SAM: Local Debug & Testing

Goals:

- Learn more about how debug your serverless application's logic
  - NOTE: Use python 3.9... python 3.6 is dead now for lambda
  - NOTE: Use Region of Ohio / us-east-2 in Student Accounts

Dependencies:

- Access to the Student Environment in AWS
- Cloud9 IDE was created previously, see previous lab entitled: "Cloud9 & SAM 101"
- Understanding the content within the lab: "HTTP GET Parameters"

Code & Files:

- https://github.com/TweekFawkes/train_intro_to_serverless

Login to the Student AWS Account

- AWS Login: https://console.aws.amazon.com/ (Links to an external site.)
- IAM Username: Hal
- IAM Password: <password>

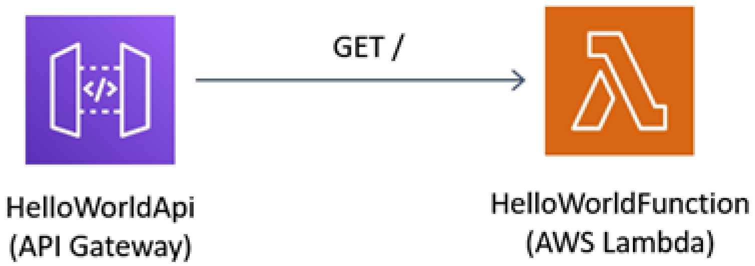
Login to the Cloud9 IDE Environment

Region: Ohio / us-east-2
Service: Cloud9

Locate the "HelloWorld101" Cloud9 environment
Click the "Open IDE" button

Download the Sample SAM App

We will build a simple SAM app with the following components:



In the terminal, run the following command(s) to create a new sam application:

```
cd ~/environment/

sam init

1

1

N

13

1

N

debug-app-001
```

We should see output similar to the following:

```
Hal:/ $ cd ~/environment/

Hal:~/environment $ sam init

```

```
You can preselect a particular runtime or package type when using the `sam
init` experience.
Call `sam init --help` to learn more.

Which template source would you like to use?
        1 - AWS Quick Start Templates
        2 - Custom Template Location
Choice: 1

Choose an AWS Quick Start application template
        1 - Hello World Example
        2 - Multi-step workflow
        3 - Serverless API
        4 - Scheduled task
        5 - Standalone function
        6 - Data processing
        7 - Infrastructure event management
        8 - Lambda EFS example
        9 - Machine Learning
Template: 1

Use the most popular runtime and package type? (Python and zip) [y/N]: N

Which runtime would you like to use?
        1 - dotnet6
        2 - dotnet5.0
        3 - dotnetcore3.1
        4 - go1.x
        5 - graalvm.java11 (provided.al2)
        6 - graalvm.java17 (provided.al2)
        7 - java11
        8 - java8.al2
        9 - java8
        10 - nodejs16.x
        11 - nodejs14.x
        12 - nodejs12.x
        13 - python3.9
        14 - python3.8
        15 - python3.7
        16 - ruby2.7
        17 - rust (provided.al2)
Runtime: 13
```

```
What package type would you like to use?
        1 - Zip
        2 - Image
Package type: 1

Based on your selections, the only dependency manager available is pip.
We will proceed copying the template using pip.

Would you like to enable X-Ray tracing on the function(s) in your
application?  [y/N]: N

Project name [sam-app]: debug-app-001

Cloning from https://github.com/aws/aws-sam-cli-app-templates (process may
take a moment)

    -----------------------
    Generating application:
    -----------------------
    Name: debug-app-001
    Runtime: python3.9
    Architectures: x86_64
    Dependency Manager: pip
    Application Template: hello-world
    Output Directory: .

    Next steps can be found in the README file at ./debug-app-001/README.md


    Commands you can use next
    =========================
    [*] Create pipeline: cd debug-app-001 && sam pipeline init --bootstrap
    [*] Validate SAM template: sam validate
    [*] Test Function in the Cloud: sam sync --stack-name {stack-name}
--watch

Hal:~/environment $
```

Inspect the source code of the following files:

- app.py -> /home/ubuntu/environment/debug-app-001/hello_world/app.py
    - Contains the logic/code for your lambda application

Passing Values via HTTP GET Params

Let's modify this apps source code slightly so we can get an more in-depth understanding of how these AWS services are working under the hood...

Change the following code segment...

```
return {
"statusCode": 200,
"body": json.dumps({
"message": "hello world",
# "location": ip.text.replace("\n", "")
}),
}
```

...to the following code...

```
return {
"statusCode": 200,
"body": str(event),
}
```

Save your changes to the code via clicking "File" and then clicking the "Save" link.

Local Debugging

We can add a main function to our app.py python application, and pass it an example "event" object so that we can quickly test our python code locally moving forward...

Add the following code to the bottom of the app.py file...

```
if __name__=="__main__":
event = dict({'resource': '/hello', 'path': '/hello/', 'httpMethod': 'GET',
'headers': {'Accept': '*/*', 'CloudFront-Forwarded-Proto': 'https',
'CloudFront-Is-Desktop-Viewer': 'true', 'CloudFront-Is-Mobile-Viewer':
'false', 'CloudFront-Is-SmartTV-Viewer': 'false',
'CloudFront-Is-Tablet-Viewer': 'false', 'CloudFront-Viewer-Country': 'US',
'Host': 'jiy58cz051.execute-api.us-east-1.amazonaws.com', 'User-Agent':
'curl/7.58.0', 'Via': '2.0 6ff4697c5089876d94430beacc9a4d5e.cloudfront.net
(CloudFront)', 'X-Amz-Cf-Id':
'N2AvPGKjnYO1pmEAEiw9WUFoDpVLAJZJLEir4IVYPiJ1CkCSOncd6Q==',
'X-Amzn-Trace-Id': 'Root=1-614cfd31-70f86876714f678132ccec87',
'X-Forwarded-For': '3.237.255.37, 130.176.133.131', 'X-Forwarded-Port':
'443', 'X-Forwarded-Proto': 'https'}, 'multiValueHeaders': {'Accept':
['*/*'], 'CloudFront-Forwarded-Proto': ['https'],
'CloudFront-Is-Desktop-Viewer': ['true'], 'CloudFront-Is-Mobile-Viewer':
['false'], 'CloudFront-Is-SmartTV-Viewer': ['false'],
'CloudFront-Is-Tablet-Viewer': ['false'], 'CloudFront-Viewer-Country':
['US'], 'Host': ['jiy58cz051.execute-api.us-east-1.amazonaws.com'],
'User-Agent': ['curl/7.58.0'], 'Via': ['2.0
6ff4697c5089876d94430beacc9a4d5e.cloudfront.net (CloudFront)'],
'X-Amz-Cf-Id':
['N2AvPGKjnYO1pmEAEiw9WUFoDpVLAJZJLEir4IVYPiJ1CkCSOncd6Q=='],
'X-Amzn-Trace-Id': ['Root=1-614cfd31-70f86876714f678132ccec87'],
'X-Forwarded-For': ['3.237.255.37, 130.176.133.131'], 'X-Forwarded-Port':
['443'], 'X-Forwarded-Proto': ['https']}, 'queryStringParameters': {'AAAA':
'BBBB'}, 'multiValueQueryStringParameters': {'AAAA': ['BBBB']},
'pathParameters': None, 'stageVariables': None, 'requestContext':
{'resourceId': '8978if', 'resourcePath': '/hello', 'httpMethod': 'GET',
'extendedRequestId': 'GIx_yHemoAMFZPg=', 'requestTime':
'23/Sep/2021:22:18:25 +0000', 'path': '/Prod/hello/', 'accountId':
'580299357056', 'protocol': 'HTTP/1.1', 'stage': 'Prod', 'domainPrefix':
'jiy58cz051', 'requestTimeEpoch': 1632435505617, 'requestId':
'a8ad1156-d894-46c2-8c6d-c54a058ed420', 'identity':
{'cognitoIdentityPoolId': None, 'accountId': None, 'cognitoIdentityId':
None, 'caller': None, 'sourceIp': '3.237.255.37', 'principalOrgId': None,
'accessKey': None, 'cognitoAuthenticationType': None,
'cognitoAuthenticationProvider': None, 'userArn': None, 'userAgent':
'curl/7.58.0', 'user': None}, 'domainName':
'jiy58cz051.execute-api.us-east-1.amazonaws.com', 'apiId': 'jiy58cz051'},
'body': None, 'isBase64Encoded': False})
context = ''
lambda_handler(event, context)
```

Now we add a print() function above the return to ensure everything is working as expected...

```
print(str(event))
```

```
return {
```

Now we can run this application locally...

```
cd /home/ubuntu/environment/debug-app-001/hello_world

python3 app.py
```

We should see output similar to the following...

```
Hal:~/environment $ cd /home/ubuntu/environment/debug-app-001/hello_world

Hal:~/environment/debug-app-001/hello_world $ python3 app.py

{'resource': '/hello', 'path': '/hello/', 'httpMethod': 'GET', 'headers':
{'Accept': '*/*', 'CloudFront-Forwarded-Proto': 'https',
'CloudFront-Is-Desktop-Viewer': 'true', 'CloudFront-Is-Mobile-Viewer':
'false', 'CloudFront-Is-SmartTV-Viewer': 'false',
'CloudFront-Is-Tablet-Viewer': 'false', 'CloudFront-Viewer-Country': 'US',
'Host': 'jiy58cz051.execute-api.us-east-1.amazonaws.com', 'User-Agent':
'curl/7.58.0', 'Via': '2.0 6ff4697c5089876d94430beacc9a4d5e.cloudfront.net
(CloudFront)', 'X-Amz-Cf-Id':
'N2AvPGKjnYO1pmEAEiw9WUFoDpVLAJZJLEir4IVYPiJ1CkCSOncd6Q==',
'X-Amzn-Trace-Id': 'Root=1-614cfd31-70f86876714f678132ccec87',
'X-Forwarded-For': '3.237.255.37, 130.176.133.131', 'X-Forwarded-Port':
'443', 'X-Forwarded-Proto': 'https'}, 'multiValueHeaders': {'Accept':
['*/*'], 'CloudFront-Forwarded-Proto': ['https'],
'CloudFront-Is-Desktop-Viewer': ['true'], 'CloudFront-Is-Mobile-Viewer':
['false'], 'CloudFront-Is-SmartTV-Viewer': ['false'],
'CloudFront-Is-Tablet-Viewer': ['false'], 'CloudFront-Viewer-Country':
['US'], 'Host': ['jiy58cz051.execute-api.us-east-1.amazonaws.com'],
'User-Agent': ['curl/7.58.0'], 'Via': ['2.0
6ff4697c5089876d94430beacc9a4d5e.cloudfront.net (CloudFront)'],
'X-Amz-Cf-Id':
['N2AvPGKjnYO1pmEAEiw9WUFoDpVLAJZJLEir4IVYPiJ1CkCSOncd6Q=='],
'X-Amzn-Trace-Id': ['Root=1-614cfd31-70f86876714f678132ccec87'],
'X-Forwarded-For': ['3.237.255.37, 130.176.133.131'], 'X-Forwarded-Port':
['443'], 'X-Forwarded-Proto': ['https']}, 'queryStringParameters': {'AAAA':
'BBBB'}, 'multiValueQueryStringParameters': {'AAAA': ['BBBB']},
'pathParameters': None, 'stageVariables': None, 'requestContext':
{'resourceId': '8978if', 'resourcePath': '/hello', 'httpMethod': 'GET',
'extendedRequestId': 'GIx_yHemoAMFZPg=', 'requestTime':
'23/Sep/2021:22:18:25 +0000', 'path': '/Prod/hello/', 'accountId':
'580299357056', 'protocol': 'HTTP/1.1', 'stage': 'Prod', 'domainPrefix':
'jiy58cz051', 'requestTimeEpoch': 1632435505617, 'requestId':
'a8ad1156-d894-46c2-8c6d-c54a058ed420', 'identity':
{'cognitoIdentityPoolId': None, 'accountId': None, 'cognitoIdentityId':
None, 'caller': None, 'sourceIp': '3.237.255.37', 'principalOrgId': None,
'accessKey': None, 'cognitoAuthenticationType': None,
'cognitoAuthenticationProvider': None, 'userArn': None, 'userAgent':
'curl/7.58.0', 'user': None}, 'domainName':
'jiy58cz051.execute-api.us-east-1.amazonaws.com', 'apiId': 'jiy58cz051'},
'body': None, 'isBase64Encoded': False}

Hal:~/environment/debug-app-001/hello_world $
```

Build the App

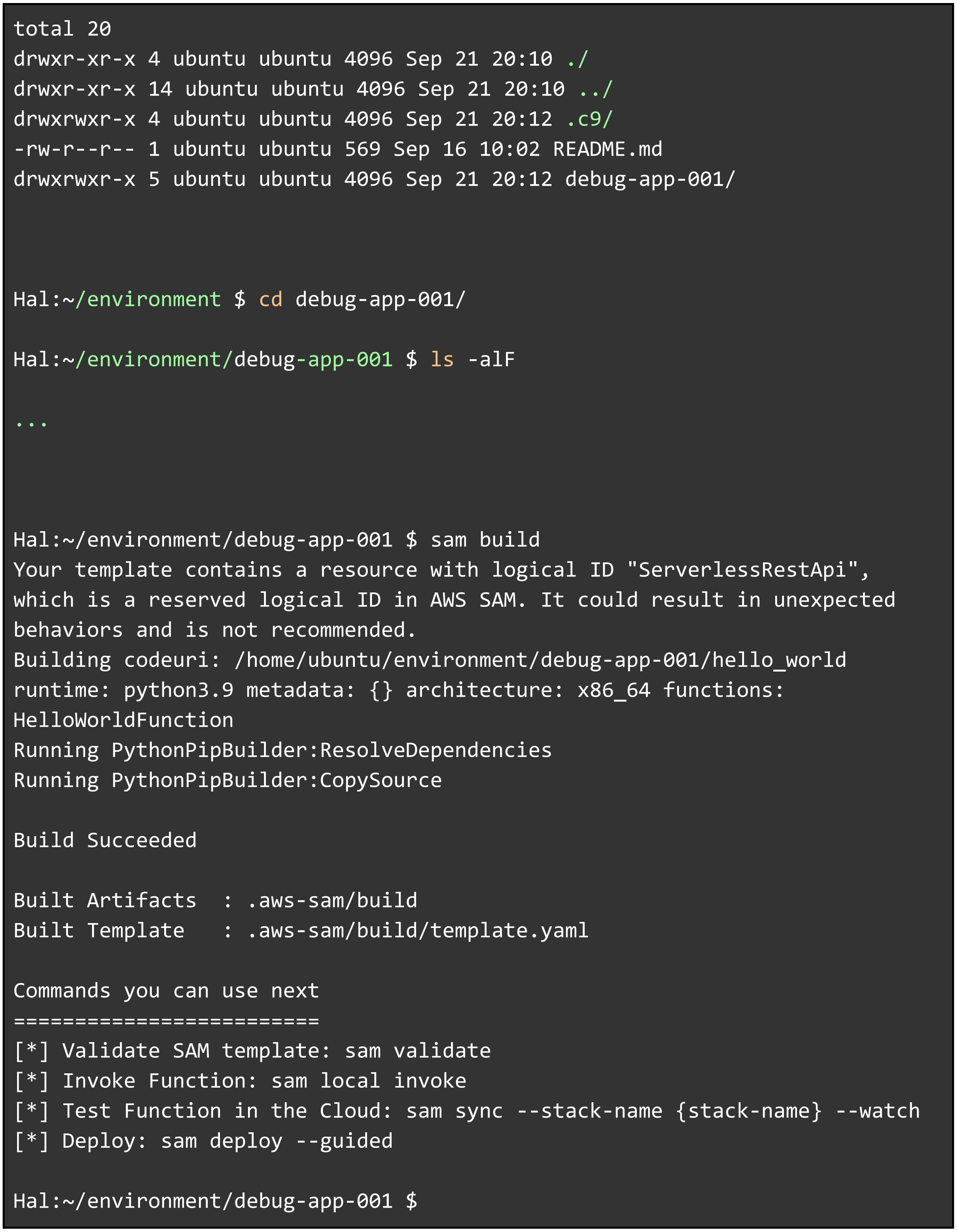
Change into the directory with the template.yaml file and build...

