Vedant Dhoke
D15C Rollno → 9
Adv. Devops assignment 2

Date
Page 85

Q.1] Create a REST API with Serverless Framework.

1] Creating and configuring the project

- Run the command to create new project. command: serverless create --template aws-nodejs --path my-rest-api
- This generates a directory with basic serverless.yml configuration file.
-

2] 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:
    createItem:
        handler: handler.create
        events:
            - http:
                path: items
                method: post
```

- This defines end points for creating an item and Retrieving an item by it's ID.

Teacher's Sign.: ____________

Date
Page

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 {
            Status code : 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 ~~the~~ test the POST request.

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

Teacher's Sign.: ____________

Date
Page

Q.2] Case study for 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 :-> 1] Install Sonarqube from official website and set it up locally.
2] Once logged in, create a new project in the Sonarqube dashboard by providing a project name and key.
3] Add the sonar-project.properties file to the root directory of the project, which contains the necessary configurations.
4] Use the sonarqube Scanner to analyze your project and uploads the result to dashboard.

ii] Using SonarCloud to Analyze Github Code → Sonar Cloud is a cloud based service that integrates with popular version control platforms like Github.

Steps to Analyze Code →

1] Sign up and Connect Github Repository. Sign up for SonarCloud using your Github account and authorize it to access your repositories.
2] Set up Github Actions to run SonarCloud scans whenever code is pushed into the repository. This ensures continuous code quality analysis as part of CI/CD.
3) Create a sonar-projects.properties file withe the necessary configurations in the root directory of the project.
4] The SonarCloud scan is triggered automatically on everypush.

Teacher's Sign.:

3] SonarLint for Real time Code Analysis in IDEs → SonarLint is a plugin for intellij IDEA and Eclipse that performs static code analysis in real - time, helping developers catch issues early during development.

Steps to install and use SonarLint →

1] Install the Plugin, for IntelliJ IDEA goto File > settings > Plugins search for SonarLint and install
2] SonarLint can be linked to Sonarqube instance to sync rules and quality profiles.
3] SonarLint runs automatically as you write code and flags any issues directly in the editor.
4] Ssonarlint provides detailed explanations and suggestions for resolving issues.

4] Analyzing a Python Project with Sonarqube → SonarQube also supports Python projects and help detect common issues like bugs, codes etc.

Steps to Analyze a python project →

1] Ensure that SonarQube is running and has Python plugin installed.
2] Configuring SonarQube for Python i.e specify the Python source files.
3] Execute sonar- scanner from the root of your python project. The analysis will upload the results to SonarQube.

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

Steps to Analyze a Node.js Project →

1] Verify that the Javascript plugin is available in your SonarQube instance.
2] Configure Project for SonarQube by adding some properties in sonar-project.properties file.
3] You can combine ESLint 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, SonarCloud and SonarLint into your development process, you can continuously monitor code quality across various programming languages.

Q.3] At a large organization your centralized operation 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 organizations 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 ~~sef~~ 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 AWSCLI download and Terraform download.

iii] 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 configurations, variables.tf contains input variables that you can customize. Output: This will define the output values.

Teacher's Sign.: ________

Date
Page

vi) Then start with
terraform init . (To setup terraform directory to hold meta data about projects).

vii) Plan your infrastructure Deployment.
```
terraform plan - var = "ami-id = ami - 0c55b159cbfafe1f0"\
- var = "instance_name = My webserver"\
- var = "instance_type = t2.micro".
```

viii) Apply changes to Deploy EC2 Instances

```
terraform apply -var = "ami-id = ami-0c55b159cbfafe1f0")
- var = "instance_name = my webserver"\
- var = "instance_type = t2.micro".
```

xi) ~~Yo to~~ You will get the output

```
instance-id = "i- 0317d7703da4a0967".
instance -public-ip = "3.93.47.25"
```

Teacher's Sign.: