SAM: Cloud9 & SAM 101

Goals:

- Create your first SAM app using the Cloud9 IDE
    - 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

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>

Create the Cloud9 IDE Environment

Region: Ohio / us-east-2
Service: Cloud9

Click the "Create environment" button

Name: HelloWorld101
Click the "Next step" button

Environment type: Select the "Create a new EC2 instance for environment (direct access)" radio button
Instance type: Select the "t3.small" radio button
Platform: Select the "Ubuntu Server 18.04 LTS" radio button
Subnet: Select the "traininglabSubnet_public" option
Click the "Next step" button

Click the "Create environment" button
This will take a few minutes to create, on average around 5 minutes.

Install Python 3.9

By default the Ubuntu Server 18.04 LTS image used by Cloud9 uses python3.6.9 which is currently no longer supported as a runtime by lambda, so we need to upgrade our python version, to get Cloud9 and Lambda working together smoothly.

Commands to Install Python 3.9:

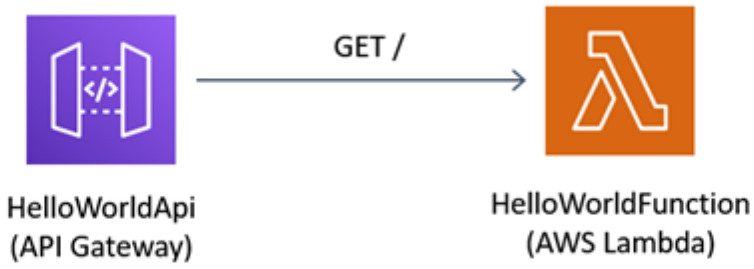
```
sudo add-apt-repository ppa:deadsnakes/ppa
sudo apt update
sudo apt install python3.9
```

Commands to Install pip for python3.9:

```
sudo apt install python3.9-distutils
curl https://bootstrap.pypa.io/get-pip.py -o get-pip.py
python3.9 get-pip.py
```

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

sam-app-001
```

We should see output similar to the following:

```
Hal:/ $ cd ~/environment/

Hal:~/environment $ sam init

        SAM CLI now collects telemetry to better understand customer needs.

        You can OPT OUT and disable telemetry collection by setting the
        environment variable SAM_CLI_TELEMETRY=0 in your shell.
        Thanks for your help!

        Learn More:
https://docs.aws.amazon.com/serverless-application-model/latest/developergu
ide/serverless-sam-telemetry.html


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]: sam-app-001

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

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


    Commands you can use next
    =========================
    [*] Create pipeline: cd sam-app-001 && sam pipeline init --bootstrap
    [*] Validate SAM template: sam validate
    [*] Test Function in the Cloud: sam sync --stack-name {stack-name}
--watch


SAM CLI update available (1.60.0); (1.57.0 installed)
To download:
https://docs.aws.amazon.com/serverless-application-model/latest/developergu
ide/serverless-sam-cli-install.html

Hal:~/environment $
```

Inspect the source code of the following files

- template.yaml -> /home/ubuntu/environment/sam-app-002/template.yaml
    - SAM Template that defines your application's AWS resources

- app.py -> /home/ubuntu/environment/sam-app-002/hello_world/app.py
    - Contains the logic/code for your lambda application

- requirements.txt -> /home/ubuntu/environment/sam-app-002/hello_world/app.py
    - Required python imports/dependencies

Check Template.yaml file:

```
...

Runtime: python3.9

...
```

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 sam-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 2532
drwxr-xr-x  4 ubuntu ubuntu    4096 Oct 25 15:19 ./
drwxr-xr-x 15 ubuntu ubuntu    4096 Oct 25 15:19 ../
drwxrwxr-x  5 ubuntu ubuntu    4096 Oct 25 15:21 .c9/
-rw-r--r--  1 ubuntu ubuntu     569 Oct 18 17:59 README.md
-rw-rw-r--  1 ubuntu ubuntu 2569494 Oct 25 15:16 get-pip.py
drwxrwxr-x  5 ubuntu ubuntu    4096 Oct 25 15:19 sam-app-001/

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

Hal:~/environment/sam-app-001 $ ls -alF
total 40
drwxrwxr-x 5 ubuntu ubuntu 4096 Oct 25 15:19 ./
drwxr-xr-x 4 ubuntu ubuntu 4096 Oct 25 15:19 ../
-rw-rw-r-- 1 ubuntu ubuntu 3730 Oct 25 15:19 .gitignore
-rw-rw-r-- 1 ubuntu ubuntu 8393 Oct 25 15:19 README.md
-rw-rw-r-- 1 ubuntu ubuntu    0 Oct 25 15:19 __init__.py
drwxrwxr-x 2 ubuntu ubuntu 4096 Oct 25 15:19 events/
drwxrwxr-x 2 ubuntu ubuntu 4096 Oct 25 15:19 hello_world/
-rw-rw-r-- 1 ubuntu ubuntu 1669 Oct 25 15:19 template.yaml
drwxrwxr-x 4 ubuntu ubuntu 4096 Oct 25 15:19 tests/

Hal:~/environment/sam-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/sam-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/sam-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.