In the terminal, run the following command(s):

```
pwd

ls -alF

cd debug-app-001

ls -alF

sam build
```

We should see output similar to the following:

```
Hal:~/environment $ pwd
/home/ubuntu/environment


Hal:~/environment $ ls -alF
```



```
total 20
drwxr-xr-x 4 ubuntu ubuntu 4096 Sep 21 20:10 ./
drwxr-xr-x 14 ubuntu ubuntu 4096 Sep 21 20:10 ../
drwxrwxr-x 4 ubuntu ubuntu 4096 Sep 21 20:12 .c9/
-rw-r--r-- 1 ubuntu ubuntu 569 Sep 16 10:02 README.md
drwxrwxr-x 5 ubuntu ubuntu 4096 Sep 21 20:12 debug-app-001/



Hal:~/environment $ cd debug-app-001/

Hal:~/environment/debug-app-001 $ ls -alF

...



Hal:~/environment/debug-app-001 $ sam build
Your template contains a resource with logical ID "ServerlessRestApi",
which is a reserved logical ID in AWS SAM. It could result in unexpected
behaviors and is not recommended.
Building codeuri: /home/ubuntu/environment/debug-app-001/hello_world
runtime: python3.9 metadata: {} architecture: x86_64 functions:
HelloWorldFunction
Running PythonPipBuilder:ResolveDependencies
Running PythonPipBuilder:CopySource

Build Succeeded

Built Artifacts  : .aws-sam/build
Built Template   : .aws-sam/build/template.yaml

Commands you can use next
=========================
[*] Validate SAM template: sam validate
[*] Invoke Function: sam local invoke
[*] Test Function in the Cloud: sam sync --stack-name {stack-name} --watch
[*] Deploy: sam deploy --guided

Hal:~/environment/debug-app-001 $
```

This will build any dependencies and then copy your source code to the ".aws-sam/build" directory to be packaged up into a ZIP file, which will be uploaded to Lambda and S3.

Local Testing via Locally Hosted API

We can also test our application locally using features built into SAM...

The "sam local" command will run the application using docker to simulate the execution environment of an api gateway and lambda deployment.

We can host our API locally via clicking the "+" button and then clicking "New Terminal"...

And then running the following command:

```
cd /home/ubuntu/environment/debug-app-001

sam local start-api
```

We should see output similar to the following:

```
Hal:~/environment $ cd /home/ubuntu/environment/debug-app-001

Hal:~/environment/debug-app-001 $ sam local start-api

Mounting HelloWorldFunction at http://127.0.0.1:3000/hello [GET]
You can now browse to the above endpoints to invoke your functions. You do
not need to restart/reload SAM CLI while working on your functions, changes
will be reflected instantly/automatically. You only need to restart SAM CLI
if you update your AWS SAM template
2022-10-27 17:05:58  * Running on http://127.0.0.1:3000/ (Press CTRL+C to
quit)
```

Now if we leave that running and flip back to our original terminal tab, we should be able to run the following command:

```
curl http://127.0.0.1:3000/hello?AAAA=BBBB
```

We should see output similar to the following:

```
Hal:~/environment $ curl http://127.0.0.1:3000/hello?AAAA=BBBB
{"message": "hello wourl http://127.0.0.1:3000/hello?AAAA=BBBB
```

```
{'body': None, 'headers': {'Accept': '*/*', 'Host': '127.0.0.1:3000',
'User-Agent': 'curl/7.58.0', 'X-Forwarded-Port': '3000',
'X-Forwarded-Proto': 'http'}, 'httpMethod': 'GET', 'isBase64Encoded':
False, 'multiValueHeaders': {'Accept': ['*/*'], 'Host': ['127.0.0.1:3000'],
'User-Agent': ['curl/7.58.0'], 'X-Forwarded-Port': ['3000'],
'X-Forwarded-Proto': ['http']}, 'multiValueQueryStringParameters': {'AAAA':
['BBBB']}, 'path': '/hello', 'pathParameters': None,
'queryStringParameters': {'AAAA': 'BBBB'}, 'requestContext': {'accountId':
'123456789012', 'apiId': '1234567890', 'domainName': '127.0.0.1:3000',
'extendedRequestId': None, 'httpMethod': 'GET', 'identity': {'accountId':
None, 'apiKey': None, 'caller': None, 'cognitoAuthenticationProvider':
None, 'cognitoAuthenticationType': None, 'cognitoIdentityPoolId': None,
'sourceIp': '127.0.0.1', 'user': None, 'userAgent': 'Custom User Agent
String', 'userArn': None}, 'path': '/hello', 'protocol': 'HTTP/1.1',
'requestId': '9c708d38-758d-4aec-9f7e-9590fb77771c', 'requestTime':
'27/Oct/2022:17:08:12 +0000', 'requestTimeEpoch': 1666890492, 'resourceId':
'123456', 'resourcePath': '/hello', 'stage': 'Prod'}, 'resource': '/hello',
'stageVariables': None, 'version': '1.0'}

Hal:~/environment $
```

We should also see some logs from the request in the new terminal tab, that look similar to the following now...

```
Hal:~/environment/debug-app-001 $ sam local start-api

Mounting HelloWorldFunction at http://127.0.0.1:3000/hello [GET]
You can now browse to the above endpoints to invoke your functions. You do
not need to restart/reload SAM CLI while working on your functions, changes
will be reflected instantly/automatically. You only need to restart SAM CLI
if you update your AWS SAM template
2022-10-27 17:08:12  * Running on http://127.0.0.1:3000/ (Press CTRL+C to
quit)
Invoking app.lambda_handler (python3.9)
Skip pulling image and use local one:
public.ecr.aws/sam/emulation-python3.9:rapid-1.57.0-x86_64.

Mounting
/home/ubuntu/environment/debug-app-001/.aws-sam/build/HelloWorldFunction as
/var/task:ro,delegated inside runtime container
START RequestId: 2ff65a97-3066-42a9-8ecf-693ee6cf6c32 Version: $LATEST
{'body': None, 'headers': {'Accept': '*/*', 'Host': '127.0.0.1:3000',
'User-Agent': 'curl/7.58.0', 'X-Forwarded-Port': '3000',
'X-Forwarded-Proto': 'http'}, 'httpMethod': 'GET', 'isBase64Encoded':
False, 'multiValueHeaders': {'Accept': ['*/*'], 'Host': ['127.0.0.1:3000'],
'User-Agent': ['curl/7.58.0'], 'X-Forwarded-Port': ['3000'],
'X-Forwarded-Proto': ['http']}, 'multiValueQueryStringParameters': {'AAAA':
['BBBB']}, 'path': '/hello', 'pathParameters': None,
'queryStringParameters': {'AAAA': 'BBBB'}, 'requestContext': {'accountId':
'123456789012', 'apiId': '1234567890', 'domainName': '127.0.0.1:3000',
'extendedRequestId': None, 'httpMethod': 'GET', 'identity': {'accountId':
None, 'apiKey': None, 'caller': None, 'cognitoAuthenticationProvider':
None, 'cognitoAuthenticationType': None, 'cognitoIdentityPoolId': None,
'sourceIp': '127.0.0.1', 'user': None, 'userAgent': 'Custom User Agent
String', 'userArn': None}, 'path': '/hello', 'protocol': 'HTTP/1.1',
'requestId': '9c708d38-758d-4aec-9f7e-9590fb77771c', 'requestTime':
'27/Oct/2022:17:08:12 +0000', 'requestTimeEpoch': 1666890492, 'resourceId':
'123456', 'resourcePath': '/hello', 'stage': 'Prod'}, 'resource': '/hello',
'stageVariables': None, 'version': '1.0'}
END RequestId: 2ff65a97-3066-42a9-8ecf-693ee6cf6c32
REPORT RequestId: 2ff65a97-3066-42a9-8ecf-693ee6cf6c32  Init Duration: 0.18
ms  Duration: 85.50 ms      Billed Duration: 86 ms  Memory Size: 128 MB
Max Memory Used: 128 MB
No Content-Type given. Defaulting to 'application/json'.
2022-10-27 17:08:16 127.0.0.1 - - [27/Oct/2022 17:08:16] "GET
/hello?AAAA=BBBB HTTP/1.1" 200 -
```

Local Testing via Lambda Function Execution

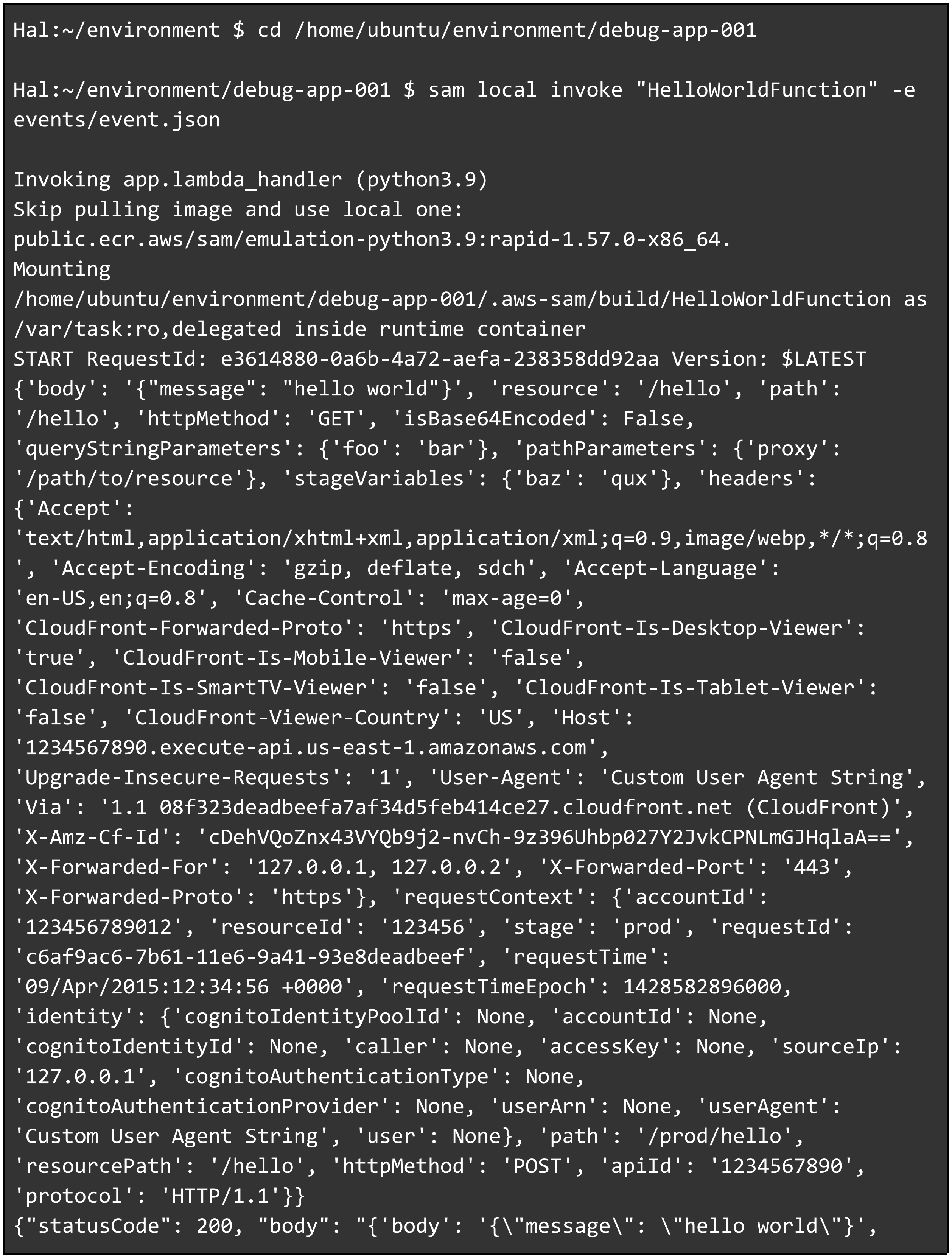
We can also test our application locally using features built into SAM...

