy it.

2) So whenever we publish a message through that topic, that message will be sent to concerned cloud function and it will respond based on the logic specified in cloud function and we can check the output in logs section if we are using Logger to log the output.

3) Run the following command to deploy the function through command google cloud sdk cli:

gcloud functions deploy my-pubsub-function(cloud function name) --gen2 --region=europe-west1(region name) --runtime=java17(programming language) --source=.(directory where pom.xml is present in local computer, now it is current directory) --entry-point=com.example.EventDrivenPubSub.EventDrivenPubS

ubApplication(fully qualified name of the function/class) --trigger-topic=newTopic(pub/sub topic Name)

4) To run the function locally use the following command and run this command in command cli where project pom.xml is present:

Add the below plugin in pom.xml to run the function in local:

<build>

<plugins>

<plugin>

<groupId>com.google.cloud.functions</groupId>

<artifactId>function-maven-plugin</artifactId>

<version>0.11.0</version>

<configuration>

<functionTarget>com.example.EventDrivenPubSub.EventDrivenPubSubApplication(fully qualified function name -> packagename.className)</functionTarget>

</configuration>

</plugin>

</plugins>

</build>

Run Commands:

mvn function:run -Drun.functionTarget=com.example.EventDrivenPubSub.EventDrivenPubSubApplication (fully qualified name of the function/class) -Drun.port=8081(Port number)

To check the output run the following curl request in postman and you can see the output in command cli where you initiated the above mvn command

curl localhost:8080 \ --> Port Number

-X POST \

-H "Content-Type: application/json" \

-H "ce-id: 123451234512345" \

-H "ce-specversion: 1.0" \

-H "ce-time: 2020-01-02T12:34:56.789Z" \

-H "ce-type: google.cloud.pubsub.topic.v1.messagePublished" \

-H "ce-source: //pubsub.googleapis.com/projects/MY-PROJECT(project-id)/topics/MY-TOPIC(pub/sub topic-name)" \

-d '{

"message": {

"data": "d29ybGQ=", -->(base64 encoded data)

"attributes": {

"attr1":"attr1-value"

}

},

"subscription": "projects/MY-PROJECT(project-id)/subscriptions/MY-SUB (pub/sub subscription-name)"

}'

Here -H means it is a part of Header and -d means it is a part of Request Body

5) We can also use the url present in Trigger section of pub-sub cloud function to test the function if the function is deployed to Google cloud.

curl https://europe-west1-cloud-spanner-430219.cloudfunctions.net/my-pubsub-function \ --> Port Number

-X POST \

-H "Content-Type: application/json" \

-H "ce-id: 123451234512345" \

-H "ce-specversion: 1.0" \

-H "ce-time: 2020-01-02T12:34:56.789Z" \

-H "ce-type: google.cloud.pubsub.topic.v1.messagePublished" \

-H "ce-source: //pubsub.googleapis.com/projects/MY-PROJECT(project-id)/topics/MY-TOPIC(pub/sub topic-name)" \

-d '{

"message": {

"data": "d29ybGQ=", -->(base64 encoded data)

"attributes": {

"attr1":"attr1-value"

}

},

"subscription": "projects/MY-PROJECT(project-id)/subscriptions/MY-SUB (pub/sub subscription-name)"

}'

Here -H means it is a part of Header and -d means it is a part of Request Body

And you see the logs in logs Section.

**Event-Driven Function Call(Storage Bucket Trigger) (it should not be a spring boot project, it will have only src folder and in outside it should pom.xml file only and in pom.xml remove <parent> tag):**

1) To deploy the cloud function manually in Google cloud function, Go to the cloud console and then go to cloud function, click on create cloud function and there give cloud function name and in trigger function name give the trigger type such as Cloud Bucket Storage and operation such as delete, finalize(create) etc and then click on deploy button.

2) Cloud Function gets triggered based on the trigger information given in Trigger type Section.

