DevOps -2022-b-online

BY

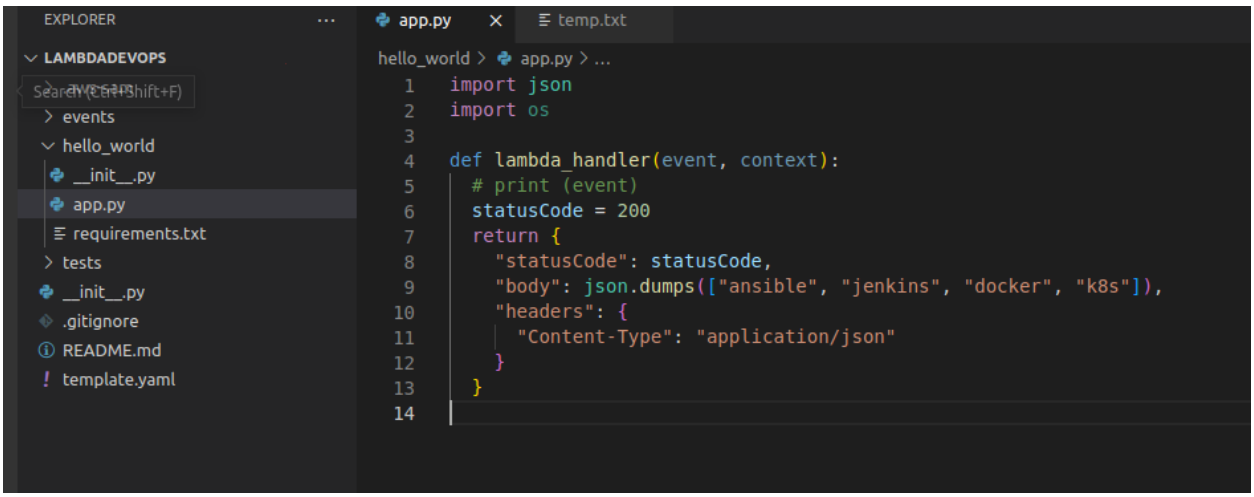
MOHAMMED EID 😊

- Name: LambdaDevOps
- Runtime: Python 3.9
- Create the SAM application in the local system.

Take the Lambda DevOps topics code and test it locally using AWS SAM.

```
devops@devops-VirtualBox:~/Desktop/sam-aws$ 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 - 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 - python3.6
        17 - ruby2.7
        18 - 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]: LambdaDevOps

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

    -----------------------
    Generating application:
    -----------------------
    Name: LambdaDevOps
    Runtime: python3.9
    Architectures: x86_64
    Dependency Manager: pip
    Application Template: hello-world
    Output Directory: .
```

```
devops@devops-VirtualBox:~/Desktop/sam-aws$ ls
LambdaDevOps
devops@devops-VirtualBox:~/Desktop/sam-aws$ code LambdaDevOps/
```

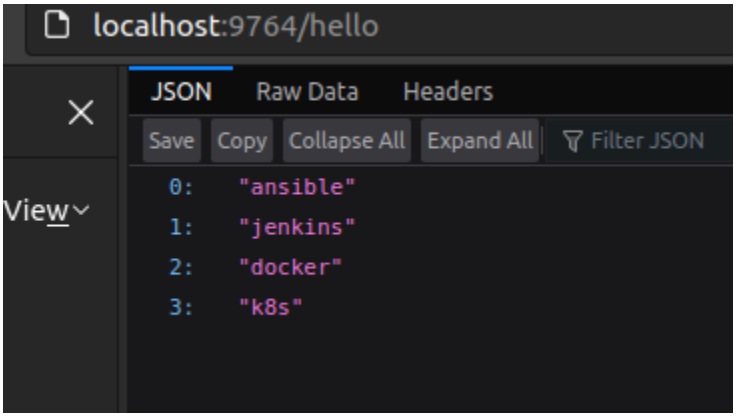



```
devops@devops-VirtualBox:~/Desktop/sam-aws/LambdaDevOps$ sam build
Your template contains a resource with logical ID "ServerlessRestApi", which is a reserved logical ID in AWS SAM. It could result in unexpected behaviors and is not recommended.
Building codeuri: /home/devops/Desktop/sam-aws/LambdaDevOps/hello_world 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
```

Now deploy the application to the lambda function and access the application by invoking the newly created API.

```
devops@devops-VirtualBox:~/Desktop/sam-aws/LambdaDevOps$ sudo sam local invoke

        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/developerguide/serverless-sam-telemetry.html

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-1.51.0-x86_64.

Mounting /home/devops/Desktop/sam-aws/LambdaDevOps/.aws-sam/build/HelloWorldFunction as /var/task:ro,delegated inside runtime container
START RequestId: 95c6c5e4-d3cc-4316-a41d-6d11959cc9ce Version: $LATEST
END RequestId: 95c6c5e4-d3cc-4316-a41d-6d11959cc9ce
REPORT RequestId: 95c6c5e4-d3cc-4316-a41d-6d11959cc9ce  Init Duration: 4.67 ms  Duration: 146.62 ms     Billed Duration: 147 ms Memory Siz
{"statusCode": 200, "body": "[\"ansible\", \"jenkins\", \"docker\", \"k8s\"]", "headers": {"Content-Type": "application/json"}}devops@devo
```

```
devops@devops-VirtualBox:~/Desktop/sam-aws/LambdaDevOps$ sudo sam local start-api -p 9764
Mounting HelloWorldFunction at http://127.0.0.1:9764/hello [GET]
You can now browse to the above endpoints to invoke your functions. You do not need to restart/reload SAM CLI while working on your functions, changes will be reflected instantly/automatically. You only
need to restart SAM CLI if you update your AWS SAM template
2022-06-08 01:58:25  * Running on http://127.0.0.1:9764/ (Press CTRL+C to quit)
2022-06-08 01:59:47 127.0.0.1 - - [08/Jun/2022 01:59:47] "GET / HTTP/1.1" 403 -
Invoking app.lambda_handler (python3.9)
Skip pulling image and use local one: public.ecr.aws/sam/emulation-python3.9:rapid-1.51.0-x86_64.

Mounting /home/devops/Desktop/sam-aws/LambdaDevOps/.aws-sam/build/HelloWorldFunction as /var/task:ro,delegated inside runtime container
END RequestId: 72a83a8b-bb64-4881-a8d8-0a2287afb135
REPORT RequestId: 72a83a8b-bb64-4881-a8d8-0a2287afb135  Init Duration: 1.33 ms  Duration: 159.70 ms     Billed Duration: 160 ms Memory Size: 128 MB     Max Memory Used: 128 MB
2022-06-08 02:00:10 127.0.0.1 - - [08/Jun/2022 02:00:10] "GET /hello HTTP/1.1" 200 -
```



Show the logs in cloudwatch of the new invocations.