Name- Pranav P.T D15A/62

04/05

DOMS Page No. 1
Date / /

Adv. DevOps

Assignment - 2

Q1] Create a REST API with serverless framework.

→ Creating REST API with serverless framework is an efficient way to deploy serverless applications that can scale automatically without managing serverlesss.

(i) Serverless framework : ~~this design~~ A powerfull tool that deployment of servers and serverless applications aeross various cloud providers such as AWS, Azure and Google cloud.

(ii) Serverless architecture : this design model allows developers to build applications without worrying about underlying infrastructure, enabling focus on code and business logic.

(iii) REST API : Representational state transfer is architecture style for designing network application

Steps for creating REST API for serverless framework

(1) Install serverless framework.

You start by installing serverless framework CLI globally using node package manager (npm). this allows you to manage serverless applications directly from your terminal.

(2) Creating a Node.js serverless project

A directory is created for your project, where you will initialize a serverless service (project). This service will house all your lambda functions configurations and cloud resources. Using the command serverless create you set up a template

for AWS Node.js microservices that will eventually deploy to AWS Node.js microservices that will eventually deploy to AWS Lambda.

(3) Project structure:
The project creates essential files like handler.js (which contains code for lambda functions) and serverless.yml.

(4) Create a REST API response
In the serverless.yml file you define function that handles post requests of HTTP.

(5) Deploy the service
With the 'sls deploy' commands serverless framework packages your applications, uploads necessary resources to AWS and set up the ~~sfr~~ infrastruce.

(6) Testing the API : Once deployed you can test REST API using tools like curl or Postman by making post requests to generated API.

Q2] Case study for SonarQube
Creating your own profile in sonarqube for testing project quality. Use Sonarqube to analyze your Github code. Install sonarLint in your Java IntelliJ IDE and analyze java code. Analyze python project with sonarqube.

→ SonarQube is an ~~error~~ open source platform used for continuous insertion of code quality. It detect bugs, code smells and security vulnerabilities in project across various programming languages.

① Profile creation in sonarQube:
Quality profiles in sonarQube are essential configurations that define rules applied during code analysis. Each project has a quality profile for every supported language with default being 'sonar way' profile comes built in for all languages.

② Using SonarCloud to analyze Github code:
SonarCloud is cloud-based counterpart of sonarQube that integrates directly with GitHub, BitBucket, Azure and GitHub repositories. To get started with sonarCloud via GitHub signup via sonarQube Cloud product page and connect your GitHub setup with each project corresponding to GitHub repository.

③ SonarLint in Java IDE:
SonarLint is an IDE that performs on-the-fly code analysis as you write code. It helps developers detect bugs, security vulnerabilities and code smells directly in the development environment such as Intellij Idea or Eclipse. To set it up, install the sonarcloud and select the project profile to analyze Java code.

④ Analyzing Python Projects with sonarQube:
SonarQube supports python test coverage, reporting but it requires third party too like coverage.py

to generate the coverage tool runs before sonar scanner and ensures reports file is saved in different path.

(5) Analyzing Node.js projects with SonarQube:
For nodejs project sonarqube can analyze JavaScript and TypeScript code. Similarly to the python setup, you can configure sonarqube to analyze node.js projects by installing the appropiate plugins and using sonarscanner to scan the ~~pj~~ projects. SonarQube will check the code against Industry standard rules and best practices, flagging issues related to security vulnerabilities bugs and performance optimization.

Q3] At a large organization, your contralized operations team may get repetitive infrastructure requests. You can use Terraform to build a 'self-serve' infrastructure.

→ Terraform's self-serve infrastructure provides a powerful use case in large organisations
(i) Self-serve infra: By using Terraform modules, you can create reusable and standardized infrastructure config. Module creation in Terraform, main.tf, variables.tf and outputs.tf.
Also after module creation its standardization is equally important.
(ii) Enabling self service for product Teams:
Create a self service or version control access.

and provide pre-configured Terraform workflows, onboard and train product teams, and the most important RBAC (Role based Access Control) for preventing unauthorized access.

(iii) Automate Infrastructure Request via ticketing systems:

Integrate Terraform Cloud or Terraform Enterprise, connect terraform with the ticketing system; automate approval workflows and monitor & log requests.

(iv) Workspaces setup for Environment segregation

To ~~merge~~ manage different environments, Terraform workspaces were setup. This ensured that Items could deploy the same infrastructure across different environments without overlap.