The invoke command directly invokes your Lambda functions, and can pass input event payloads that you provide. With this command, you pass the event payload in the file event.json that the sample application provides.

We can test this application locally via the following commands...

```
cd /home/ubuntu/environment/debug-app-001

sam local invoke "HelloWorldFunction" -e events/event.json
```

We should see output similar to the following:



```
Hal:~/environment $ cd /home/ubuntu/environment/debug-app-001

Hal:~/environment/debug-app-001 $ sam local invoke "HelloWorldFunction" -e
events/event.json

Invoking app.lambda_handler (python3.9)
Skip pulling image and use local one:
public.ecr.aws/sam/emulation-python3.9:rapid-1.57.0-x86_64.
Mounting
/home/ubuntu/environment/debug-app-001/.aws-sam/build/HelloWorldFunction as
/var/task:ro,delegated inside runtime container
START RequestId: e3614880-0a6b-4a72-aefa-238358dd92aa Version: $LATEST
{'body': '{"message": "hello world"}', 'resource': '/hello', 'path':
'/hello', 'httpMethod': 'GET', 'isBase64Encoded': False,
'queryStringParameters': {'foo': 'bar'}, 'pathParameters': {'proxy':
'/path/to/resource'}, 'stageVariables': {'baz': 'qux'}, 'headers':
{'Accept':
'text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
', 'Accept-Encoding': 'gzip, deflate, sdch', 'Accept-Language':
'en-US,en;q=0.8', 'Cache-Control': 'max-age=0',
'CloudFront-Forwarded-Proto': 'https', 'CloudFront-Is-Desktop-Viewer':
'true', 'CloudFront-Is-Mobile-Viewer': 'false',
'CloudFront-Is-SmartTV-Viewer': 'false', 'CloudFront-Is-Tablet-Viewer':
'false', 'CloudFront-Viewer-Country': 'US', 'Host':
'1234567890.execute-api.us-east-1.amazonaws.com',
'Upgrade-Insecure-Requests': '1', 'User-Agent': 'Custom User Agent String',
'Via': '1.1 08f323deadbeefa7af34d5feb414ce27.cloudfront.net (CloudFront)',
'X-Amz-Cf-Id': 'cDehVQoZnx43VYQb9j2-nvCh-9z396Uhbp027Y2JvkCPNLmGJHqlaA==',
'X-Forwarded-For': '127.0.0.1, 127.0.0.2', 'X-Forwarded-Port': '443',
'X-Forwarded-Proto': 'https'}, 'requestContext': {'accountId':
'123456789012', 'resourceId': '123456', 'stage': 'prod', 'requestId':
'c6af9ac6-7b61-11e6-9a41-93e8deadbeef', 'requestTime':
'09/Apr/2015:12:34:56 +0000', 'requestTimeEpoch': 1428582896000,
'identity': {'cognitoIdentityPoolId': None, 'accountId': None,
'cognitoIdentityId': None, 'caller': None, 'accessKey': None, 'sourceIp':
'127.0.0.1', 'cognitoAuthenticationType': None,
'cognitoAuthenticationProvider': None, 'userArn': None, 'userAgent':
'Custom User Agent String', 'user': None}, 'path': '/prod/hello',
'resourcePath': '/hello', 'httpMethod': 'POST', 'apiId': '1234567890',
'protocol': 'HTTP/1.1'}}
{"statusCode": 200, "body": "{'body': '{\"message\": \"hello world\"}',
'resource': '/hello', 'path': '/hello', 'httpMethod': 'GET',
'isBase64Encoded': False, 'queryStringParameters': {'foo': 'bar'},
'pathParameters': {'proxy': '/path/to/resource'}, 'stageVariables': {'baz':
'qux'}, 'headers': {'Accept':
'text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
', 'Accept-Encoding': 'gzip, deflate, sdch', 'Accept-Language':
'en-US,en;q=0.8', 'Cache-Control': 'max-age=0',
'CloudFront-Forwarded-Proto': 'https', 'CloudFront-Is-Desktop-Viewer':
'true', 'CloudFront-Is-Mobile-Viewer': 'false',
'CloudFront-Is-SmartTV-Viewer': 'false', 'CloudFront-Is-Tablet-Viewer':
'false', 'CloudFront-Viewer-Country': 'US', 'Host':
'1234567890.execute-api.us-east-1.amazonaws.com',
'Upgrade-Insecure-Requests': '1', 'User-Agent': 'Custom User Agent String',
'Via': '1.1 08f323deadbeefa7af34d5feb414ce27.cloudfront.net (CloudFront)',
'X-Amz-Cf-Id': 'cDehVQoZnx43VYQb9j2-nvCh-9z396Uhbp027Y2JvkCPNLmGJHqlaA==',
'X-Forwarded-For': '127.0.0.1, 127.0.0.2', 'X-Forwarded-Port': '443',
'X-Forwarded-Proto': 'https'}, 'requestContext': {'accountId':
'123456789012', 'resourceId': '123456', 'stage': 'prod', 'requestId':
'c6af9ac6-7b61-11e6-9a41-93e8deadbeef', 'requestTime':
'09/Apr/2015:12:34:56 +0000', 'requestTimeEpoch': 1428582896000,
'identity': {'cognitoIdentityPoolId': None, 'accountId': None,
'cognitoIdentityId': None, 'caller': None, 'accessKey': None, 'sourceIp':
'127.0.0.1', 'cognitoAuthenticationType': None,
'cognitoAuthenticationProvider': None, 'userArn': None, 'userAgent':
'Custom User Agent String', 'user': None}, 'path': '/prod/hello',
'resourcePath': '/hello', 'httpMethod': 'POST', 'apiId': '1234567890',
'protocol': 'HTTP/1.1'}}"}END RequestId:
e3614880-0a6b-4a72-aefa-238358dd92aa
REPORT RequestId: e3614880-0a6b-4a72-aefa-238358dd92aa  Init Duration: 1.00
ms  Duration: 102.18 ms     Billed Duration: 103 ms Memory Size: 128 MB
Max Memory Used: 128 MB

Hal:~/environment/debug-app-001 $
```

