Open Port Check

Goals:

- Create and deploy a SAM app that will attempt to discover valid subdomains

Dependencies:

- 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/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

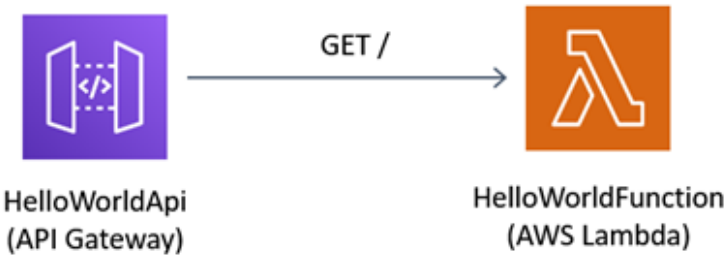
Region: US East (N. Virginia) us-east-1

Service: Cloud9

Locate the "HelloWorld101" Cloud9 environment

Click the "Open IDE" button

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



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

```
cd /home/ubuntu/environment

sam init

1

1

11

portcheck-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]: portcheck-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: portcheck-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:

- template.yaml -> /home/ubuntu/environment/portcheck-app-001/template.yaml

-- SAM Template that defines your application's AWS resources

Change the "CodeUri" and "Path" to be the following values in the "template.yaml" file:

```
...
Globals:
Function:
Timeout: 900
...
Properties:
CodeUri: port_check/
Handler: app.lambda_handler
Runtime: python3.6
Timeout: 900
```

```
MemorySize: 512

...

Properties:

Path: /portcheck

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 "port_check":

```
pwd

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

ls -alF

mv hello_world/ port_check/

ls -alF
```

We should see output similar to the following:

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


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


red_team_040:~/environment/portcheck-app-001 $ ls -alF
total 40
drwxrwxr-x 5 ubuntu ubuntu 4096 Sep 23 21:45 ./
drwxr-xr-x 6 ubuntu ubuntu 4096 Sep 23 21:45 ../
-rw-rw-r-- 1 ubuntu ubuntu 3730 Sep 23 21:45 .gitignore
-rw-rw-r-- 1 ubuntu ubuntu 8240 Sep 23 21:45 README.md
-rw-rw-r-- 1 ubuntu ubuntu 0 Sep 23 21:45 __init__.py
drwxrwxr-x 2 ubuntu ubuntu 4096 Sep 23 21:45 events/
drwxrwxr-x 2 ubuntu ubuntu 4096 Sep 23 21:45 hello_world/
```

```
-rw-rw-r-- 1 ubuntu ubuntu 1643 Sep 23 21:45 template.yaml
drwxrwxr-x 3 ubuntu ubuntu 4096 Sep 23 21:45 tests/


red_team_040:~/environment/portcheck-app-001 $ mv hello_world/ port_check/


red_team_040:~/environment/portcheck-app-001 $ ls -alF
total 40
drwxrwxr-x 5 ubuntu ubuntu 4096 Sep 23 23:44 ./
drwxr-xr-x 9 ubuntu ubuntu 4096 Sep 23 23:43 ../
-rw-rw-r-- 1 ubuntu ubuntu 3730 Sep 23 23:43 .gitignore
-rw-rw-r-- 1 ubuntu ubuntu 8240 Sep 23 23:43 README.md
-rw-rw-r-- 1 ubuntu ubuntu 0 Sep 23 23:43 __init__.py
drwxrwxr-x 2 ubuntu ubuntu 4096 Sep 23 23:43 events/
drwxrwxr-x 2 ubuntu ubuntu 4096 Sep 23 23:43 port_check/
-rw-rw-r-- 1 ubuntu ubuntu 1643 Sep 23 23:43 template.yaml
drwxrwxr-x 3 ubuntu ubuntu 4096 Sep 23 23:43 tests/


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

Inspect the source code of the following files:

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

-- Contains the logic/code for your lambda application

When passing the lambda function information via the API gateway as HTTP GET Parameters, e.g.

```
red_team_040:~/environment/dirb-app-010 $ curl https://EXAMPLE.execute-api.us-east-1.amazonaws.com/Prod/dirb/?AAAA=BBBB
```

