

Assignment 8

Submitted by :	
Name	Naman Panchal
E-mail	naman.panchal.ce@gmail.com

Que: 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.



Answer:

Here to implement above AWS architecture, I have used AWS SAM (Serverless Application Model) CLI to build and deploy serverless application on AWS.

A serverless application is a combination of Lambda functions, event sources, and other resources that work together to perform tasks. Ref: <https://docs.aws.amazon.com/serverless-application-model/latest/developerguide/serverless-getting-started-hello-world.html>

Here, I have deployed Hello World application using AWS SAM. 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.

Please find below attached screenshots for this implementation.

AWS SAM prerequisites:

- Install the latest release of the AWS Serverless Application Model Command Line Interface (AWS SAM CLI)

#Step 1 - Download a sample AWS SAM application

➤ `sam init`

```
PS C:\Users\naman> 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]: N

Which runtime would you like to use?
  1 - aot.dotnet7 (provided.al2)
  2 - dotnet6
  3 - dotnet5.0
  4 - dotnetcore3.1
  5 - go1.x
  6 - go (provided.al2)
  7 - graalvm.java11 (provided.al2)
  8 - graalvm.java17 (provided.al2)
  9 - java11
 10 - java8.al2
 11 - java8
 12 - nodejs18.x
 13 - nodejs16.x
 14 - nodejs14.x
 15 - nodejs12.x
```

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

-----
Generating application:
-----
Name: sam-app
Runtime: nodejs16.x
Architecture: x86_64
Dependency Manager: npm
Application Template: hello-world-powertools-typescript
Output Directory: .

Next steps can be found in the README file at ./sam-app/README.md

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

#Step 2 - Build your application

- `cd sam-app`
- `sam build`

```
PS C:\Users\naman> cd .\sam-app\  
PS C:\Users\naman\sam-app> sam build  
Building codeuri: C:\Users\naman\sam-app\hello-world runtime: nodejs16.x metadata: {'BuildMethod': 'esbuild', 'BuildProperties': {'Minify': True, 'Target': 'es2020', 'Ent  
Points': ['app.ts']}} architecture: x86_64 functions: HelloWorldFunction  
Running NodejsNpmEsbuildBuilder:CopySource  
Running NodejsNpmEsbuildBuilder:NpmInstall  
Running NodejsNpmEsbuildBuilder:EsbuildBundle  
  
Build Succeeded  
  
Built Artifacts : .aws-sam\build  
Built Template : .aws-sam\build\template.yaml  
  
Commands you can use next  
=====
```

#Step 3 - Deploy your application

- `sam deploy --guided`

```
PS C:\Users\naman\sam-app> sam deploy --guided  
  
Configuring SAM deploy  
=====
```

Looking for config file [samconfig.toml] : Not found

Setting default arguments for 'sam deploy'
=====

Stack Name [sam-app]:
AWS Region [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]: 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:
Creating the required resources...
Successfully created!
Managed S3 bucket: aws-sam-cli-managed-default-samclisourcebucket-df36mtnio6wr
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 sam-app/6d7b0549c31940143c45f724d6ca0486 45249 / 45249 (100.00%)

