# SAM: HTTP GET Parameters

## Goals:

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

### 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>

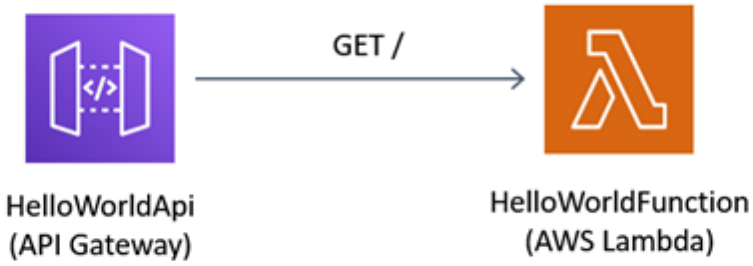
## Login to the Cloud9 IDE Environment

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

getparam-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]: getparam-app-001

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

    -----------------------
    Generating application:
    -----------------------
    Name: getparam-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
./getparam-app-001/README.md


    Commands you can use next
    =========================
    [*] Create pipeline: cd getparam-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/getparam-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.

# 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 getparam-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 getparam-app-001/



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

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

...



Hal:~/environment/getparam-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/getparam-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/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

N

y

Y

[ENTER]
```

```
[ENTER]

y
```

We should see output similar to the following:

```
Hal:~/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-2]:
        #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
        #Preserves the state of previously provisioned resources when an
operation fails
        Disable rollback [y/N]: N
        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-142o3zytl001y
         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-001/3b968f157ecd55c9bb2bc471c82379fd  466315 / 466315  (100.00%)

        Deploying with following values
        ===============================
        Stack name                   : getparam-app-001
        Region                       : us-east-2
        Confirm changeset            : True
        Disable rollback             : False
        Deployment s3 bucket         : aws-sam-cli-managed-default-samclisourcebucket-142o3zytl001y
        Capabilities                 : ["CAPABILITY_IAM"]
        Parameter overrides          : {}
        Signing Profiles             : {}

Initiating deployment
=====================
Uploading to getparam-app-001/a96e73cdd1c0b4e9e30f564e3b4c9ac7.template  1207 / 1207  (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-2:013109453517:changeSet/samcli-deploy166688
9537/a57b14e2-691f-4ff6-a5e4-63e7b20189aa


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

2022-10-27 16:52:28 - Waiting for stack create/update to complete

CloudFormation events from stack operations (refresh every 0.5 seconds)
-------------------------------------------------------------------------
-------------------------------------------------------------------------
-------------------------------------------------
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      -
CREATE_IN_PROGRESS
AWS::ApiGateway::Deployment
ServerlessRestApiDeployment47fc2d5f9d           -
CREATE_IN_PROGRESS                              AWS::Lambda::Permission
HelloWorldFunctionHelloWorldPermissionProd      Resource creation
Initiated
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::013109453517:role/getparam-app-001-HelloWorldFunctionRole-1K2A
I6987J3PE

Key                 HelloWorldApi
Description         API Gateway endpoint URL for Prod stage for Hello World
function
Value
https://0l08ii78s1.execute-api.us-east-2.amazonaws.com/Prod/hello/

Key                 HelloWorldFunction
Description         Hello World Lambda Function ARN
Value
arn:aws:lambda:us-east-2:013109453517:function:getparam-app-001-HelloWorldF
unction-8GXBy46Zw0H3
------------------------------------------------------------------------------
------------------------------------------------------------------------------
-----------------------------------------------------

Successfully created/updated stack - getparam-app-001 in us-east-2

Hal:~/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://0l08ii78s1.execute-api.us-east-2.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://0l08ii78s1.execute-api.us-east-2.amazonaws.com/Prod/hello/?AAAA=BBB
B
```

We should see output similar to the following:

```
Hal:~/environment/getparam-app-001 $ curl
https://0l08ii78s1.execute-api.us-east-2.amazonaws.com/Prod/hello/?AAAA=BBB
B
{'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-ASN': '16509',
'CloudFront-Viewer-Country': 'US', 'Host':
'0l08ii78s1.execute-api.us-east-2.amazonaws.com', 'User-Agent':
'curl/7.58.0', 'Via': '2.0 b7f97186b1999ddac2896624abb211e4.cloudfront.net
(CloudFront)', 'X-Amz-Cf-Id':
'cO2jhzG9m9I7XWxzmnT6B7u8su8GfSs5auNRkO9lN1d17rRtWCPMgg==',
'X-Amzn-Trace-Id': 'Root=1-635ab7dd-3e74307846820d1200870d97',
'X-Forwarded-For': '3.144.40.202, 15.158.61.45', '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-ASN': ['16509'], 'CloudFront-Viewer-Country':
['US'], 'Host': ['0l08ii78s1.execute-api.us-east-2.amazonaws.com'],
'User-Agent': ['curl/7.58.0'], 'Via': ['2.0
b7f97186b1999ddac2896624abb211e4.cloudfront.net (CloudFront)'],
'X-Amz-Cf-Id':
['cO2jhzG9m9I7XWxzmnT6B7u8su8GfSs5auNRkO9lN1d17rRtWCPMgg=='],
'X-Amzn-Trace-Id': ['Root=1-635ab7dd-3e74307846820d1200870d97'],
'X-Forwarded-For': ['3.144.40.202, 15.158.61.45'], 'X-Forwarded-Port':
['443'], 'X-Forwarded-Proto': ['https']}, 'queryStringParameters': {'AAAA':
```

```
'BBBB'}, 'multiValueQueryStringParameters': {'AAAA': ['BBBB']},
'pathParameters': None, 'stageVariables': None, 'requestContext':
{'resourceId': 'an3frh', 'resourcePath': '/hello', 'httpMethod': 'GET',
'extendedRequestId': 'arGqnGBXiYcFifg=', 'requestTime':
'27/Oct/2022:16:54:53 +0000', 'path': '/Prod/hello/', 'accountId':
'013109453517', 'protocol': 'HTTP/1.1', 'stage': 'Prod', 'domainPrefix':
'0l08ii78s1', 'requestTimeEpoch': 1666889693357, 'requestId':
'7abf028f-8c73-4c41-abaa-91429bb413e3', 'identity':
{'cognitoIdentityPoolId': None, 'accountId': None, 'cognitoIdentityId':
None, 'caller': None, 'sourceIp': '3.144.40.202', 'principalOrgId': None,
'accessKey': None, 'cognitoAuthenticationType': None,
'cognitoAuthenticationProvider': None, 'userArn': None, 'userAgent':
'curl/7.58.0', 'user': None}, 'domainName':
'0l08ii78s1.execute-api.us-east-2.amazonaws.com', 'apiId': '0l08ii78s1'},
'body': None, 'isBase64Encoded': False}

Hal:~/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.

# References


- Tutorial: Deploying a Hello World application - https://docs.aws.amazon.com/serverless-application-model/latest/developerguide/serverless-getting-started-hello-world.html