The "event" object will contain data similar to the following...

```
{'resource': '/dirb', 'path': '/dirb/', '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': '7ierqt1j17.execute-api.us-east-1.amazonaws.com', 'User-Agent': 'curl/7.58.0', 'Via': '2.0 237bd7e86f7f99cead16dc4ecb5fed20.cloudfront.net (CloudFront)', 'X-Amz-Cf-Id': '2_HaEWlB9X5fOnGYWnQJVj09JvA9ztuSZ7h9fGCLECepvTzOoZBRJw==', 'X-Amzn-Trace-Id': 'Root=1-614a56b1-0bf806fa6a43613863d243b4', 'X-Forwarded-For': '3.226.252.96, 70.132.60.74', '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': ['7ierqt1j17.execute-api.us-east-1.amazonaws.com'], 'User-Agent': ['curl/7.58.0'], 'Via': ['2.0 237bd7e86f7f99cead16dc4ecb5fed20.cloudfront.net (CloudFront)'], 'X-Amz-Cf-Id': ['2_HaEWlB9X5fOnGYWnQJVj09JvA9ztuSZ7h9fGCLECepvTzOoZBRJw=='], 'X-Amzn-Trace-Id': ['Root=1-614a56b1-0bf806fa6a43613863d243b4'], 'X-Forwarded-For': ['3.226.252.96, 70.132.60.74'], 'X-Forwarded-Port': ['443'], 'X-Forwarded-Proto': ['https']}, 'queryStringParameters': {'AAAA': 'BBBB'}, 'multiValueQueryStringParameters': {'AAAA': ['BBBB']}, 'pathParameters': None, 'stageVariables': None, 'requestContext': {'resourceId': '80a50y', 'resourcePath': '/dirb', 'httpMethod': 'GET', 'extendedRequestId': 'GCJ7tETQIAMF4jg=', 'requestTime': '21/Sep/2021:22:03:29 +0000', 'path': '/Prod/dirb/', 'accountId': '580299357056', 'protocol': 'HTTP/1.1', 'stage': 'Prod', 'domainPrefix': '7ierqt1j17', 'requestTimeEpoch': 1632261809163, 'requestId': 'f8e77801-303a-4aa3-b32c-2149491d6f66', 'identity': {'cognitoIdentityPoolId': None, 'accountId': None, 'cognitoIdentityId': None, 'caller': None, 'sourceIp': '3.226.252.96', 'principalOrgId': None, 'accessKey': None, 'cognitoAuthenticationType': None, 'cognitoAuthenticationProvider': None, 'userArn': None, 'userAgent': 'curl/7.58.0', 'user': None}, 'domainName': '7ierqt1j17.execute-api.us-east-1.amazonaws.com', 'apiId': '7ierqt1j17'}, 'body': None, 'isBase64Encoded': False}

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

```
event['queryStringParameters']

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

Add the following imports to the top of the "app.py" file:

```
import json

import socket
```

Add the following logic to the application's lambda_handler() function to process the input via a GET query parameter called "RootDomainName":

```
sReturn = "NULL"

try:
```

```
sTargetIp = str(event['queryStringParameters']['TargetIp'])
sTcpPort = str(event['queryStringParameters']['TcpPort'])

print("[~] sTargetIp: " + sTargetIp)
print("[~] sTcpPort: " + sTcpPort)

sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

sock.settimeout(2) #2 Second Timeout

result = sock.connect_ex((sTargetIp,int(sTcpPort)))

if result == 0:
print("Port is open")
sReturn = sTargetIp + ":" + sTcpPort + "/TCP is open"
else:
print("Port is not open")
sReturn = sTargetIp + ":" + sTcpPort + "/TCP is closed"

sock.close()
except Exception as e:
print("[!] Exception (e):" + str(e))

return {
"statusCode": 200,
"body": sReturn,
}
```

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

Run the following commands to build and deploy the application...

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

sam build

sam deploy --guided

portcheck-app-001



sam deploy --guided
```

We should see output similar to the following...

