Triggering FME jobs with AWS Lambda

Introduction

When a new file is uploaded to AWS S3, trigger a Lambda function passes the file name/bucket/path to an FME workspace that will use the S3Connector transformer to download the file to the FME Server Temp folder.

Prerequisites

- AWS account, S3 bucket
- Accessible FME Server
- FME desktop

Steps

1. Create the FME workspace
    a. Add a creator. Requires no configuration.
    b. Add a S3Connector and connect the output of the creator to the input of the S3Connector.
    c. Configure the S3Connector authentication: change the credential source to "Web Connection", add a web connection using an AWS access/secret key, and specify the region.
    d. Configure the S3Connector request: change the action to "Download" and set up the Bucket and Path to be User Parameters.
    e. Configure the S3Connector destination: change the "Download as" to "File" and "Download to Folder" to your preferred location (FME_SHAREDRESOURCE_TEMP in this example).
2. Publish the workspace
    a. Publish the workspace from the FME desktop application to ensure the requisite packages are published alongside the workspace.
3. Create the FME Server Automation

   a. Add a webhook trigger. Take note of the webhook URL once the automation is saved.
   b. On the Output Keys tab, add two webhook keys for Bucket and File. Take note of the key names.
   c. Add a workspace action and connect the workspace action to the success port of the webhook trigger.
   d. Set the workspace to the workspace you published earlier.
   e. Set the published parameters to match the webhook output keys.
   f. Save and start the automation.

4. Create the python script
   a. The code is provided in `lambda_function.py`.
   b. Replace the webhook URL on line 13 with the URL from step 3a.
   c. Ensure that the dictionary keys on line 17 match the webhook keys from step 3b.

5. Deploy the script to AWS Lambda
   a. Open the Functions page on the Lambda console.
   b. Choose “Create function” and then “Use a blueprint”.
   c. Under Blueprints, enter s3 in the search box and choose “s3-get-object-python”.
   d. Choose Configure.
   e. Enter a function name.
   f. For Execution role, choose “Create a new role from AWS policy templates” and enter a role name
   g. Under S3 trigger, choose the S3 bucket that you created previously.
   h. When you configure an S3 trigger using the Lambda console, the console modifies your function's resource-based policy to allow Amazon S3 to invoke the function.
   i. Choose Create function (we will change the python to our script later)
   j. Go to the Lambda console and select our newly created function.
   k. Copy the contents of the python script created in step 4

l. Navigate to the lambda function .py file and paste over it.
m. Click deploy.