SAM: Subdomain Discovery

Goals:

- Create and deploy a SAM app that will attempt to discover valid subdomains
    - 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"
- Understanding the content within the lab: "Local Debug & Testing"

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>

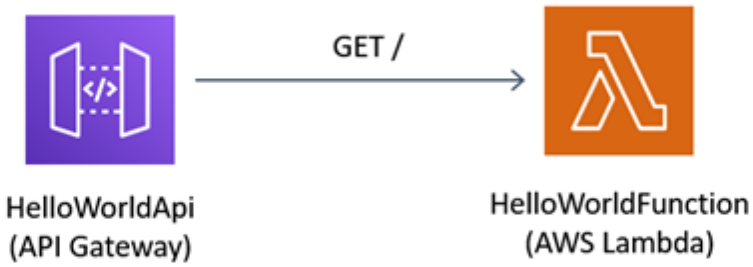
Login to the Cloud9 IDE Environment

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

subdomain-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]: subdomain-app-001

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

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


    Commands you can use next
    =========================
    [*] Create pipeline: cd subdomain-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 $
```

Code the App

Inspect the source code of the following files:

- template.yaml -> /home/ubuntu/environment/subdomain-app-001/template.yaml
    - SAM Template that defines your application's AWS resources

Change the "Path" to be the URI "subdomain" in the "template.yaml" file:
Inspect the source code of the following files:

```
...

Globals:

Function:

Timeout: 900

...

Properties:

CodeUri: subdomain/

Handler: app.lambda_handler

Runtime: python3.9

Timeout: 900

MemorySize: 512

Events:

HelloWorld:

Type: Api
```

```
Properties:

Path: /subdomain

Method: get

...
```

Click "File" -> "Save" or Ctrl+S on Windows, to save the "template.yaml" file

Next, move the "hello_world" directory to be called "subdomain":

```
pwd

cd /home/ubuntu/environment/subdomain-app-001/

ls -alF

mv hello_world/ subdomain/

ls -alF
```

We should see output similar to the following:

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

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

Hal:~/environment/subdomain-app-001 $ ls -alF
total 40
drwxrwxr-x 5 ubuntu ubuntu 4096 Oct 27 17:32 ./
drwxr-xr-x 8 ubuntu ubuntu 4096 Oct 27 17:32 ../
-rw-rw-r-- 1 ubuntu ubuntu 3730 Oct 27 17:32 .gitignore
-rw-rw-r-- 1 ubuntu ubuntu 8459 Oct 27 17:32 README.md
-rw-rw-r-- 1 ubuntu ubuntu    0 Oct 27 17:32 __init__.py
drwxrwxr-x 2 ubuntu ubuntu 4096 Oct 27 17:32 events/
drwxrwxr-x 2 ubuntu ubuntu 4096 Oct 27 17:32 hello_world/
-rw-rw-r-- 1 ubuntu ubuntu 1726 Oct 27 17:34 template.yaml
drwxrwxr-x 4 ubuntu ubuntu 4096 Oct 27 17:32 tests/
```

```
Hal:~/environment/subdomain-app-001 $ mv hello_world/ subdomain/

Hal:~/environment/subdomain-app-001 $ ls -alF
total 40
drwxrwxr-x 5 ubuntu ubuntu 4096 Oct 27 17:34 ./
drwxr-xr-x 8 ubuntu ubuntu 4096 Oct 27 17:32 ../
-rw-rw-r-- 1 ubuntu ubuntu 3730 Oct 27 17:32 .gitignore
-rw-rw-r-- 1 ubuntu ubuntu 8459 Oct 27 17:32 README.md
-rw-rw-r-- 1 ubuntu ubuntu    0 Oct 27 17:32 __init__.py
drwxrwxr-x 2 ubuntu ubuntu 4096 Oct 27 17:32 events/
drwxrwxr-x 2 ubuntu ubuntu 4096 Oct 27 17:32 subdomain/
-rw-rw-r-- 1 ubuntu ubuntu 1726 Oct 27 17:34 template.yaml
drwxrwxr-x 4 ubuntu ubuntu 4096 Oct 27 17:32 tests/

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

Inspect the source code of the following files:

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

Change the code so it looks like the following file:
https://github.com/TweekFawkes/train_intro_to_serverless/blob/main/code/subdomain-app-001/subdomain/app.py

Click "File" -> "Save" or Ctrl+S on Windows, to save the "app.py" file

Add the "namelist.txt" file into the same directory as the "app.py" file:
https://github.com/TweekFawkes/train_intro_to_serverless/blob/main/code/subdomain-app-001/subdomain/namelist.txt

Build and Deploy

Build and Deploy the app:

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

sam build

sam deploy --guided
```

```
subdomain-app-001

[ENTER]

y

Y

N

y

Y

[ENTER]

[ENTER]

y
```

