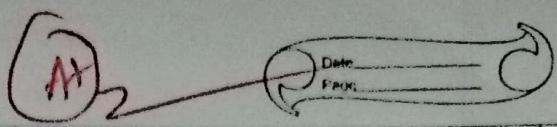


Name: Shejal Nilesh Tiwari
Div: D15C Roll No: 57
ADevOps Assignment No: 2



Create a REST API with Serverless Framework.

] Creating and configuring the project
For that we will run the command:

--template aws-nodejs --path my-rest-api
This will create new project serverless.

This generates a directory with basic serverless.yml configuration file.

] Editing the serverless.yml

) Open serverless.yml and define your API structure

Minimal configuration looks like service:
my-rest-api
provider:

name: aws
runtime: nodejs14.x

functions:

create item:

handler: handler.create

events:

http:

path: items

method: post

This defines end points for creating an item and retrieving an item by its id.

3] Writing Lambda Functions

- In project folder, create a handler.js file to handle the logic for the endpoints.
- Example code for creating an item

```
module.exports.create = async(event) => {
    const item = JSON.parse(event.body);
    return {
        statusCode: 200,
        body: JSON.stringify({ message: 'item created', item })
    }
}
```

4] Deploying the REST API

Run the command 'serverless deploy' to deploy project. This command provisions the AWS resources and returns the API's base URL

5] Testing the API

Once deployed, test the API using tools like Postman. To test the POST request

```
curl -X POST https://your-api-url/items
-d '{"name": "Book", "price": 15}'
```

At a large organization, your centralized operations team may get many repetitive infrastructure requests. You can use Terraform to build a "self serve" infrastructure model that lets products team manage their own infrastructure independently. You can create and use Terraform modules that codify the standards for deploying and managing services in your organization, allowing teams to efficiently deploy services in compliance with your organization's practices. Terraform Cloud can also integrate with ticketing systems like ServiceNow to automatically generate new infrastructure requests.

i) In the question they have asked for creating a self-service module, so we will create a module to create an EC2 instance for the product team.

ii) For creating a CLI to manage the EC2 instance creation, we will start by downloading AWS CLI download and Terraform download.

Now we will configure aws cli. To configure it is important to create access key and secret key for your aws account and check your region name for my account it was us-east-1 (Virginia). You can select output format as json.

- iv) Set up a terraform project directory and you need to create 3 files in it main.tf, variables.tf, output.tf.
main file contains infrastructure configuration
variables.tf contains input variables that you can customize.
output: This will define the output variables.
- v) Then start with terraform init (To setup terraform directory to hold metadata about project)

- vi) Plan your infrastructure Deployment:
terrafrom plan -var = "ami_id = ami-0c55b159cbfafef0"\br/>- var = "instance_name = MyWebServer"\br/>- var = "instance_type = t2.micro".
- vii) Apply changes to deploy EC2 Instances

terrafrom apply -var = "ami_id = ami-0c55b159cbfafef0"\br/>- var = "instance_name = MyWebServer"\br/>- var = "instance_type = t2.micro".

- viii) You will get the output

instance_id = "i-0abcdef1234".
instance_id = "i-0317d7703da4a0967"
instance_public_ip = "3.93.47.25"

- ix) You can destroy for removing the resources
terrafrom destroy -var = "ami_id = ami-0c55b159cbfafef0"\br/>- var = "instance_name = MyWebServer"\br/>- var = "instance_type = t2.micro".

Case Study for Sonarqube.

Create or own profile in sonarqube for testing project quality

Use Sonarcloud to analyze your Github code.

Install sonarlint in or Java intelliJ ide or eclipse ide and analyze your Java code.

Analyze python project with sonarqube

Analyze node.js project with sonarqube

i) Setting up your profile for sonarqube →
 creating a profile in sonarqube allows developers to analyze the quality of their projects and track improvements over time.

Steps for setup :

- a) Install sonarqube from official website and set it up locally.
- b) Once logged in, create a new project on the sonarqube dashboard by providing a project name and key.
- c) Add the sonar-project.properties file to the root directory of the project, which contains the necessary configurations.
- d) Use the sonarqube scanner to analyze your project and uploads the result to dashboard.
- e) Using sonarcloud to Analyze Github Code →
 sonarcloud is a cloud based service that integrates with popular version control platforms like Github.

Steps to Analyze Code →

- a) Sign up and Connect Github Repository: Sign for SonarCloud using your Github account and authorize it to access your repository.
- b) Set up Github Actions to run SonarCloud scan whenever code is pushed into the repository. This ensures continuous code quality analysis as part of CI/CD.
- c) Create a sonar-project.properties file with the necessary configurations in the root directory of the project.
- d) The SonarCloud scan is triggered automatically on every push.

3] SonarLint for Real Time Code Analysis in IDE

SonarLint is a plugin for IntelliJ IDEA and Eclipse that performs static code analysis in real-time, helping developer catch issues early during development.

Steps to install and use SonarLint →

- a) Install the Plugin, for IntelliJ IDEA go to File > Settings > Plugins > Scratch Search for SonarLint and install.
- b) SonarLint can be linked to SonarQube instance to sync rules and quality profiles.
- c) SonarLint runs automatically as you write code and flags any issues directly in the editor.

Sonarlint provides detailed explanations and suggestions for resolving issues.

Analyzing a Python Project with Sonargube →
Sonargube also supports python projects and help detect common issues like bugs, code etc.

Steps to Analyze a python project →

Ensure that sonargube is running and has Python plugin installed.

Configuring Sonargube for Python ie specify the Python source files.

Execute sonar-scanner from the root of your python project. The analysis will upload the results to sonargube.

Analyzing a Node.js Project with Sonargube →
Node.js projects can also benefit from static analysis through Sonargube. By integrating it into Node.js, we can identify common code issues related to Javascript.

Steps to Analyze a Nodejs Project →

Verify that the Javascript plugin is available in your Sonargube instance.

Configure Project for Sonargube by adding some properties in sonar-project.properties file.

- 3] You can combine EC2 instance with SonarQube for a more comprehensive Javascript analysis.
- 4] Use Sonar-Scanner to analyze the project and view the detailed analysis in the SonarQube dashboard.

Conclusion →

By integrating tools like SonarQube, SonarScanner and SonarLint into your development process, you can continuously monitor code quality across various programming languages.