05/05

Page 1

Advance DevOps Assignment 2

1) Create a REST API with the serverless framework
Steps to create a REST API with serverless frame work
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. It allows you to create manage and deploy serverless applications across various cloud providers, including AWS.

2) Create a new Service with AWS Node.js template : ~~serverless~~
create -- template aws-nodejs --path rest-api. This command initialises a new ~~technolo~~ serverless service called rest-api. It creates a folder containing basic files and a template specifically configured for building serverless applications using Node.js on AWS Lambda.

3) Navigate to the project directory :

```
cd rest-api
```

This command changes directory into the newly created project directory to manage files and configurations specific to our service.

4) Initialize Node.js project and install dependencies.

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

The Express dependency builds the REST API and serverless-http integrates express with AWS Lambda

Teacher's Sign.: _____

[Q2] Case study for Sonarqube :-

- Create your own profile in Sonarqube for testing project quality
- Use Sonarcloud to analyse your Github code.
- Install ~~Sonarqube~~ Sonarlint in your Java Intellij IDE or Eclipse IDE and analyse your Java Code.
- Analyse Python project with sonarqube.
- Analyse Node.js project with sonarqube.

Ans] Create your own profile in Sonarqube

1] Download and install Sonarqube from the official website

Unzip the file and start the server by running:
/bin/windows-x86-64/startSonar.bat.
This launches Sonarqube locally and can be accessed at http] cancel.

5) Edit the serverless yml file to include :

```
service: rest-api
provider:
  name: aws
  runtime: nodejs 14.x
  stage: dev
  region: us-east-1
functions:
  app:
    handler: handler.app
    events:
      - http
        path: /
        methods: any
```

Teacher's Sign.:

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('/helloworld', (req, res) => res.json({ message:
'Hello World' }));
module.exports.app = serverless(app);
```

This creates a simple Express app with a single route /helloworld and exports it in a Lambda-compatible format

Deploy the service:

```
serverless deploy
```

Deploys the API to AWS setting up resources like Lambda and API Gateway. A URL is generated for testing

Test the deployed API:

```
curl https://<api-id>.execute-api.<region>.amazonaws.com/
dev/helloworld
```

Using above command, it returns a JSON message:-

```
{"message": "Hello world"}
```

Redeploy after updates: `serverless deploy`

After modifying redeploy it to update the API with our changes.

Remove the service: `serverless remove`

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

Teacher's Sign.: _____

Q.2] Case study for sonarqube :-

- Create your own profile in sonarqube for testing project quality
- Use Sonarcloud to analyze your GitHub code
- Install ~~Sonarqube~~ Sonarlint in your Java IntelliJ IDE or Eclipse IDE and analyze your Java code
- Analyze Python project with sonarqube
- Analyze Nodejs project with sonarqube

Ans] Create your own profile in Sonarqube :-

1] Download and install SonarQube from the offical website unzip the file and start the server by running ./bin/windows-x86-64/startSonar.bat This launches Sonarqube locally and can be accessed at http://localhost:9000.

2] Log in to SonarQube using the default credentials (user-name: admin, password: admin). After logging in, change the password.

3] Navigate to Projects tab, click on 'Create New Project' assign a project key and name and generate a project token.

- Use SonarCloud to analyze your Github code :-

1] Sign up for sonarcloud from the official website using your github account.

2] In SonarCloud under projects > Create Project, choose your Github repository and grant SonarCloud access to it.

3] Add a sonar-project properties file in the root of your repository with the following code :

sonar.projectKey : <your-project-key>

sonar.organisation : <your-organisation>

Teacher's Sign.:

sonar host url = https://sonarcloud.io
Use Sonar Scanner to analyse the code by running the following command : sonar-scanner
This uploads analysis results from your local machine to SonarCloud.

Install SonarLint in your Java IntelliJ or Eclipse IDE and analyse your Java code.
Install SonarLint by going onto IntelliJ or Eclipse, going to Pluggins / Marketplace and search for SonarLint install and restart your IDE
In the IDE, configure SonarLint by linking it to your, SonarQube or SonarCloud project to sync the rules and profiles.
Open a Java project and use SonarLint to analyse it. It will display issues directly in the IDE while coding.
SonarLint provides ~~real-time~~ feedback on code quality based on SonarQube rules.

Analyze Python Project with SonarQube:-
Set up a Python code in a project and ensure that SonarQube is running locally
Download and configure SonarScanner from its offical website and in the sonar-project properties, file edit to include the following:-
Sonar projectKey : python project
Sonar language : py
Sonar sources:
Run the analysis of the projects by executing the following from the project directory : sonar-scanner. The results will be

Teacher's Sign.:

pushed to your local SonarQube server and the analysis is visible on the dashboard.

- Analyse NodeJS project using sonarQube:-

1] Set up a Node.JS project

2] In SonarQube ensure that all Javascript / Typescript plugins have been installed. Plugins can be installed from the marketplace tab in SonarQube.

3] create sonar-project properties file in your project root and include the following in it:-
sonar.projectkey = node_project
sonar.language = js
sonar.sources =

4] Run the analysis of the project by sonar-scanner command.
SonarQube will analyze the Node.js project and show results on the dashboard, highlighting code quality, bugs and vulnerabilities.

Q.3] In organisations managing repititive infrastructure requests can strain centralized operations teams, slowing down processes. Adopting a self-serve infrastructure model using Terraform decentralizes this responsibility.

=> In large organizations, centralized operations teams' repetitive infrastructure requests, leads to delays. Terraform provides a solution by enabling a self-service infrastructure model, enabling product teams to manage their own infrastructure.

=> Terraform modules codify organizational standards,

Teacher's Sign.: