04/05

Assignment 2

Create a REST API with the serverless framework

Step 1 : Install Prerequisites

(a) Install Node.js from the official website

(b) Install the serverless framework globally using npm

```
npm install -g serverless
```

Step 2 : Create a Serverless project

(a) Create a new directory for your project & navigate into it

(b) Initialize a new serverless service using the AWS Node.js template

Step 3 : Setup AWS credentials

(a) Configure AWS CLI if not already set up

(b) Set up your AWS Access Key, Secret Key, Region & output Format

Step 4: Modify serverless.yml configuration

(a) Open serverless.yml & define your REST endpoints, Lambda functions, and even

(b) Add HTTP event triggers for the CRUD operations (Post, GET, PUT, DELETE)

Step 5: Write Lambda Function Handlers

(a) In handler.js, write the function to handle the API requests (eg. create, retrieve, update, delete)

Step 6: Deploy your service

(a) Deploy the REST API to AWS

(b) After deployment, note down the API gateway endpoints URLs from the console

Step 7: Test the API

Use tools like curl, Postman or browser to test the deployed API using the provided URLs

Case study for Sonarqube:

- Create your own profile in sonarqube
- Use sonarcloud to analyse your github code
- Install sonarlint in your Java intellij id or eclipse ide & analyse your java code
- Analyse python project with sonarqube
- Analyse node.js project with sonarqube
- Analyse node js project with sonarqube

Sol:

In modern software development, maintaining code quality is paramount for ensuring the long term sustainability & reliability of a project. Sonarqube & sonarcloud offer powerful tools for static code analysis, helping developers identify and fix issues in their code.

① Setting up Sonarqube for local testing

- Installation: Download & install Sonarqube from sonarqube.org
- set up a database for sonarqube
- start the sonarqube server by running the sonar.sh script in the sonarqube installation directory
- Open http://localhost:9000 in your browser & login and create a new project.

(2) Use sonarcloud to Analyse your github code

- Go to sonarcloud
- Sign up with your github account & give sonarcloud access to your repositories
- Create a new project & select the github repository to analyse
- Configure the project key and organization
- Follow the provided instruction to set up sonar-project.properties file in the root of your repository
- Use github actions, Jenkins or manually trigger the analysis using sonarscanner analyse the code quality in the cloud

(3) Install sonarlint in IntelliJ IDEA or eclipse for Java code Analysis

- Open IntelliJ IDEA and navigate to File → Settings → Plugins
- Search for "sonarlint" and install it
- Restart IntelliJ after the installation is complete
- Open your java project, right-click and select Analyse → sonarlint. This will analyse the code and highlight any issues

(2) Use sonarcloud to Analyse your github code

- Go to sonarcloud
- Sign up with your github account & give sonarcloud access to your repositories
- Create a new project & select the github repository to analyse
- Configure the project key and organization
- Follow the provided instruction to set up sonar-project.properties file in the root of your repository
- Use Github actions, Jenkins or manually trigger the analysis using sonarscanner to analyse the code quality in the cloud

(3) Install sonarlint in IntelliJ IDEA or Eclipse for Java code Analysis

- Open IntelliJ IDEA and navigate to File → Settings → Plugins
- Search for "sonarlint" and install it
- Restart IntelliJ after the installation is complete
- Open your java project, right-click and select Analyse → sonarlint. This will analyse the code and highlight any issues

Analyse Python Project with sonarqube

Install the sonarqube scanner on your machine
In your Python project directory, create a
sonar-project.properties file with the
following content

```
sonar.projectKey = your-project-key
sonar.sources = .
sonar.language = py
sonar.python.version = 3.x
sonar.sourceEncoding = UTF-8
```

Run the analysis by executing the following
command in the terminal

```
- sonar-scanner
```

After analysis, view the results on your
sonarqube dashboard

Analyse Node.js project with sonarqube

In your Node.js project directory, create a
sonar-project.properties file with the same
content as mentioned in the previous case
Install SonarScanner globally if not installed
Run the analysis using sonar-scanner
Check the analysis results on the SonarQube
dashboard to identify and fix potential
issues

Q3 Creating a self-service infrastructure model using Terraform for a large organisation involves the following steps:

- Step 1:- Define Infrastructure standards:-
  Establish clear standards and best practices for infrastructure deployment, including naming conventions, resource types, tagging policies and security compliance. This foundation ensures consistency across the organisation.

- Step 2:- Create Terraform modules:-
  Develop reusable Terraform modules based on your organisation's standards. Each module defines resources and configurations, allowing teams to deploy infrastructure efficiently & consistently.

- Step 3:- Set up terraform cloud or enterprise
  Use terraform cloud for centralized management of configurations & state files, enabling collaboration & access control for infrastructure changes.

- Step 4:- Configure version control
  Integrate terraform modules with version control (eg. Github). This tracks changes, facilitates collaboration & ensures proper versioning for updates & compatibility.

Step 5:- Integrate with service now to automate infrastructure requests. This trigger runs terraform and thus streamlining the process for teams

Step 6:- Provide Documentation & training :-

Create Documentation & training for using terraform modules & submitting requests helping teams understand & follow best practices

Step 7:- Monitor & support :-

Monitor the usage of the self-service model & provide ongoing support to users. Gathering feedbacks helps identify that the infrastructure remains compliant and super efficient.

By following above steps, organisations can enable product teams to manage their own infrastructure through a standardised terraform approach