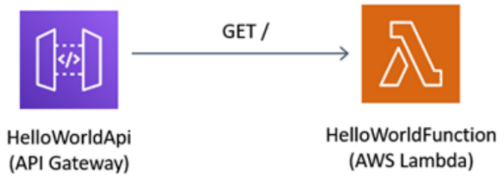
Assignment – 8

1. **Explain the below aws architecture diagram in detail, also deploy the same aws architecture.**

- For this assignment you need to take a look and study the documentation for SAM CLI, you need to deploy a Hello, World application on aws lambda.
- Make sure when you test the lambda url it will respond as Hello, World.9



In this diagram, you download, build, and deploy a sample Hello World application using AWS SAM. You then test the application in the AWS Cloud, and optionally test it locally on your development host.

This application implements a basic API backend. It consists of an Amazon API Gateway endpoint and an AWS Lambda function. When you send a GET request to the API Gateway endpoint, the Lambda function is invoked. This function returns a hello world message.

Step 1: Download the sample Hello World application using AWS SAM.

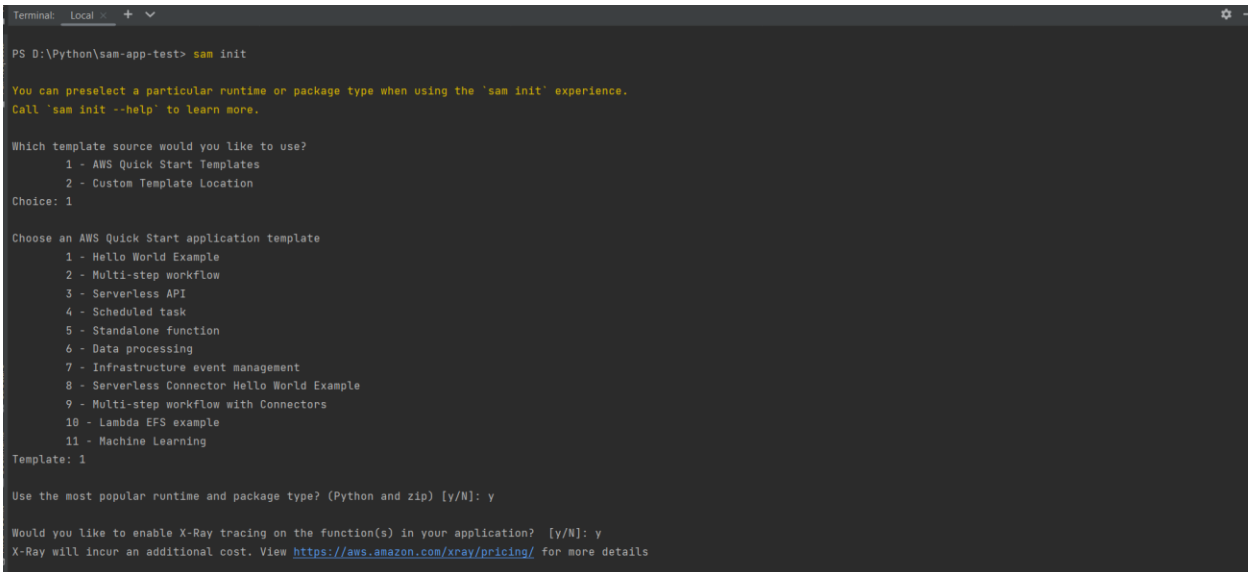
Step 2: Build the application using the necessary dependencies and configurations.

Step 3: Deploy the application to the AWS Cloud using AWS SAM.

