Name : Himanshu Naik
D15C / 63

Page No.
Date

Assignment 2

Create REST API with serverless framework

a) Install serverless framework globally using the following command on terminal

npm install -g serverless

This command install serverless framework on machine globally using npm It allows you to create manage & deploy serverless applications across various cloud providers including AWS.

) Create a new service with aws node js template serverless create template aws node js -. path rest api. This command initialises a new serverless service called rest.api including basic files & template configures wrt NodeJS & AWS lambda

) Navigate the direct Projectory.

d) Initialise NodeJS project & install all the dependencies npm init y
It builds the REST API & serverless http integrates with AWS lambda

```
service: rest api
 provider:
      name: aws
 runtime: nodejs, 14.8
   stage: dev
    region: useast1
functions
 apps
    handler: handler.app
 events:
        Path: /
         method: any
```

Teacher's Sign

This configures specifies service name, w
provider settings & defines the lambda
function with HTTP event trigger
7) Edit handler.js to add express app
const express = require(express);
const serverless = require('serverless-http);
app.get('/hello world', (req, res) => res.json({
    message: 'Hello World'}));
module.exports.app = serverless(app).
9) Deploy the service: go rules deploy
Deploys the API to AWS, setting up resources
Lambda & API Gateway A URL is generated for
10) Test the deployed API - serverless remo
i) To remove the service Associated with A
ensuring that there are no charges for
services.

Amazon API Gateway is managed service enabling developers to define API or websocket API endpoints & connect them with backend. It handles authentication, access control, monitoring & Tracing of API requests. API gateway integrates with AWS service like Lamda, SNS, IAM & Cognito Identity tools

API Gateway sits between backend services & API users, routing HTTP requests to corresponding backends. It provides to manage API definition & endpoints mapping. It generates API references from definitions & make them available as documentation. API gateway ties together serverless functions & API definitions enabling truly serverless web applications.

a) AWS Lambda: run lambda functions to generate HTTP responses.
b) AWS SNS: Publish notifications when endpoints are accessed
c) Amazon cognito: Provide authentication & authorization
Benefits
1) Simplified API management
2) Enhanced security through authentication & authorization
Benefits
1) Simplified API management
2) Enhanced security through authentication & authorization
3) Improved scalability & reliability

d) Integration with AWS services
e) Reduced administrative burden

Drawback
a Added latency → It can introduce additiona
latency, potentially impacting application
performance
b) Limited fine tuning capabilities → the perfor
parameters can't be customized.

① AWS Lambda → create function with runtime
= Node.js 18x
② Add a trigger → trigger configuration = API gate
In intent → select "RestAPI" & security as "open"
⑤ Name the api. as "dimanshunalle"

Q2 Create your own profile in SonarQube for testing

a) Install SonarQube locally using Docker
docker run -d -name sonarqube -e
Sonar -ES -Bootstrap _checks_ disable = true -p
9000 : 9000 sonarqube : latest

b) Run SonarQube by http://localhost:9000 login with username & Password.

c) Create a new profile via quality profiles sectu
After creating a project, run sonarqube scann
which will analyze it
Sonar scanner
-Dsonar projectkey = <key>
-Dsonar sources = .
-Dsonar host. url = http://localhost 9000
-Dsonar login = <token>

Analyze your code via sonarcloud.

1) Sonarcloud is a cloud based version of sonarqube. After creating your account go to My Projects & select Analyze new project
2) Choose the github repository. the code will be analyzed & results will be displayed on sonar cloud dashboard
3) we can get feedback on code quality, vulnerabilities & code smell directly on sonar cloud interface

Install sonar lint in IntelliJ & analyze Java code

Sonarlint is an IDE Plugin that helps analyze code quality issues directly with

d Analyze a python project
① Ensure that SonarQube is running
② Add SonarPython plugin in Sonarqube in
Manage Jenkins → Configure
③ Configure Sonar-project properties
sonar.projectKey = mypython.project
sonar.sources =
sonar.language = py
sonar.host.url = http://localhost 9000
sonar.login = <your token>
④ At last run your pipeline & build it to che
it console out

e) Analyze a node JS project
a) Build a Pipeline in Jenkins
b) Configure your sonar.project properties
sonar.ProjectKey = my_nodejs - project
. sonar.sources = JS
sonar.host.url = http://localhost:9000
sonar.login = <your_token>

3 In organisations managing repetitive infrastructure requests can strain centralized operations teams, slowing down process. Adopting a self serve infrastructure model using Terraform decentralizes this responsibility empowering product team to manage their own infrastructure

Terraform, a leading infrastructure as a code tool, enables organisation to automate & manage infrastructure using declarative configuration files which reduces manual errors, improves operational efficiency & making it scalable. Central to a self serve model are reusable version controlled terraform modules, which allows to standardize infrastructure deployments. Kubernetes models may include RBAC which can be parameterized to allow customization without sacrificing compilance.

Terraform uses graph based planning algorithm to optimize infrastructure deployments minimizing dependencies & conflicts. Terraform filter resource [illegible] allow for targeted resource management enabling selective deployment & management. Also terraform dependency locking [illegible] ensures consistent provider versions across deployment

Case Study = Airbnb

Airbnb uses Terraform to manage their complex infrastructure across multiple cloud providers include AWS, GCP & Azure

Terraform features used :-
a) Modules :- To standardizing infrastructure configs
b) Work space = Managing different environm
c) State Management :- to track infrastructure changes
d) Pac - fine grined policy managemen
e) Terraform Enterprise = collaboration, security & governance.