```
Hal:~/environment/sam-app-001 $ ls -alF .aws-sam/
total 16
drwxrwxr-x 3 ubuntu ubuntu 4096 Oct 25 15:22 ./
drwxrwxr-x 6 ubuntu ubuntu 4096 Oct 25 15:22 ../
drwxr-xr-x 3 ubuntu ubuntu 4096 Oct 25 15:22 build/
-rw-rw-r-- 1 ubuntu ubuntu  390 Oct 25 15:22 build.toml

Hal:~/environment/sam-app-001 $ ls -alF .aws-sam/build/
total 16
drwxr-xr-x  3 ubuntu ubuntu 4096 Oct 25 15:22 ./
drwxrwxr-x  3 ubuntu ubuntu 4096 Oct 25 15:22 ../
drwxrwxr-x 12 ubuntu ubuntu 4096 Oct 25 15:22 HelloWorldFunction/
-rw-rw-r--  1 ubuntu ubuntu 1100 Oct 25 15:22 template.yaml

Hal:~/environment/sam-app-001 $ ls -alF .aws-sam/build/HelloWorldFunction/
total 56
drwxrwxr-x 12 ubuntu ubuntu 4096 Oct 25 15:22 ./
drwxr-xr-x  3 ubuntu ubuntu 4096 Oct 25 15:22 ../
-rw-rw-r--  1 ubuntu ubuntu    0 Oct 25 15:19 __init__.py
-rw-rw-r--  1 ubuntu ubuntu 1151 Oct 25 15:19 app.py
drwxrwxr-x  2 ubuntu ubuntu 4096 Oct 25 15:22 certifi/
drwxrwxr-x  2 ubuntu ubuntu 4096 Oct 25 15:22 certifi-2022.9.24.dist-info/
drwxrwxr-x  4 ubuntu ubuntu 4096 Oct 25 15:22 charset_normalizer/
drwxrwxr-x  2 ubuntu ubuntu 4096 Oct 25 15:22
charset_normalizer-2.1.1.dist-info/
drwxrwxr-x  2 ubuntu ubuntu 4096 Oct 25 15:22 idna/
drwxrwxr-x  2 ubuntu ubuntu 4096 Oct 25 15:22 idna-3.4.dist-info/
drwxrwxr-x  2 ubuntu ubuntu 4096 Oct 25 15:22 requests/
drwxrwxr-x  2 ubuntu ubuntu 4096 Oct 25 15:22 requests-2.28.1.dist-info/
-rw-rw-r--  1 ubuntu ubuntu    8 Oct 25 15:19 requirements.txt
drwxrwxr-x  5 ubuntu ubuntu 4096 Oct 25 15:22 urllib3/
drwxrwxr-x  2 ubuntu ubuntu 4096 Oct 25 15:22 urllib3-1.26.12.dist-info/

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

Deploy the SAM App

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

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

```
sam deploy --guided

sam-app-001

[ENTER]

y

Y

N

y

Y

[ENTER]

[ENTER]

y
```

We should see output similar to the following:

```
Hal:~/environment/sam-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]: sam-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:
        Creating the required resources...
        Successfully created!
         Managed S3 bucket:
aws-sam-cli-managed-default-samclisourcebucket-d3bpvxlmq92
         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 sam-app-001/1007ba6f2fd6aed6daf87b6f392f298d  466367 / 466367
(100.00%)

        Deploying with following values
        ===============================
        Stack name                   : sam-app-001
        Region                       : us-east-2
        Confirm changeset            : True
        Disable rollback             : False
        Deployment s3 bucket         :
aws-sam-cli-managed-default-samclisourcebucket-d3bpvxlmq92
        Capabilities                 : ["CAPABILITY_IAM"]
```

```
        Parameter overrides          : {}
        Signing Profiles             : {}

Initiating deployment
=====================
Uploading to sam-app-001/c96f68fa2a8e321966991551a85f0769.template  1190 /
1190  (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-2:047851407302:changeSet/samcli-deploy166671
1573/febb0212-1eba-4359-96b7-cef4de9cf24e


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

2022-10-25 15:28:03 - 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                              sam-app-001
```

```
-
------------------------------------------------------------------------------
------------------------------------------------------------------------------
------------------------------------------------------------------------------
------------
CloudFormation outputs from deployed stack
------------------------------------------------------------------------------
------------------------------------------------------------------------------
------------------------------------------------------------------------------
------------
Outputs
------------------------------------------------------------------------------
------------------------------------------------------------------------------
------------------------------------------------------------------------------
------------
Key                 HelloWorldFunctionIamRole
Description         Implicit IAM Role created for Hello World function
Value
arn:aws:iam::047851407302:role/sam-app-001-HelloWorldFunctionRole-917TRXZMF
CXE

Key                 HelloWorldApi
Description         API Gateway endpoint URL for Prod stage for Hello World
function
Value
https://p9jr43jgs8.execute-api.us-east-2.amazonaws.com/Prod/hello/

Key                 HelloWorldFunction
Description         Hello World Lambda Function ARN
Value
arn:aws:lambda:us-east-2:047851407302:function:sam-app-001-HelloWorldFuncti
on-EK2zw39PbTgh
------------------------------------------------------------------------------
------------------------------------------------------------------------------
------------------------------------------------------------------------------
------------

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

Hal:~/environment/sam-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://p9jr43jgs8.execute-api.us-east-2.amazonaws.com/Prod/hello/
```

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

```
curl https://p9jr43jgs8.execute-api.us-east-2.amazonaws.com/Prod/hello/
```

We should see output similar to the following:

```
Hal:~/environment/sam-app-001 $ curl
https://p9jr43jgs8.execute-api.us-east-2.amazonaws.com/Prod/hello/

{"message": "hello world"}

Hal:~/environment/sam-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