```
red_team_040:~/environment/portcheck-app-001 $ cd /home/ubuntu/environment/portcheck-app-001
red_team_040:~/environment/portcheck-app-001 $ sam build
Building codeuri: /home/ubuntu/environment/portcheck-app-001/port_check 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/portcheck-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]: portcheck-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 portcheck-app-001/f64902daae7dc1cb743d3a38653305ba 444965 / 444965 (100.00%)

Deploying with following values
==============================
Stack name : portcheck-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 portcheck-app-001/d06e57efee98bbadcb8ac2b44746a9c9.template 1124 / 1124
(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 ServerlessRestApiDeployment9a29b48186 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-deploy1632441454/60061bfa-7466-4a9b-99f3-85b78e851316
```

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

2021-09-23 23:57:49 - 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 -
CREATE_IN_PROGRESS AWS::ApiGateway::Deployment ServerlessRestApiDeployment9a29b48186
-
CREATE_IN_PROGRESS AWS::ApiGateway::Deployment ServerlessRestApiDeployment9a29b48186
Resource creation Initiated
CREATE_IN_PROGRESS AWS::Lambda::Permission HelloWorldFunctionHelloWorldPermissionProd
Resource creation Initiated
CREATE_COMPLETE AWS::ApiGateway::Deployment ServerlessRestApiDeployment9a29b48186 -
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 portcheck-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/portcheck-app-001-HelloWorldFunctionRole-1UUR3CT5S3QPY

Key HelloWorldApi
Description API Gateway endpoint URL for Prod stage for Hello World function
Value https://5iy3e3kfek.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:portcheck-app-001-HelloWorldFunction-
hJU8qn6BI7rC
---------------------------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------

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

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

Test the deployment via the following command (replacing the URL with the URL from your deployment):

```
curl "https://5iy3e3kfek.execute-api.us-east-1.amazonaws.com/Prod/portcheck/?
TargetIp=13.82.46.24&TcpPort=22"

curl "https://5iy3e3kfek.execute-api.us-east-1.amazonaws.com/Prod/portcheck/?
TargetIp=13.82.46.24&TcpPort=23"

curl "https://5iy3e3kfek.execute-api.us-east-1.amazonaws.com/Prod/portcheck/?
TargetIp=13.82.46.24&TcpPort=80"

curl "https://5iy3e3kfek.execute-api.us-east-1.amazonaws.com/Prod/portcheck/?
TargetIp=13.82.46.24&TcpPort=443"

curl "https://5iy3e3kfek.execute-api.us-east-1.amazonaws.com/Prod/portcheck/?
TargetIp=13.82.46.24&TcpPort=10000"
```

We should see output similar to the following...

```
red_team_040:~/environment/portcheck-app-001 $ curl "https://5iy3e3kfek.execute-api.us-east-
1.amazonaws.com/Prod/portcheck/?TargetIp=13.82.46.24&TcpPort=22"

13.82.46.24:22/TCP is open
```

```
red_team_040:~/environment/portcheck-app-001 $ curl "https://5iy3e3kfek.execute-api.us-east-1.amazonaws.com/Prod/portcheck/?TargetIp=13.82.46.24&TcpPort=23"
13.82.46.24:23/TCP is closed

red_team_040:~/environment/portcheck-app-001 $ curl "https://5iy3e3kfek.execute-api.us-east-1.amazonaws.com/Prod/portcheck/?TargetIp=13.82.46.24&TcpPort=80"
13.82.46.24:80/TCP is open

red_team_040:~/environment/portcheck-app-001 $ curl "https://5iy3e3kfek.execute-api.us-east-1.amazonaws.com/Prod/portcheck/?TargetIp=13.82.46.24&TcpPort=443"
13.82.46.24:443/TCP is closed

red_team_040:~/environment/portcheck-app-001 $ curl "https://5iy3e3kfek.execute-api.us-east-1.amazonaws.com/Prod/portcheck/?TargetIp=13.82.46.24&TcpPort=10000"
13.82.46.24:10000/TCP is closed

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

Clean Up

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. sam-app-001

#2 - AWS Region: e.g. us-east-1

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

```
aws cloudformation delete-stack --stack-name sam-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 sam-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": "sam-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)**

https://stackoverflow.com/questions/6817640/catch-any-error-in-python (https://stackoverflow.com/questions/6817640/catch-any-error-in-python)

https://stackoverflow.com/questions/19196105/how-to-check-if-a-network-port-is-open (https://stackoverflow.com/questions/19196105/how-to-check-if-a-network-port-is-open)