HTTP GET Parameters

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"

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**:

```
sam init

1

1

11

getparam-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]: getparam-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: getparam-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 $
```

Passing Values via HTTP GET Params

Inspect the source code of the following files:

- app.py -> /home/ubuntu/environment/helloworld-app-001/port_check/app.py

-- Contains the logic/code for your lambda application

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.

Build

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

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

```
cd ~/environment/

pwd

ls -alF
```

```
cd getparam-app-001

ls -alF


sam build
```

We should see output similar to the following:

```
red_team_040:~ $ cd ~/environment/


red_team_040:~/environment $ pwd
/home/ubuntu/environment


red_team_040:~/environment $ ls -alF
...
drwxrwxr-x 5 ubuntu ubuntu 4096 Sep 23 22:02 getparam-app-001/
...


red_team_040:~/environment $ cd getparam-app-001/


red_team_040:~/environment/getparam-app-001 $ ls -alF
total 40
drwxrwxr-x 5 ubuntu ubuntu 4096 Sep 23 22:02 ./
drwxr-xr-x 7 ubuntu ubuntu 4096 Sep 23 22:02 ../
-rw-rw-r-- 1 ubuntu ubuntu 3730 Sep 23 22:02 .gitignore
-rw-rw-r-- 1 ubuntu ubuntu 8230 Sep 23 22:02 README.md
-rw-rw-r-- 1 ubuntu ubuntu 0 Sep 23 22:02 __init__.py
drwxrwxr-x 2 ubuntu ubuntu 4096 Sep 23 22:02 events/
drwxrwxr-x 2 ubuntu ubuntu 4096 Sep 23 22:10 hello_world/
-rw-rw-r-- 1 ubuntu ubuntu 1641 Sep 23 22:02 template.yaml
drwxrwxr-x 3 ubuntu ubuntu 4096 Sep 23 22:02 tests/



red_team_040:~/environment/getparam-app-001 $ sam build


Building codeuri: /home/ubuntu/environment/getparam-app-001/hello_world runtime: python3.6
metadata: {} 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
=========================
[*] Invoke Function: sam local invoke
[*] Deploy: sam deploy --guided


red_team_040:~/environment/getparam-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.

Deploy the SAM App

Next, we will deploy our new SAM App! :)

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

```
sam deploy --guided



getparam-app-001

[ENTER]

y

Y

y

Y

[ENTER]

[ENTER]

y
```

We should see output similar to the following:

```
red_team_040:~/environment/getparam-app-001 $ sam deploy --guided

Configuring SAM deploy
======================

Looking for config file [samconfig.toml] : Not found

Setting default arguments for 'sam deploy'
=========================================
Stack Name [sam-app]: getparam-app-001
AWS Region [us-east-1]:
#Shows you resources changes to be deployed and require a 'Y' to initiate deploy
Confirm changes before deploy [y/N]: y
#SAM needs permission to be able to create roles to connect to the resources in your template
Allow SAM CLI IAM role creation [Y/n]: y
HelloWorldFunction may not have authorization defined, Is this okay? [y/N]: y
Save arguments to configuration file [Y/n]: y
SAM configuration file [samconfig.toml]:
SAM configuration environment [default]:

Looking for resources needed for deployment:
Managed S3 bucket: aws-sam-cli-managed-default-samclisourcebucket-1sivrgk5lqe6g
A different default S3 bucket can be set in samconfig.toml

Saved arguments to config file
Running 'sam deploy' for future deployments will use the parameters saved above.
The above parameters can be changed by modifying samconfig.toml
Learn more about samconfig.toml syntax at
https://docs.aws.amazon.com/serverless-application-model/latest/developerguide/serverless-sam-cli-config.html

Uploading to getparam-app-001/82668aa6744253308f2de772c355a03c 444816 / 444816 (100.00%)

Deploying with following values
===============================
Stack name : getparam-app-001
Region : us-east-1
Confirm changeset : True
Deployment s3 bucket : aws-sam-cli-managed-default-samclisourcebucket-1sivrgk5lqe6g
Capabilities : ["CAPABILITY_IAM"]
Parameter overrides : {}
Signing Profiles : {}
```

```
Initiating deployment
====================
Uploading to getparam-app-001/b7b4a5a705a171f4d216c0efb905fb73.template 1117 / 1117 (100.00%)

Waiting for changeset to be created..

CloudFormation stack changeset
--------------------------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------
Operation LogicalResourceId ResourceType Replacement
--------------------------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------
+ Add HelloWorldFunctionHelloWorldPermissionProd AWS::Lambda::Permission N/A
+ Add HelloWorldFunctionRole AWS::IAM::Role N/A
+ Add HelloWorldFunction AWS::Lambda::Function N/A
+ Add ServerlessRestApiDeployment47fc2d5f9d AWS::ApiGateway::Deployment N/A
+ Add ServerlessRestApiProdStage AWS::ApiGateway::Stage N/A
+ Add ServerlessRestApi AWS::ApiGateway::RestApi N/A
--------------------------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------

Changeset created successfully. arn:aws:cloudformation:us-east-1:580299357056:changeSet/samcli-deploy1632435339/4c3db263-9f91-41ad-9c78-8b6d69ae3b19


Previewing CloudFormation changeset before deployment
========================================================
Deploy this changeset? [y/N]: y

2021-09-23 22:15:52 - Waiting for stack create/update to complete

CloudFormation events from changeset
--------------------------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------
ResourceStatus ResourceType LogicalResourceId ResourceStatusReason
--------------------------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------
CREATE_IN_PROGRESS AWS::IAM::Role HelloWorldFunctionRole -
CREATE_IN_PROGRESS AWS::IAM::Role HelloWorldFunctionRole Resource creation Initiated
CREATE_COMPLETE AWS::IAM::Role HelloWorldFunctionRole -
CREATE_IN_PROGRESS AWS::Lambda::Function HelloWorldFunction -
CREATE_IN_PROGRESS AWS::Lambda::Function HelloWorldFunction Resource creation Initiated
CREATE_COMPLETE AWS::Lambda::Function HelloWorldFunction -
CREATE_IN_PROGRESS AWS::ApiGateway::RestApi ServerlessRestApi -
CREATE_IN_PROGRESS AWS::ApiGateway::RestApi ServerlessRestApi Resource creation Initiated
```

```
CREATE_COMPLETE AWS::ApiGateway::RestApi ServerlessRestApi -
CREATE_IN_PROGRESS AWS::Lambda::Permission HelloWorldFunctionHelloWorldPermissionProd
Resource creation Initiated
CREATE_IN_PROGRESS AWS::Lambda::Permission HelloWorldFunctionHelloWorldPermissionProd -
CREATE_IN_PROGRESS AWS::ApiGateway::Deployment ServerlessRestApiDeployment47fc2d5f9d -
CREATE_IN_PROGRESS AWS::ApiGateway::Deployment ServerlessRestApiDeployment47fc2d5f9d
Resource creation Initiated
CREATE_COMPLETE AWS::ApiGateway::Deployment ServerlessRestApiDeployment47fc2d5f9d -
CREATE_IN_PROGRESS AWS::ApiGateway::Stage ServerlessRestApiProdStage -
CREATE_IN_PROGRESS AWS::ApiGateway::Stage ServerlessRestApiProdStage Resource creation
Initiated
CREATE_COMPLETE AWS::ApiGateway::Stage ServerlessRestApiProdStage -
CREATE_COMPLETE AWS::Lambda::Permission HelloWorldFunctionHelloWorldPermissionProd -
CREATE_COMPLETE AWS::CloudFormation::Stack getparam-app-001 -
------------------------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------

CloudFormation outputs from deployed stack
------------------------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------
Outputs
------------------------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------
Key HelloWorldFunctionIamRole
Description Implicit IAM Role created for Hello World function
Value arn:aws:iam::580299357056:role/getparam-app-001-HelloWorldFunctionRole-8TO0EB0MTEBR

Key HelloWorldApi
Description API Gateway endpoint URL for Prod stage for Hello World function
Value https://jiy58cz051.execute-api.us-east-1.amazonaws.com/Prod/hello/

Key HelloWorldFunction
Description Hello World Lambda Function ARN
Value arn:aws:lambda:us-east-1:580299357056:function:getparam-app-001-HelloWorldFunction-
jecZHsSPzqRV
------------------------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------

Successfully created/updated stack - getparam-app-001 in us-east-1



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

Test the SAM App

We will test the SAM App...

We will want to locate the URL to our newly deployed API gateway, for example in our above output:

```
https://jiy58cz051.execute-api.us-east-1.amazonaws.com/Prod/hello/
```

We can pass the lambda function information via the API gateway as HTTP GET Parameters...

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

```
curl https://jiy58cz051.execute-api.us-east-1.amazonaws.com/Prod/hello/?AAAA=BBBB
```

We should see output similar to the following:

```
red_team_040:~/environment/getparam-app-001 $ curl https://jiy58cz051.execute-api.us-east-
1.amazonaws.com/Prod/hello/?AAAA=BBBB


{'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}


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

The "event" object will contain data similar to the above output.

We can see from this output that the GET parameter "AAAA" value of "BBBB" is contained within the following "event" object:

```
event['queryStringParameters']

event['queryStringParameters']['AAAA']
```

The "event" object is of type dictionary in python.

Clean Up via CloudFormation

SAM uses the AWS CloudFormation service to deploy resources, hence we can use the CloudFormation service to clean up the SAM application deployment. We will need to know the following information:

- #1 - Stack Name: e.g. getparam-app-001
- #2 - AWS Region: e.g. us-east-1

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

```
aws cloudformation delete-stack --stack-name getparam-app-001 --region us-east-1
```

We should see output similar to the following:

```
red_team_040:~/environment/sam-app-001 $ aws cloudformation delete-stack --stack-name getparam-
app-001 --region us-east-1
```

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

Next we can check to ensure the delete was succuessful...

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

```
aws cloudformation list-stacks
```

We should see output similar to the following:

```
red_team_040:~/environment/sam-app-001 $ aws cloudformation list-stacks
{
"StackSummaries": [
{
"StackId": "arn:aws:cloudformation:us-east-1:580299357056:stack/sam-app-001/7704fd60-1b1d-11ec-8228-0eea388cb225",
"StackName": "sam-app-001",
"TemplateDescription": "getparam-app-001\nSample SAM Template for sam-app-001\n",
"CreationTime": "2021-09-21T20:49:50.792Z",
"LastUpdatedTime": "2021-09-21T20:52:28.346Z",
"DeletionTime": "2021-09-21T21:00:06.896Z",
"StackStatus": "DELETE_COMPLETE",
"DriftInformation": {
"StackDriftStatus": "NOT_CHECKED"
}
},

...
```

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)**