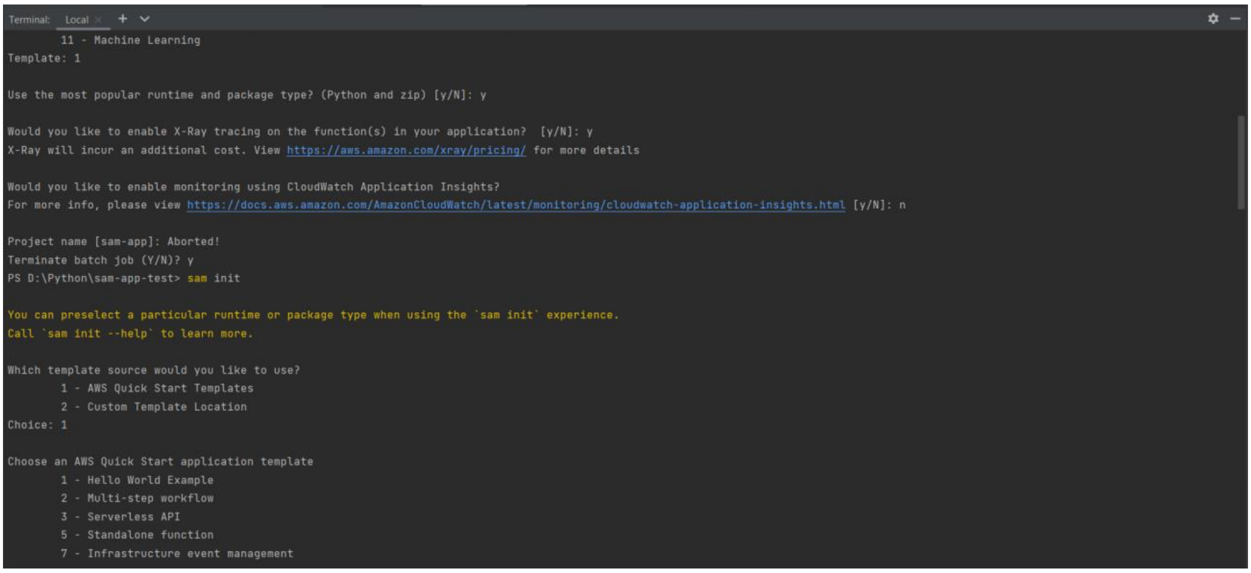
Step 4: Test the application by sending a GET request to the Amazon API Gateway endpoint.

Step 5: Observe that the Lambda function is invoked and returns a "hello world" message.

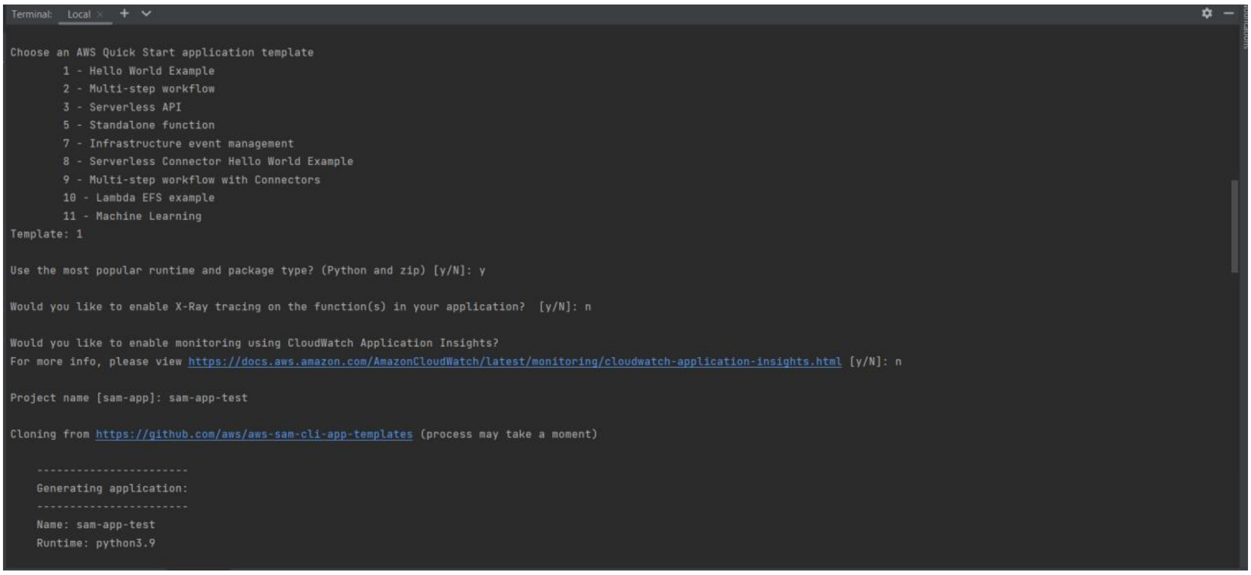
Step 6: (Optional) Test the application locally on your development host to ensure it is working as expected.



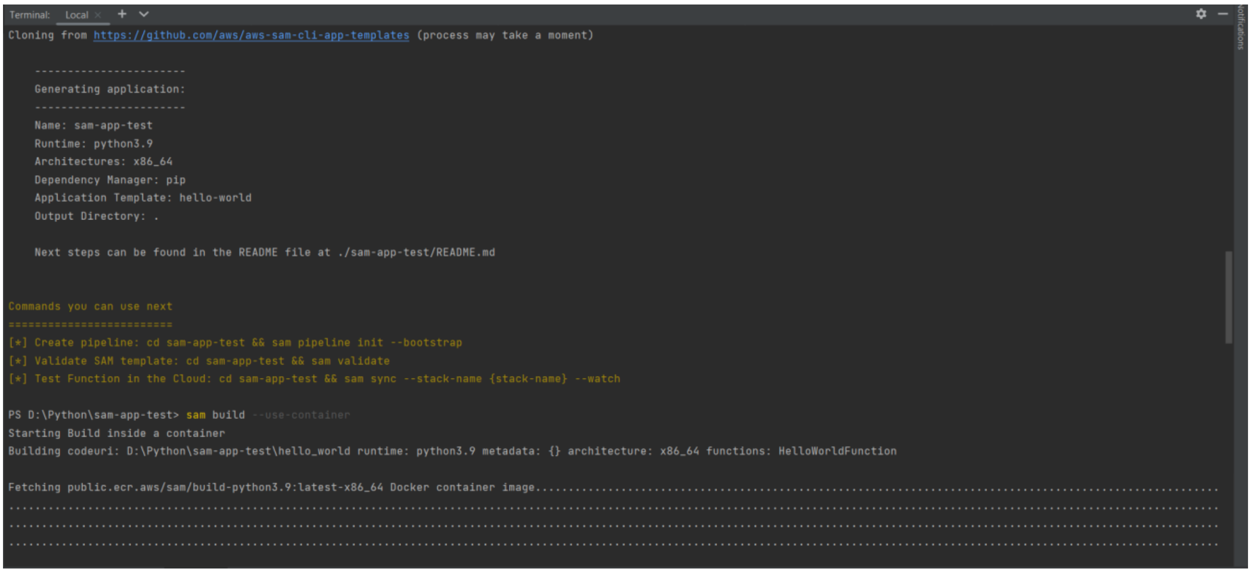
```
PS D:\Python\sam-app-test> sam init

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 - Serverless Connector Hello World Example
        9 - Multi-step workflow with Connectors
        10 - Lambda EFS example
        11 - Machine Learning
Template: 1

Use the most popular runtime and package type? (Python and zip) [y/N]: y

Would you like to enable X-Ray tracing on the function(s) in your application?  [y/N]: y
X-Ray will incur an additional cost. View https://aws.amazon.com/xray/pricing/ for more details
```



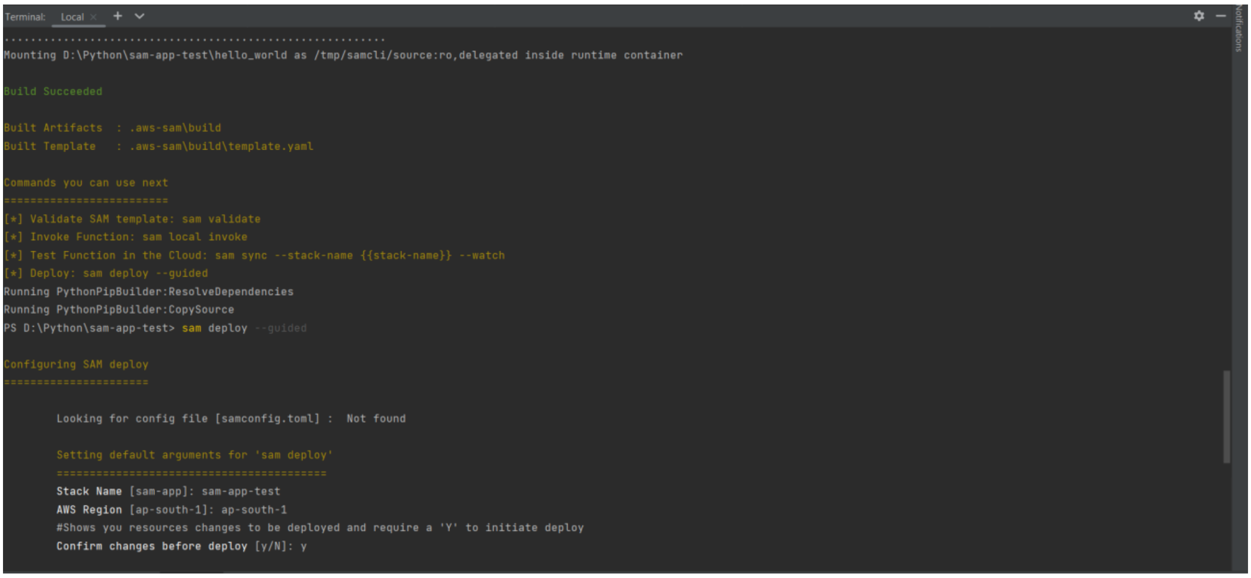
```
        11 - Machine Learning
Template: 1

Use the most popular runtime and package type? (Python and zip) [y/N]: y

Would you like to enable X-Ray tracing on the function(s) in your application?  [y/N]: y
X-Ray will incur an additional cost. View https://aws.amazon.com/xray/pricing/ for more details

Would you like to enable monitoring using CloudWatch Application Insights?
For more info, please view https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/cloudwatch-application-insights.html [y/N]: n

Project name [sam-app]: Aborted!
Terminate batch job (Y/N)? y
PS D:\Python\sam-app-test> 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
        5 - Standalone function
        7 - Infrastructure event management
```



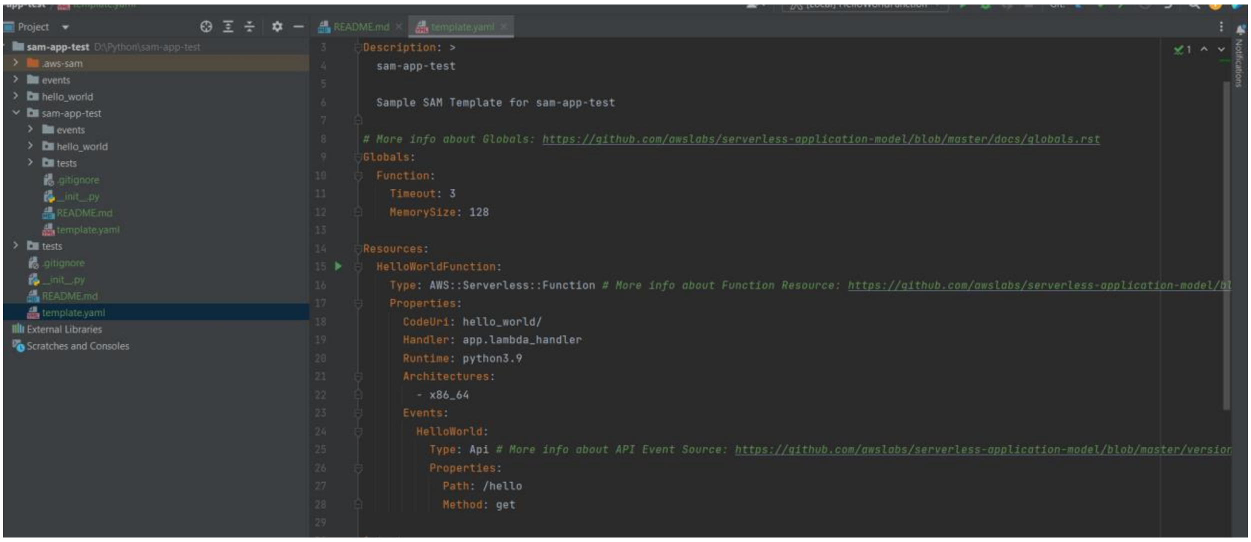
```
Choose an AWS Quick Start application template
        1 - Hello World Example
        2 - Multi-step workflow
        3 - Serverless API
        5 - Standalone function
        7 - Infrastructure event management
        8 - Serverless Connector Hello World Example
        9 - Multi-step workflow with Connectors
        10 - Lambda EFS example
        11 - Machine Learning
Template: 1

Use the most popular runtime and package type? (Python and zip) [y/N]: y

Would you like to enable X-Ray tracing on the function(s) in your application?  [y/N]: n

Would you like to enable monitoring using CloudWatch Application Insights?
For more info, please view https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/cloudwatch-application-insights.html [y/N]: n

Project name [sam-app]: sam-app-test

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

    -----------------------
    Generating application:
    -----------------------
    Name: sam-app-test
    Runtime: python3.9
```



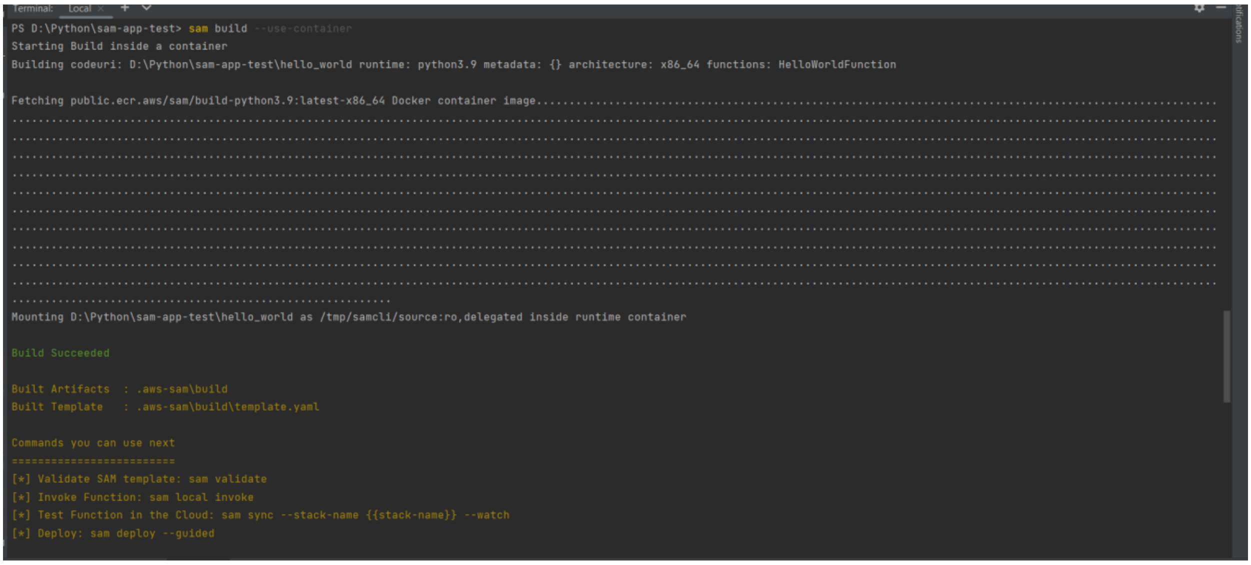
```
Cloning from https://github.com/aws/aws-sam-cli-app-templates (process may take a moment)

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


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

PS D:\Python\sam-app-test> sam build --use-container
Starting Build inside a container
Building codeuri: D:\Python\sam-app-test\hello_world runtime: python3.9 metadata: {} architecture: x86_64 functions: HelloWorldFunction

Fetching public.ecr.aws/sam/build-python3.9:latest-x86_64 Docker container image......................................................................
..........................................................................................................................................................................
..........................................................................................................................................................................
..........................................................................................................................................................................
```