We should see output similar to the following...

```
Hal:/ $ cd /home/ubuntu/environment/subdomain-app-001
Hal:~/environment/subdomain-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/subdomain-app-001/subdomain
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/subdomain-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]: subdomain-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/developergu
```

```
ide/serverless-sam-cli-config.html

Uploading to subdomain-app-001/2a57aee5eb05afd31196aa3d6f6eefa1  467293 /
467293  (100.00%)

        Deploying with following values
        ===============================
        Stack name                   : subdomain-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 subdomain-app-001/9432178a933a97b2a7346ff48c497a39.template
1257 / 1257  (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
ServerlessRestApiDeployment3caa84f1bd
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-deploy166689
2320/a045ab06-6761-4e98-9c70-038fdd33d1e2


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

2022-10-27 17:38:49 - 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::ApiGateway::Deployment
ServerlessRestApiDeployment3caa84f1bd               -
CREATE_IN_PROGRESS                                  AWS::Lambda::Permission
HelloWorldFunctionHelloWorldPermissionProd          -
CREATE_IN_PROGRESS                                  AWS::Lambda::Permission
HelloWorldFunctionHelloWorldPermissionProd          Resource creation
Initiated
CREATE_IN_PROGRESS
AWS::ApiGateway::Deployment
ServerlessRestApiDeployment3caa84f1bd               Resource creation
Initiated
CREATE_COMPLETE
AWS::ApiGateway::Deployment
ServerlessRestApiDeployment3caa84f1bd               -
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                          subdomain-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/subdomain-app-001-HelloWorldFunctionRole-PH0
7K3SOSPZS

Key                 HelloWorldApi
Description         API Gateway endpoint URL for Prod stage for Hello World
function
Value
https://d54kogutvb.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:subdomain-app-001-HelloWorld
Function-ITYJmwZv3Txv
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
-----------------------------------------------------

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

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

Test Deployment

Let's test our deployment (change the URL to the URL created for your deployment):

```
curl
https://d54kogutvb.execute-api.us-east-2.amazonaws.com/Prod/subdomain/?Root
DomainName=lizardblue.com
```

We should see output similar to the following:

```
Hal:~/environment/subdomain-app-001 $ curl
https://d54kogutvb.execute-api.us-east-2.amazonaws.com/Prod/subdomain/?Root
DomainName=lizardblue.com
[+] START
[+] Subdomain discovered: cdn.lizardblue.com -> 54.231.140.161
[+] Subdomain discovered: cdn2.lizardblue.com -> 52.219.177.84
[+] Subdomain discovered: hash.lizardblue.com -> 108.156.172.118,
108.156.172.51, 108.156.172.75, 108.156.172.89
[+] Subdomain discovered: images.lizardblue.com -> 52.219.80.116
[+] Subdomain discovered: ixhash.lizardblue.com -> 23.21.224.182,
3.95.97.203, 34.200.79.90, 34.206.167.232, 34.237.26.77, 52.3.70.65,
52.7.59.127, 54.85.138.216
[+] END

Hal:~/environment/subdomain-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://stackoverflow.com/questions/6817640/catch-any-error-in-python
- https://stackoverflow.com/questions/19196105/how-to-check-if-a-network-port-is-open