Name - Chinmay H Chaudhari
DIV - D15C
Roll No - 6

05/05

Assignment 2

Create a REST API with the serverless framework.
Steps to create a REST API with serverless framework:

1) Install serverless framework globally using the following command on the terminal.

```
npm install -g serverless
```

This command installs the serverless framework on your machine globally using npm.

2) Create a new service with AWS Nodejs template:

```
serverless create --template aws-nodejs --path rest-api
```

This command intitialises a new serverless service called rest-api.

3) Navigate to the project directory

```
cd rest-api
```

This command changes the directory

4) Initialise Node.js project and install dependencies.

```
npm init -y
npm install express serverless-http
```

Teacher's Sign.: _____

Q2 Case study for Sonarqube

→ ① Create your own profile in Sonarqube

a. Download and install sonarqube from the official website.

b. Log into sonarqube using the default credent (username: ~~adim~~admin, password: admin)

c. Navigate to project tab, click on 'Create new assign a project key and name and genrate project token.

② Use sonarqube to analyse your Github code.

a. Sign up for Sonar Cloud from the official u using your Github account.

b. In sonarcloud under projects > Create project choose your Github repository and grant

c.

⑤ Edit the serverless yml file to include

```
service: rest-api
provider:
  name: aws
  runtime: nodejs14.1
  stage: dev
  region: us-~~est~~-east-1
functions:
  app:
    handler: handler.app
    ~~ww~~events:
      - http:
          path: /
          method: any
```

This configuration specifies the service name AWS provider settings and defines the lambda function with HTTP event trigger.

Edit handler.js to add the Express app

```
const express = require("Express")
const serverless = require("serverless-http");
const app = express()

app.get('/hello-world', (req, res) => res.json({ message:
    "hello world" }));
module.exports.app = serverless(app);
```

⑦ Deploy the service

Deploys the API to AWS setting up resources l
lambda and API Gateway. A URL is generate
for testing.

⑧ Test the deployed API

Curl https://<api-id>.execute-api.<region>.ama
com/dev/helloworld

⑨ Redeploy after updates

After modifying the code, regarding it to upd
the API with our changes.

⑩ Remove the services.

The above command requires removes all AWS
resources associated with the API ensuring th
there are no charges for unused services.

Case study for Sonarqube

1) Create your own profile in sonarqube
   a. Download and install sonarqube 6.6 from the official website, unzip the file and start the server by running.

   b. Log in to sonarqube using the default credentials (username: admin, password: admin)

   c. Navigate to projects tab, click on 'Create new project' assign on project key and name and generate a project token.

2) Use Sonarcloud to analysis your Github code
   a. Sign up for Sonar Cloud from the official website using your Github account.

   b. In sonarcloud under projects > create project choose your github repository and grant sonarcloud access to it.

   c. Add a sonar projects properties file in the root of your repository with the following code

   Sonar project key : <your-project key>
   Sonar.organisation : <your-organisation>

Teacher's Sign.:

③ Use sonarscanner to analyse the code key by running the following command: sonar-scanne This uploads analysis results from your loc machine to sonar cloud.

④ Install Sonar lint in your Java IntelliJ or Eclip and analyse your Java code.

a. Install sonarlint by going onto IntelliJ or Eclipse going to Plugins / Market place an search for Sonarlint to install and restart

b. In the IDE configure sonarlint by linking your sonarqube or Sonar cloud project to the rules and profiles.

c. Open a Java project and vice-versa Sonarlint analyse it. It will display issues directly in IDE while coding.

⑤ Analyse python project with sonarqube.

a. Set up a python code in a project and en that sonarqube is running locally

⑥ Download and configure sonar scanne from its official website and in the son project properties file, edit to idude the following

Teacher's Sign.:

⑥ Analyse Node.js project with Sonarqube

a. Set up a Node.js project

b. In sonarcube, ensure that all JS/TS plugins have been installed. Plugins can be installed from the marketplace tab in sonarqube.

c. Create sonar-project.properties file in your project root and include the following in it

```
sonar.projectKey = node-project
sonar.language = js
sonar.sources = .
```

d. Run the analysis of the project by executing the sonar-scanner command.
Sonarqube will analyse the Node.js project and show results on the dashboard highlighting code quality, bugs and vulnerabilities

Teacher's Sign

Q3 In organisations managing repetitive infrastruct
requests strain centralised operations teams, s
down process adopting a self-serve infrastructu
model using terraform decentralisation this
responsibility, empowering produce teams to mana
their own infrastructure.

→ ① In large organisations often rely on centralise
operations teams to handle repetitive infrastruc
② This centralized approach can lead to delays,
inefficiencies, and bootlenecks in service deploymen
③ Using Terraform, organisations can implement a
self-service infrastructure model, enabling pro
teams to manage their own infrastructure.
④ Terraform modules codify organizational stan
ensuring consistent, secure, and compliant infra
deployment.
⑤ These ~~test~~ reusable modules allows teams to d
services efficiently, following best practices.
⑥ Terraform cloud can integrate with ticketing
systems like servicenow to automate the c
of infrastructure requests.
⑦ This integration streamlines the process by au
generating and tracking requests.
⑧ The self service model enhances agility, reduc
operational bottlenecks, and empowers teams
take control of their infrastructure.

Teacher's Sign.: