Local Debug & Testing

Goals:

- Learn more about how lambda accepts GET parameter values via the API gateway service

Dependencies:

- 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/Stage2Sec/CaptureTheCloud/tree/master/train_aws_sam

Login to the Student AWS Red Team Account

AWS Login: **https://console.aws.amazon.com/ (https://console.aws.amazon.com/)**

IAM Username: <red_team_###>

IAM Password: <password>

Cloud9 IDE Environment

Region: US East (N. Virginia) us-east-1

Service: Cloud9

Locate the "HelloWorld101" Cloud9 environment

Click the "Open IDE" button

In the terminal, run the following command(s) to build a sample using **python 3.6**:

```
cd ~/environment/

sam init

1

1

11

debug-app-001

1
```

We should see output similar to the following:

```
red_team_040:~/environment $ sam init
Which template source would you like to use?
1 - AWS Quick Start Templates
2 - Custom Template Location


Choice: 1


What package type would you like to use?
1 - Zip (artifact is a zip uploaded to S3)
2 - Image (artifact is an image uploaded to an ECR image repository)


Package type: 1



Which runtime would you like to use?
1 - nodejs14.x
2 - python3.9
3 - ruby2.7
4 - go1.x
5 - java11
6 - dotnetcore3.1
7 - nodejs12.x
8 - nodejs10.x
```

```
9 - python3.8
10 - python3.7
11 - python3.6
12 - python2.7
13 - ruby2.5
14 - java8.al2
15 - java8
16 - dotnetcore2.1

Runtime: 11


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


Cloning app templates from https://github.com/aws/aws-sam-cli-app-templates

AWS quick start application templates:
1 - Hello World Example
2 - EventBridge Hello World
3 - EventBridge App from scratch (100+ Event Schemas)
4 - Step Functions Sample App (Stock Trader)'

Template selection: 1


----------------------
Generating application:
----------------------
Name: debug-app-001
Runtime: python3.6
Dependency Manager: pip
Application Template: hello-world
Output Directory: .

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

red_team_040:~/environment $
```

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...

```
red_team_040:~/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}
```

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 click 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:

```
red_team_040:~/environment/debug-app-001 $ cd

red_team_040:~ $ cd /home/ubuntu/environment/debug-app-001

red_team_040:~/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

2021-09-23 22:55:04 * 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:

```
red_team_040:~/environment/debug-app-001 $ curl 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': '8ff645c2-6e13-4e7d-8ad4-fc154cf45980', 'requestTime': '23/Sep/2021:22:55:04 +0000', 'requestTimeEpoch': 1632437704, 'resourceId': '123456', 'resourcePath': '/hello', 'stage': 'Prod'}, 'resource': '/hello', 'stageVariables': None, 'version': '1.0'}


red_team_040:~/environment/debug-app-001 $
```

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

```
red_team_040:~/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

2021-09-23 22:55:04 * Running on http://127.0.0.1:3000/ (Press CTRL+C to quit)

Invoking app.lambda_handler (python3.6)
Skip pulling image and use local one: public.ecr.aws/sam/emulation-python3.6:rapid-1.31.0.
Mounting /home/ubuntu/environment/debug-app-001/hello_world as /var/task:ro,delegated inside runtime container
{'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': '8ff645c2-6e13-4e7d-8ad4-fc154cf45980', 'requestTime': '23/Sep/2021:22:55:04 +0000', 'requestTimeEpoch': 1632437704, 'resourceId': '123456', 'resourcePath': '/hello', 'stage': 'Prod'}, 'resource': '/hello', 'stageVariables': None, 'version': '1.0'}
END RequestId: bd7e38ee-e825-4aa4-93f0-24b632c93c79
REPORT RequestId: bd7e38ee-e825-4aa4-93f0-24b632c93c79 Init Duration: 2.27 ms Duration: 406.20 ms Billed Duration: 500 ms Memory Size: 128 MB Max Memory Used: 128 MB
No Content-Type given. Defaulting to 'application/json'.


2021-09-23 22:56:34 127.0.0.1 - - [23/Sep/2021 22:56:34] "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:

```
red_team_040:~ $ cd /home/ubuntu/environment/debug-app-001


red_team_040:~/environment/debug-app-001 $ sam local invoke "HelloWorldFunction" -e events/event.json
```

```
Invoking app.lambda_handler (python3.6)
Skip pulling image and use local one: public.ecr.aws/sam/emulation-python3.6:rapid-1.31.0.

Mounting /home/ubuntu/environment/debug-app-001/hello_world as /var/task:ro,delegated inside runtime container
START RequestId: b7017b89-85a4-4cc8-ad75-07f419df60f7 Version: $LATEST
{'body': '{"message": "hello world"}', 'resource': '/{proxy+}', 'path': '/path/to/resource', 'httpMethod': 'POST', '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/path/to/resource', 'resourcePath': '/{proxy+}', 'httpMethod': 'POST', 'apiId': '1234567890', 'protocol': 'HTTP/1.1'}}
END RequestId: b7017b89-85a4-4cc8-ad75-07f419df60f7
REPORT RequestId: b7017b89-85a4-4cc8-ad75-07f419df60f7 Init Duration: 1.63 ms Duration: 154.97 ms Billed Duration: 200 ms Memory Size: 128 MB Max Memory Used: 128 MB
{"statusCode": 200, "body": "{'body': '{\"message\": \"hello world\"}', 'resource': '/{proxy+}', 'path': '/path/to/resource', 'httpMethod': 'POST', '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/path/to/resource', 'resourcePath': '/{proxy+}', 'httpMethod': 'POST', 'apiId': '1234567890', 'protocol': 'HTTP/1.1'}}"}red_team_040:~/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:

```
red_team_040:~/environment/debug-app-001 $ cd /home/ubuntu/environment/debug-app-001


red_team_040:~/environment/debug-app-001 $ sam local invoke "HelloWorldFunction" -e events/api-event.json
```

```
Invoking app.lambda_handler (python3.6)
Skip pulling image and use local one: public.ecr.aws/sam/emulation-python3.6:rapid-1.31.0.

Mounting /home/ubuntu/environment/debug-app-001/hello_world as /var/task:ro,delegated inside runtime container
START RequestId: 464c9ca0-af9f-451d-8c95-5ce0c38ae8df 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'}}
END RequestId: 464c9ca0-af9f-451d-8c95-5ce0c38ae8df
REPORT RequestId: 464c9ca0-af9f-451d-8c95-5ce0c38ae8df Init Duration: 0.66 ms Duration: 175.87 ms Billed Duration: 200 ms Memory Size: 128 MB Max Memory Used: 128 MB
{"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'}}"}

red_team_040:~/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/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/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://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 (https://stackoverflow.com/questions/60758287/how-to-generate-an-event-with-sam-local-generate-event-with-queryparameters)