Deploying with following values
=====

Stack name	: sam-app
Region	: ap-south-1

```
Uploading to sam-app/6d7b0549c31940143c45f724d6ca0486 45249 / 45249 (100.00%)

Deploying with following values
=====
Stack name           : sam-app
Region              : ap-south-1
Confirm changeset    : True
Disable rollback     : True
Deployment s3 bucket : aws-sam-cli-managed-default-samclisourcebucket-df36mtnio6wr
Capabilities         : ["CAPABILITY_IAM"]
Parameter overrides  : {}
Signing Profiles     : {}

Initiating deployment
=====
Uploading to sam-app/21b30b55e061d36479ec52eaaa980660.template 1568 / 1568 (100.00%)

Waiting for changeset to be created..
CloudFormation stack changeset

-----
Operation              LogicalResourceId              ResourceType                    Replacement
-----
+ Add                  HelloWorldFunctionHelloWorldPermissionPr
                        od
                        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:ap-south-1:166639039766:changeSet/samcli-deploy1673174959/27bb0685-87c6-4a75-83af-707305e0f124

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

2023-01-08 16:19:39 - Waiting for stack create/update to complete
```

```
2023-01-08 16:19:39 - 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    ServerlessRestApiDeployment47fc2d5f9d -
CREATE_IN_PROGRESS      AWS::Lambda::Permission        HelloWorldFunctionHelloWorldPermissionPr
                                od
                                AWS::Lambda::Permission        HelloWorldFunctionHelloWorldPermissionPr
                                od
                                Resource creation Initiated
CREATE_IN_PROGRESS      AWS::ApiGateway::Deployment    ServerlessRestApiDeployment47fc2d5f9d -
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        HelloWorldFunctionHelloWorldPermissionPr
                                od
                                AWS::CloudFormation::Stack    sam-app                          -
-----

CloudFormation outputs from deployed stack

-----
```

```
CloudFormation outputs from deployed stack

-----
Outputs
-----
Key                HelloWorldFunctionIamRole
Description         Implicit IAM Role created for Hello World function
Value              arn:aws:iam::166639039766:role/sam-app-HelloWorldFunctionRole-BNPP6EPhuZ061B

Key                HelloWorldApi
Description         API Gateway endpoint URL for Prod stage for Hello World function
Value              https://0vnuqf7c001.execute-api.ap-south-1.amazonaws.com/Prod/hello/

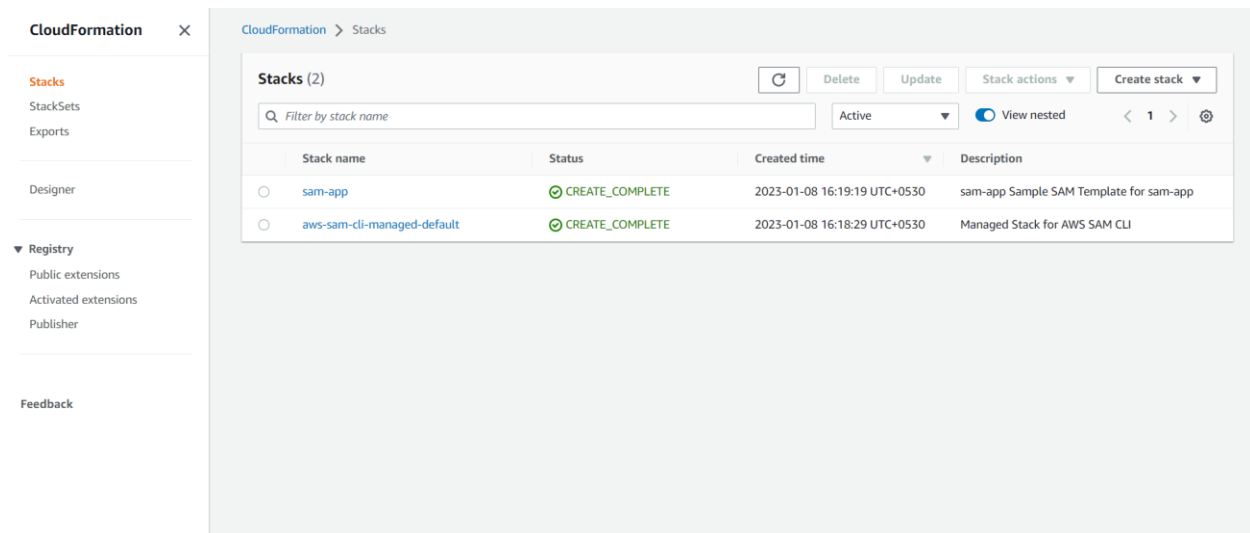
Key                HelloWorldFunction
Description         Hello World Lambda Function ARN
Value              arn:aws:lambda:ap-south-1:166639039766:function:sam-app-HelloWorldFunction-KV70uSejLMCR

-----

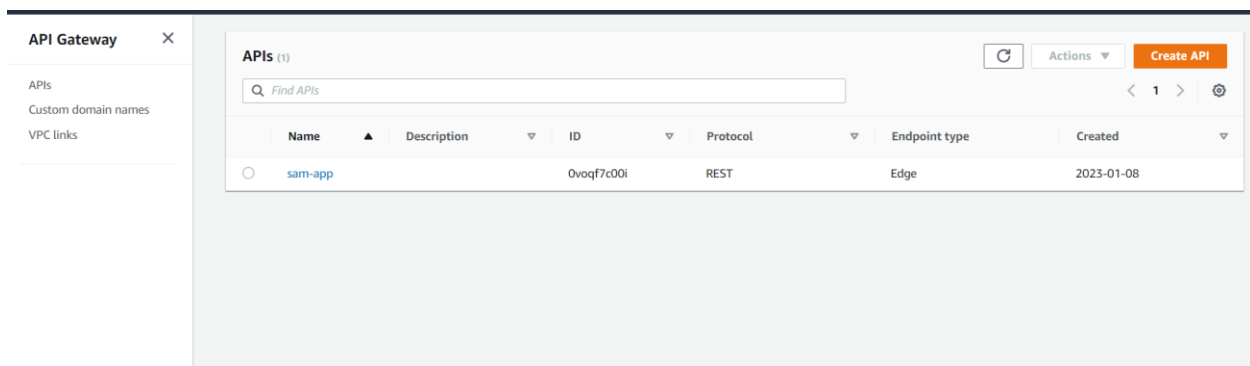
Successfully created/updated stack - sam-app in ap-south-1
```

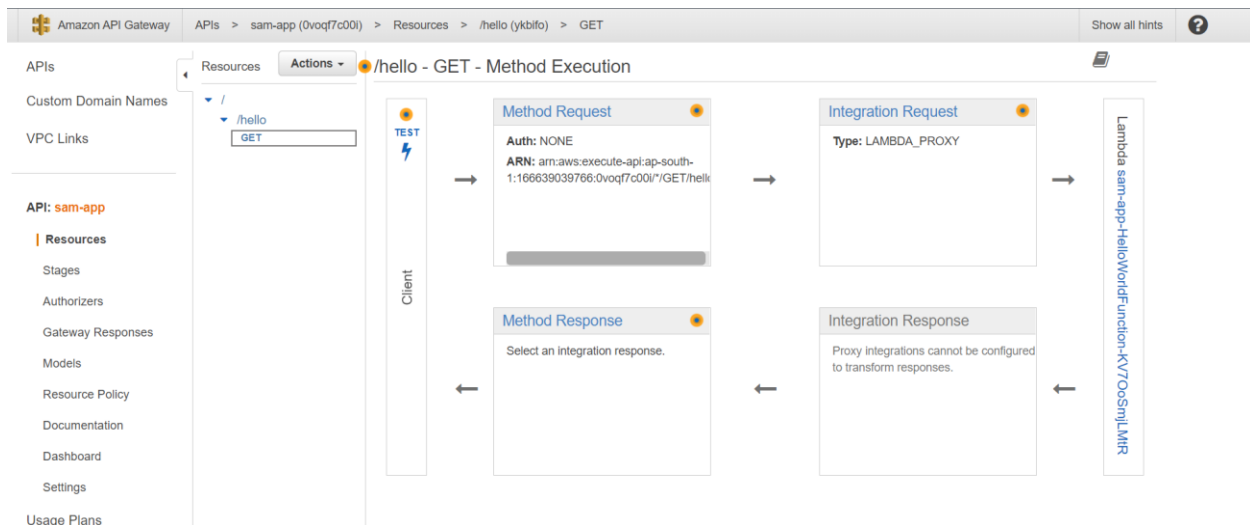
AWS resources:

- **CloudFormation**
 - CloudFormation is creating required components as seen in above screenshots. Same also we can see by logging into AWS management console and then going to CloudFormation.



- **API Gateway**





- **Lambda**

AWS Lambda

Dashboard

Applications

Functions

Additional resources

Code signing configurations

Layers

Replicas

Related AWS resources

Step Functions state machines

Lambda > Applications

Applications (1) Info

Search by keyword

Create application

	Name	Description	Last modified	Status
	sam-app	sam-app Sample SAM Template for sam-app	3 minutes ago	Create complete

AWS Lambda

Dashboard
Applications
sam-app
Functions

▼ Additional resources
Code signing configurations
Layers
Replicas

▼ Related AWS resources
Step Functions state machines

Lambda > Applications > sam-app

sam-app


Overview | Deployments | Monitoring


▼ Getting started Dismiss


Welcome to your new application view. From here, you can view the resources that make up your application, and monitor performance, errors, and traffic metrics. [Learn more](#)


Set up your development environment

You can use the following tools and services to build, test, and deploy your applications.

 [Visual Studio](#)
Quickly create .NET Core functions from a blueprint. Compile, deploy, configure, and test your function from within Visual Studio.

 [Visual Studio Code](#)
Configure the AWS Toolkit for Visual Studio Code to edit your serverless applications in Microsoft Visual Studio Code.

 [JetBrains](#)
Configure the AWS Toolkit for JetBrains to edit your serverless applications in JetBrains IDEs.

 [AWS SAM CLI](#)
Define the infrastructure for your serverless application in an AWS SAM template. Deploy your function and other application resources from the command line. Build, test, and debug function code locally in a Docker container that emulates the Lambda execution environment.

AWS Lambda Partners provide services and tools that you can use with your Lambda functions. [View all the current AWS Lambda Partners](#)



API endpoint

Endpoint
<https://0voqf7c00i.execute-api.ap-south-1.amazonaws.com/Prod>

Resources

(6)

Q

Logical ID	Physical ID	Type	Last modified
 HelloWorldFunction	sam-app-HelloWorldFunction-KV7Oo5njLM8R	Lambda Function	4 minutes ago
 ServerlessRestApi	0voqf7c00i	ApiGateway RestApi	4 minutes ago

When we hit API gateway endpoint, will get {"message": "hello world"} from our own created backend (using Amazon API Gateway endpoint and an AWS Lambda function) as per below screenshot.

← → ↻ 🏠 🔒 0voqf7c00i.execute-api.ap-south-1.amazonaws.com/Prod/hello/ 📄 ☆

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


So, as per infra diagram I have created and deployed necessary infrastructure on AWS.