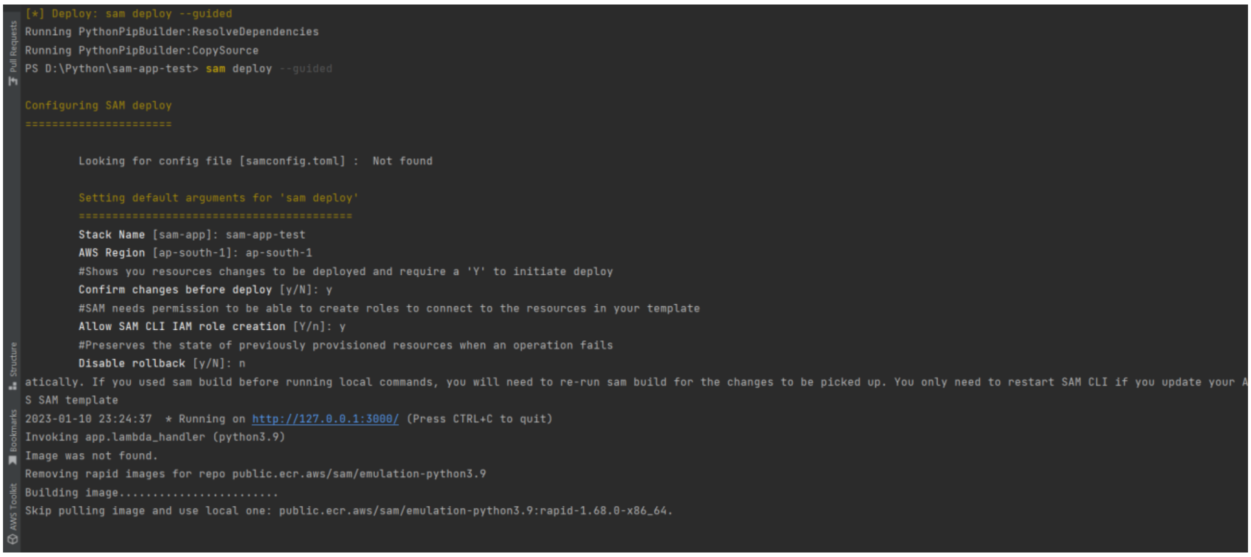
```
..........................................................
Mounting D:\Python\sam-app-test\hello_world as /tmp/samcli/source:ro,delegated inside runtime container

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
Running PythonPipBuilder:ResolveDependencies
Running PythonPipBuilder:CopySource
PS D:\Python\sam-app-test> 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-test
        AWS Region [ap-south-1]: ap-south-1
        #Shows you resources changes to be deployed and require a 'Y' to initiate deploy
        Confirm changes before deploy [y/N]: y
```