The default template comes with a default "aws-proxy" event for the aws api gateway service.

We can generate our own input event via the commands:

```
cd /home/ubuntu/environment/debug-app-001

sam local generate-event apigateway aws-proxy --body "" --path "hello"
--method GET > /home/ubuntu/environment/debug-app-001/events/api-event.json
```

We can add GET parameters to this file via opening the "/home/ubuntu/environment/debug-app-001/events/api-event.json" file within cloud9 and modifying it's content in the following section from...

```
"queryStringParameters": {
"foo": "bar"
},
```

...to…

```
"queryStringParameters": {
"AAAA": "BBBB"
},
```

We can then text this via the following commands:

```
cd /home/ubuntu/environment/debug-app-001

sam local invoke "HelloWorldFunction" -e events/api-event.json
```

We should see output similar to the following:

```
Hal:/ $ cd /home/ubuntu/environment/debug-app-001

Hal:~/environment/debug-app-001 $ sam local invoke "HelloWorldFunction" -e
events/api-event.json
Invoking app.lambda_handler (python3.9)
Skip pulling image and use local one:
public.ecr.aws/sam/emulation-python3.9:rapid-1.57.0-x86_64.

Mounting
/home/ubuntu/environment/debug-app-001/.aws-sam/build/HelloWorldFunction as
/var/task:ro,delegated inside runtime container
START RequestId: 27e2f6a3-a95f-40ee-977d-ce86bed787f6 Version: $LATEST
{'body': '', 'resource': '/{proxy+}', 'path': '/hello', 'httpMethod':
'GET', 'isBase64Encoded': True, 'queryStringParameters': {'AAAA': 'BBBB'},
'multiValueQueryStringParameters': {'foo': ['bar']}, 'pathParameters':
{'proxy': '/hello'}, 'stageVariables': {'baz': 'qux'}, 'headers':
{'Accept':
```

```
'text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
', 'Accept-Encoding': 'gzip, deflate, sdch', 'Accept-Language':
'en-US,en;q=0.8', 'Cache-Control': 'max-age=0',
'CloudFront-Forwarded-Proto': 'https', 'CloudFront-Is-Desktop-Viewer':
'true', 'CloudFront-Is-Mobile-Viewer': 'false',
'CloudFront-Is-SmartTV-Viewer': 'false', 'CloudFront-Is-Tablet-Viewer':
'false', 'CloudFront-Viewer-Country': 'US', 'Host':
'1234567890.execute-api.us-east-1.amazonaws.com',
'Upgrade-Insecure-Requests': '1', 'User-Agent': 'Custom User Agent String',
'Via': '1.1 08f323deadbeefa7af34d5feb414ce27.cloudfront.net (CloudFront)',
'X-Amz-Cf-Id': 'cDehVQoZnx43VYQb9j2-nvCh-9z396Uhbp027Y2JvkCPNLmGJHqlaA==',
'X-Forwarded-For': '127.0.0.1, 127.0.0.2', 'X-Forwarded-Port': '443',
'X-Forwarded-Proto': 'https'}, 'multiValueHeaders': {'Accept':
['text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.
8'], 'Accept-Encoding': ['gzip, deflate, sdch'], 'Accept-Language':
['en-US,en;q=0.8'], 'Cache-Control': ['max-age=0'],
'CloudFront-Forwarded-Proto': ['https'], 'CloudFront-Is-Desktop-Viewer':
['true'], 'CloudFront-Is-Mobile-Viewer': ['false'],
'CloudFront-Is-SmartTV-Viewer': ['false'], 'CloudFront-Is-Tablet-Viewer':
['false'], 'CloudFront-Viewer-Country': ['US'], 'Host':
['0123456789.execute-api.us-east-1.amazonaws.com'],
'Upgrade-Insecure-Requests': ['1'], 'User-Agent': ['Custom User Agent
String'], 'Via': ['1.1 08f323deadbeefa7af34d5feb414ce27.cloudfront.net
(CloudFront)'], 'X-Amz-Cf-Id':
['cDehVQoZnx43VYQb9j2-nvCh-9z396Uhbp027Y2JvkCPNLmGJHqlaA=='],
'X-Forwarded-For': ['127.0.0.1, 127.0.0.2'], 'X-Forwarded-Port': ['443'],
'X-Forwarded-Proto': ['https']}, 'requestContext': {'accountId':
'123456789012', 'resourceId': '123456', 'stage': 'prod', 'requestId':
'c6af9ac6-7b61-11e6-9a41-93e8deadbeef', 'requestTime':
'09/Apr/2015:12:34:56 +0000', 'requestTimeEpoch': 1428582896000,
'identity': {'cognitoIdentityPoolId': None, 'accountId': None,
'cognitoIdentityId': None, 'caller': None, 'accessKey': None, 'sourceIp':
'127.0.0.1', 'cognitoAuthenticationType': None,
'cognitoAuthenticationProvider': None, 'userArn': None, 'userAgent':
'Custom User Agent String', 'user': None}, 'path': '/prod/hello',
'resourcePath': '/{proxy+}', 'httpMethod': 'GET', 'apiId': '1234567890',
'protocol': 'HTTP/1.1'}}
{"statusCode": 200, "body": "{'body': '', 'resource': '/{proxy+}', 'path':
'/hello', 'httpMethod': 'GET', 'isBase64Encoded': True,
'queryStringParameters': {'AAAA': 'BBBB'},
'multiValueQueryStringParameters': {'foo': ['bar']}, 'pathParameters':
{'proxy': '/hello'}, 'stageVariables': {'baz': 'qux'}, 'headers':
```

```
{'Accept':
'text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
', 'Accept-Encoding': 'gzip, deflate, sdch', 'Accept-Language':
'en-US,en;q=0.8', 'Cache-Control': 'max-age=0',
'CloudFront-Forwarded-Proto': 'https', 'CloudFront-Is-Desktop-Viewer':
'true', 'CloudFront-Is-Mobile-Viewer': 'false',
'CloudFront-Is-SmartTV-Viewer': 'false', 'CloudFront-Is-Tablet-Viewer':
'false', 'CloudFront-Viewer-Country': 'US', 'Host':
'1234567890.execute-api.us-east-1.amazonaws.com',
'Upgrade-Insecure-Requests': '1', 'User-Agent': 'Custom User Agent String',
'Via': '1.1 08f323deadbeefa7af34d5feb414ce27.cloudfront.net (CloudFront)',
'X-Amz-Cf-Id': 'cDehVQoZnx43VYQb9j2-nvCh-9z396Uhbp027Y2JvkCPNLmGJHqlaA==',
'X-Forwarded-For': '127.0.0.1, 127.0.0.2', 'X-Forwarded-Port': '443',
'X-Forwarded-Proto': 'https'}, 'multiValueHeaders': {'Accept':
['text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.
8'], 'Accept-Encoding': ['gzip, deflate, sdch'], 'Accept-Language':
['en-US,en;q=0.8'], 'Cache-Control': ['max-age=0'],
'CloudFront-Forwarded-Proto': ['https'], 'CloudFront-Is-Desktop-Viewer':
['true'], 'CloudFront-Is-Mobile-Viewer': ['false'],
'CloudFront-Is-SmartTV-Viewer': ['false'], 'CloudFront-Is-Tablet-Viewer':
['false'], 'CloudFront-Viewer-Country': ['US'], 'Host':
['0123456789.execute-api.us-east-1.amazonaws.com'],
'Upgrade-Insecure-Requests': ['1'], 'User-Agent': ['Custom User Agent
String'], 'Via': ['1.1 08f323deadbeefa7af34d5feb414ce27.cloudfront.net
(CloudFront)'], 'X-Amz-Cf-Id':
['cDehVQoZnx43VYQb9j2-nvCh-9z396Uhbp027Y2JvkCPNLmGJHqlaA=='],
'X-Forwarded-For': ['127.0.0.1, 127.0.0.2'], 'X-Forwarded-Port': ['443'],
'X-Forwarded-Proto': ['https']}, 'requestContext': {'accountId':
'123456789012', 'resourceId': '123456', 'stage': 'prod', 'requestId':
'c6af9ac6-7b61-11e6-9a41-93e8deadbeef', 'requestTime':
'09/Apr/2015:12:34:56 +0000', 'requestTimeEpoch': 1428582896000,
'identity': {'cognitoIdentityPoolId': None, 'accountId': None,
'cognitoIdentityId': None, 'caller': None, 'accessKey': None, 'sourceIp':
'127.0.0.1', 'cognitoAuthenticationType': None,
'cognitoAuthenticationProvider': None, 'userArn': None, 'userAgent':
'Custom User Agent String', 'user': None}, 'path': '/prod/hello',
'resourcePath': '/{proxy+}', 'httpMethod': 'GET', 'apiId': '1234567890',
'protocol': 'HTTP/1.1'}}"}END RequestId:
27e2f6a3-a95f-40ee-977d-ce86bed787f6
REPORT RequestId: 27e2f6a3-a95f-40ee-977d-ce86bed787f6  Init Duration: 0.11
ms  Duration: 68.46 ms      Billed Duration: 69 ms  Memory Size: 128 MB
Max Memory Used: 128 MB
```

```
Hal:~/environment/debug-app-001 $
```

References

- Tutorial: Deploying a Hello World application - https://docs.aws.amazon.com/serverless-application-model/latest/developerguide/serverless-getting-started-hello-world.html
- https://docs.aws.amazon.com/serverless-application-model/latest/developerguide/sam-cli-command-reference-sam-local-generate-event.html
- https://docs.aws.amazon.com/serverless-application-model/latest/developerguide/serverless-sam-cli-using-generate-event.html
- https://stackoverflow.com/questions/60758287/how-to-generate-an-event-with-sam-local-generate-event-with-queryparameters