Prathamesh Bilve
D15A / 32

AI - DevOps

Assignment - 2

Q1 Create a REST API with servless framework

— creating a REST API with servales framework is an efficient way to deploy servaless applications that can scale automatically without managing server

i) Servaless framework: A powerfull tool that deployment of services & servaless application across various cloud providers such as AWS, azure & google cloud.

ii) servales archetecture: This design model allows developers to build application without worrying about underlying infrastructure enabling focus on code & business logic

iii) REST API: Representational State Transfer application is archeleture style for designing network applications

Steps

1) Install servaless framework using node package manager (npm) globally

2) creating a node.js servales project
a directory is created for project where we will initialize a servales service. This source will have all your lambda functions configuration & cloud resources. using the command services create you set up a template for AWS Node.js microservices that will eventually deployed to AWS lambda

3) project structure

4) creat a REST API resource:
In servelas yml file you define function that handles POST request of HTTP

5) Deploy the service

6) Testing the API
once deployed you can test REST API using tools like curl or POSTMAN

7) Storing data in DynamoDB
To store submitted candidate data you will integrate AWS dynamoDB as a database.

8) Adding more functionalities
adding more functionalities like list all candidates get candidates by ID

9) AWS IAM permissions
you need to ensure that serverless frame work is given right permissions to intract with AWS resources like Dynamo DB

10) Monitoring & Maintenance
After deploying serverless frame work provides service information like deployed endpoints API key , log streams

Q.2 Casa Study for SonarQube

- creating your own prusc in sonarquba for testing project quality use Sonarqube to analyse your github code. Install Sonarlink in you java ide and analyse java code. Analyse python project with SonarQube

→ SonarQube is an open Sourc platform used for continous inspection of code quality. It detects bugs, code smells & security, vulnerabilities in project across various programming language.

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 comes built in for all languages. custum profiles can be created by creating or extending existing ones. Copying creates an independent profile while extending inherit rules from parent profile & reflects future changes respectively. you can activate or deactivate rules priotize rules & configure parameters to tailor profile to respective projects. Permissions to manage quality profiles are restricted to users with administrative privileges

Sonarqube allows for the comparison of two profiles to check for diffrences in activated rules & users can make changes via event log.

Quality profiles can also be imported from other instances via backup & restore To ensure profiles include new rules its important to check against updated built in profiles or use Sonarqube rules page.

To ensure prioritics include new rules its important to check against updated new rules.

2) Using SonarQube to analyse Github code

S1 - use github account to access sonar cloud

S2 - connect your desired github repo to sonar cloud.

S3 - use git account to scan code

S4 - view bugs, vulnerabilities & quality batches in sonar Cloud

3) Install SonarQube in Java IDE

S1 - ~~Sonar~~ Search and install SonarLint from plugin market place

S2 - Detect bugs, code smell, security vulnerabilities

S3 - link to same code or sonar cloud for consistent checking.

S4 - maintain quality standards during development

4) Install SonarQube & enable Sonar python.

S1 - Install SonarQube & enable Sonar python

S2 - execute Sonar Scanner from project root

S3 - Check dashboard for bugs & Security hotspots

S4 - maintain quality in regular scans ensure code maintainability

5) Analyse Nodejs project using SonarQube

S1 - Install sonar-javascript plugin for Nodejs.

S2 - Define project key & source exclusion on sonar project properties

S3 - use Sonar-Scanner to analyse code

S4 - error project standards are must.

8 At a large organisation you centralized 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 Infrastructure
By using terraform modules you can create reusable & standardized infrastructure config module creation in Terraform main.tf variables.tf & outputs.tf
Also other module creation in terraform, main.tf its standardization is equally important.

ii) Enabling self service for product teams:
Create a Self service or version control Access and provide a preconfigured terraform workflows onboard and train product teams and most important RBAC for preventing unauthorized access.

iii) Automate Infrastructure Request via Ticketing systems:
Integrate terraform cloud or enterprise connect terraform with tickiting systems automate approval workflows & monitor & log requests

iv) workspace setups for environment segregation.
To manage different Environment, Terraform workspaces was setup This ensured that terraform cloud deploy the same infrastructure across different environments without overlap.