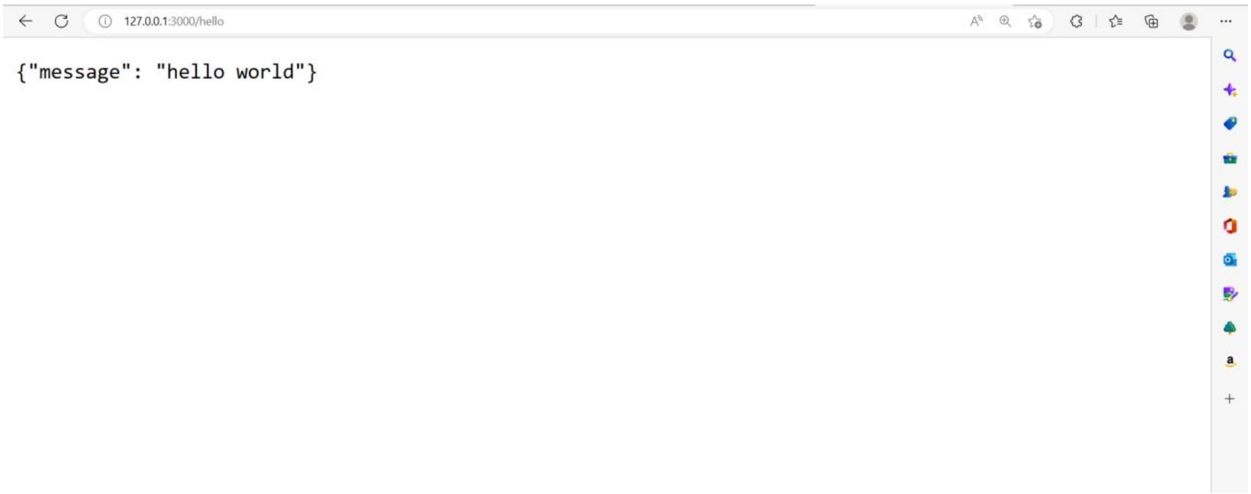
```
Description: >
  sam-app-test

  Sample SAM Template for sam-app-test

# More info about Globals: https://github.com/awslabs/serverless-application-model/blob/master/docs/globals.rst
Globals:
  Function:
    Timeout: 3
    MemorySize: 128

Resources:
  HelloWorldFunction:
    Type: AWS::Serverless::Function # More info about Function Resource: https://github.com/awslabs/serverless-application-model/bl
    Properties:
      CodeUri: hello_world/
      Handler: app.lambda_handler
      Runtime: python3.9
      Architectures:
        - x86_64
      Events:
        HelloWorld:
          Type: Api # More info about API Event Source: https://github.com/awslabs/serverless-application-model/blob/master/version

```



```
PS D:\Python\sam-app-test> sam build --use-container
Starting Build inside a container
Building codeuri: D:\Python\sam-app-test\hello_world runtime: python3.9 metadata: {} architecture: x86_64 functions: HelloWorldFunction

Fetching public.ecr.aws/sam/build-python3.9:latest-x86_64 Docker container image.............................................................................................
..............................................................................................................................................................................
..............................................................................................................................................................................
..............................................................................................................................................................................
..............................................................................................................................................................................
..............................................................................................................................................................................
..............................................................................................................................................................................
..............................................................................................................................................................................
..............................................................................................................................................................................
..............................................................................................................................................................................
..............................................................................................................................................................................
.......................................................
Mounting D:\Python\sam-app-test\hello_world as /tmp/samcli/source:ro,delegated inside runtime container

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
```



```
[*] Deploy: sam deploy --guided
Running PythonPipBuilder:ResolveDependencies
Running PythonPipBuilder:CopySource
PS D:\Python\sam-app-test> 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-test
        AWS Region [ap-south-1]: ap-south-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
        #Preserves the state of previously provisioned resources when an operation fails
        Disable rollback [y/N]: n
atically. If you used sam build before running local commands, you will need to re-run sam build for the changes to be picked up. You only need to restart SAM CLI if you update your A
S SAM template
2023-01-10 23:24:37  * Running on http://127.0.0.1:3000/ (Press CTRL+C to quit)
Invoking app.lambda_handler (python3.9)
Image was not found.
Removing rapid images for repo public.ecr.aws/sam/emulation-python3.9
Building image........................
Skip pulling image and use local one: public.ecr.aws/sam/emulation-python3.9:rapid-1.68.0-x86_64.
```



```
{"message": "hello world"}
```