**AWS Server-Less Application Model**

**Necessary commands**

**The AWS Serverless Application Model (AWS SAM) CLI** provides a simplified way to build, deploy, and manage serverless applications on AWS. Here is a list of some commonly used AWS SAM CLI commands along with their descriptions and installation steps:

**Installation:**

To install the AWS SAM CLI, follow the installation instructions for your operating system from the official documentation:

[Install the AWS SAM CLI](https://docs.aws.amazon.com/serverless-application-model/latest/developerguide/serverless-sam-cli-install.html)

**Commands:**

1. **aws configure:** command is used to set up the credentials and configuration for the AWS Command Line Interface
2. **sam init:** Initialize a new serverless application based on a template.
3. **sam build:** Build your serverless application code and dependencies.
4. **sam deploy:** Deploy your serverless application to AWS.
5. **sam local invoke:** Invoke a Lambda function locally with sample event data.
6. **sam local start-api:** Start a local API Gateway for testing API endpoints.
7. **sam logs:** Fetch logs for deployed Lambda functions.
8. **sam validate**: Validate a SAM template.
9. **sam package:** Package and upload your application code and CloudFormation templates to an S3 bucket.
10. **sam publish:** Publish a Lambda Layer version.
11. **sam deploy:** Deploy your application to AWS.
12. **sam delete:** Delete a deployed application from AWS.
13. **sam package:** Package your application artifacts for deployment.
14. **sam local generate-event:** Generate sample event payloads for testing.
15. **sam local start-lambda:** Start a local endpoint for invoking Lambda functions.
16. **sam local generate-event:** Generate sample event payloads for testing.
17. **sam local start-lambda:** Start a local endpoint for invoking Lambda functions.
18. **sam build --use-container:** Use a Docker container to build your application, enabling consistent build environments.
19. **sam deploy --guided:** Interactively guide the deployment process, prompting for deployment options.
20. **sam publish layer-version:** Publish a Lambda Layer version to the AWS Serverless Application Repository.
21. **sam logs -n <function-name>:** Fetch logs for a specific Lambda function.
22. **sam teardown:** Tear down the resources created by a SAM deployment.

**Follow steps to build lambda**

**For help and looking commands and other information-”aws sam help ”**

**🡪Step 1 First step to Enter your aws account credentials to used-“aws configure”**

C:\Users\Admin>aws configure

AWS Access Key ID [\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*YBGT]:

AWS Secret Access Key [\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*F8C3]:

**🡪Step 2 Sam initialize- type” sam init ”**

C:\Users\Admin>**sam init**

You can preselect a particular runtime or package type when using the `sam init` experience.

Call `sam init --help` to learn more.

**🡪Step 3 Select Quick Templates-Enter” 1”**

**Which template source would you like to use?**

1 - AWS Quick Start Templates

2 - Custom Template Location

Choice: **1**

**🡪Step 4 Select Template-Enter “1”**

Choose an AWS Quick Start application template

1 - Hello World Example

2 - Data processing

3 - Hello World Example with Powertools for AWS Lambda

4 - Multi-step workflow

5 - Scheduled task

6 - Standalone function

7 - Serverless API

8 - Infrastructure event management

9 - Lambda Response Streaming

10 - Serverless Connector Hello World Example

11 - Multi-step workflow with Connectors

12 - Full Stack

13 - Lambda EFS example

14 - Hello World Example With Powertools

15 - DynamoDB Example

16 - Machine Learning

Template: **1**

**Default Language**

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

**🡪Step 5 for Select Programming Language-Enter”14” for python**

Which runtime would you like to use?

1 - aot.dotnet7 (provided.al2)

2 - dotnet6

3 - go1.x

4 - go (provided.al2)

5 - graalvm.java11 (provided.al2)

6 - graalvm.java17 (provided.al2)

7 - java17

8 - java11

9 - java8.al2

10 - java8

11 - nodejs18.x

12 - nodejs16.x

13 - nodejs14.x

14 - python3.9

15 - python3.8

16 - python3.7

17 - python3.11

18 - python3.10

19 - ruby3.2

20 - ruby2.7

21 - rust (provided.al2)

Runtime: 14

**🡪Step 6 Select the package-Enter”1”**

What package type would you like to use?

1 - Zip

2 - Image

Package type: **1**

**Default settings to change according to your needs**

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]:

Would you like to enable monitoring using CloudWatch Application Insights?

For more info, please view

**🡪Step 7 please Enter the name your application-“helloworld”**

Project name [sam-app]: **helloworld**

-----------------------

Generating application:

-----------------------

Name: helloworld

Runtime: python3.9

Architectures: x86\_64

Dependency Manager: pip

Application Template: hello-world

Output Directory: .

Configuration file: helloworld\samconfig.toml

Commands you can use next

=========================

[\*] Create pipeline: cd helloworld && sam pipeline init --bootstrap

[\*] Validate SAM template: cd helloworld && sam validate

[\*] Test Function in the Cloud: cd helloworld && sam sync --stack-name {stack-name} --watch

**🡪Step 8 Please select the your application directories using linux commands**

C:\Users\Admin>**cd helloworld**

**🡪Step 9 Build your serverless application code and dependencies-Enter “sam build”**

C:\Users\Admin\helloworld>**sam build**

Starting Build use cache

Manifest file is changed (new hash: 3298f13049d19cffaa37ca931dd4d421) or dependency folder (.aws-sam\deps\ff3117fe-605c-42ce-be61-517917093bbd) is missing for

(HelloWorldFunction), downloading dependencies and copying/building source

Building codeuri: C:\Users\Admin\helloworld\hello\_world runtime: python3.9 metadata: {} architecture: x86\_64 functions: HelloWorldFunction

Running PythonPipBuilder:CleanUp

Running PythonPipBuilder:ResolveDependencies

Running PythonPipBuilder:CopySource

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

**🡪Step 10 Deploy your application on cloud/lambda-function-Enter”sam deploy-g”&”sam deploy --guided”**

C:\Users\Admin\helloworld>**sam deploy --guided**

Configuring SAM deploy

======================

Looking for config file [samconfig.toml] : Found

Reading default arguments : Success

Setting default arguments for 'sam deploy'

=========================================

Stack Name [helloworld]:

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]:

#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]:

#Preserves the state of previously provisioned resources when an operation fails

Disable rollback [y/N]:

HelloWorldFunction has no authentication. Is this okay? [y/N]: **y**

Save arguments to configuration file [Y/n]:

SAM configuration file [samconfig.toml]:

SAM configuration environment [default]:

Looking for resources needed for deployment:

Managed S3 bucket: aws-sam-cli-managed-default-samclisourcebucket-1ojavc6pppsmw

A different default S3 bucket can be set in samconfig.toml and auto resolution of buckets turned off by setting resolve\_s3=False

Parameter "stack\_name=helloworld" in [default.deploy.parameters] is defined as a global parameter [default.global.parameters].

This parameter will be only saved under [default.global.parameters] in C:\Users\Admin\helloworld\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 helloworld/49855c9d3a2cb8aaa2b29ee9e71a5ba8 608599 / 608599 (100.00%)

Deploying with following values

===============================

Stack name : helloworld

Region : us-east-1

Confirm changeset : True

Disable rollback : False

Deployment s3 bucket : aws-sam-cli-managed-default-samclisourcebucket-1ojavc6pppsmw

Capabilities : ["CAPABILITY\_IAM"]

Parameter overrides : {}

Signing Profiles : {}

Initiating deployment

=====================

Uploading to helloworld/06ade3760dcd4edede0633bb4dbd2dca.template 1254 / 1254 (100.00%)

Waiting for changeset to be created..

CloudFormation stack changeset

-------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Operation LogicalResourceId ResourceType Replacement

-------------------------------------------------------------------------------------------------------------------------------------------------------------------------

+ Add HelloWorldFunctionHelloWorldPermissionPr AWS::Lambda::Permission N/A

od

+ 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-1:899589157385:changeSet/samcli-deploy1690884181/38c0bb1e-3e6e-4227-91b4-11b293cc3eac

Previewing CloudFormation changeset before deployment

======================================================

Deploy this changeset? [y/N]: **y**

2023-08-01 15:33:59 - Waiting for stack create/update to complete

CloudFormation events from stack operations (refresh every 5.0 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 HelloWorldFunctionHelloWorldPermissionPr -

od

CREATE\_IN\_PROGRESS AWS::ApiGateway::Deployment ServerlessRestApiDeployment47fc2d5f9d -

CREATE\_IN\_PROGRESS AWS::Lambda::Permission HelloWorldFunctionHelloWorldPermissionPr Resource creation Initiated

od

CREATE\_COMPLETE AWS::Lambda::Permission HelloWorldFunctionHelloWorldPermissionPr -

od

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::CloudFormation::Stack helloworld -

-------------------------------------------------------------------------------------------------------------------------------------------------------------------------

CloudFormation outputs from deployed stack

-------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Outputs

-------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Key HelloWorldFunctionIamRole

Description Implicit IAM Role created for Hello World function

Value arn:aws:iam::899589157385:role/helloworld-HelloWorldFunctionRole-Q2XRI98QYBB6

Key HelloWorldApi

Description API Gateway endpoint URL for Prod stage for Hello World function

Value https://p12gy7ny2l.execute-api.us-east-1.amazonaws.com/Prod/hello/

Key HelloWorldFunction

Description Hello World Lambda Function ARN

Value arn:aws:lambda:us-east-1:899589157385:function:helloworld-HelloWorldFunction-NyUqP4mfWOgX

-------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Successfully created/updated stack - helloworld in us-east-1

C:\Users\Admin\helloworld>