3) To run the function locally use the following command and run this command in command cli where project pom.xml is present:

Add the below plugin in pom.xml to run the function in local:

<build>

<plugins>

<plugin>

<groupId>com.google.cloud.functions</groupId>

<artifactId>function-maven-plugin</artifactId>

<version>0.11.0</version>

<configuration>

<functionTarget>com.example.CloudFunctionBucket.CloudFunctionBucketApplication</functionTarget>

</configuration>

</plugin>

</plugins>

</build>

Run Commands:

mvn function:run -Drun.functionTarget=com.example.CloudFunctionBucket.CloudFunctionBucketApplication(fully qualified name of the function/class) -Drun.port=8081(Port number)

To check the output run the following curl request in postman and you can see the output in command cli where you initiated the above mvn command

curl localhost:8080 \ -->(Port Number)

-X POST \

-H "Content-Type: application/json" \

-H "ce-id: 123451234512345" \

-H "ce-specversion: 1.0" \

-H "ce-time: 2020-01-02T12:34:56.789Z" \

-H "ce-type: google.cloud.storage.object.v1.finalized" \ (Specify Operation when to trigger, finalized -> means when we upload a new file in a bucket)

-H "ce-source: //storage.googleapis.com/projects/\_/buckets/MY-BUCKET-NAME(Bucket Name)" \

-H "ce-subject: objects/MY\_FILE.txt(Name of the file with extension)" \

-d '{

"bucket": "MY\_BUCKET", --> (Bucket Name)

"contentType": "text/plain",(file type where to do the operation)

"kind": "storage#object",

"md5Hash": "...",

"metageneration": "1",

"name": "MY\_FILE.txt", (Name of the file)

"size": "352",

"storageClass": "MULTI\_REGIONAL",

"timeCreated": "2020-04-23T07:38:57.230Z",

"timeStorageClassUpdated": "2020-04-23T07:38:57.230Z",

"updated": "2020-04-23T07:38:57.230Z"

}'

Here -H means it is a part of Header and -d means it is a part of Request Body

4) If the function has been deployed to the google cloud console then we can also use below technique to check the cloud function deployed:

Go to the PostMan and create a Post Url with the following details

curl https://australia-southeast2-cloud-spanner-430219.cloudfunctions.net/function-bucket \ -->(Url present in Trigger section of cloud function)

-X POST \

-H "Content-Type: application/json" \

-H "ce-id: 123451234512345" \

-H "ce-specversion: 1.0" \

-H "ce-time: 2020-01-02T12:34:56.789Z" \

-H "ce-type: google.cloud.storage.object.v1.finalized" \ (Specify Operation when to trigger, finalized -> means when we upload a new file in a bucket)

-H "ce-source: //storage.googleapis.com/projects/\_/buckets/MY-BUCKET-NAME(Bucket Name)" \

-H "ce-subject: objects/MY\_FILE.txt(Name of the file with extension)" \

-d '{

"bucket": "MY\_BUCKET", --> (Bucket Name)

"contentType": "text/plain",(file type where to do the operation)

"kind": "storage#object",

"md5Hash": "...",

"metageneration": "1",

"name": "MY\_FILE.txt", (Name of the file)

"size": "352",

"storageClass": "MULTI\_REGIONAL",

"timeCreated": "2020-04-23T07:38:57.230Z",

"timeStorageClassUpdated": "2020-04-23T07:38:57.230Z",

"updated": "2020-04-23T07:38:57.230Z"

}'

You can see the output in Logs section of cloud function deployed, if we are using Logger class for logging the output.

5) Use the below command in Google Cloud SDK command console to deploy the function to google cloud as cloud function.

gcloud functions deploy my-bucket-function(bucket name) --gen2 --region=us-east1(region name) --runtime=java17(programming language) --source=.(location of the directory in local machine where pom.xml file is present) --entry-point=com.example.CloudFunctionBucket.CloudFunctionBucketApplication(fully qualified name of the function/class) --trigger-event-filters="type=google.cloud.storage.object.v1.finalized"(operation type here finalized means upload new file in bucket) --trigger-event-filters="bucket=justicebucket"(name of the bucket) --trigger-location=us (location of trigger, optional parameter)