

Adv Devops Assignment 2

- Q Create a REST API with serverless framework
- creating REST API with serverless framework is an efficient way to deploy serverless framework is an efficient applications that can scale automatically without managing server. ~~serverless framework~~ :- A powerful tool that deployment of services and serverless applications across various cloud providers such as AWS, And Google cloud
- ii) serverless architecture :- This design model allows developers to build applications without worrying about underlying infrastructure enabling focus on code & business logic
 - iii) REST API :- Representational state transfer is architecture style for designing network applications

Steps for creating REST API for serverless framework

- i) Install serverless framework
you start by installing serverless framework globally using node package manager (npm). This allow you to manage serverless apps directly from your terminal
- ii) 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 commands `serverless create` you set up a template for AWS Node is `microservices` that will eventually deploy to AWS lambda

iii) project structure :-

The project scaffold creates essential files like `handler.js` which contains code for lambda function and `serverless.yml`.

iv) create a REST API Resource

In the `serverless.yml` file you define function that handles post request of HTTP

v) Deploy the service

With the `sls deploy` commands `serverless` framework packages your applications, uploads necessary resources AWS and set up the infrastructure

vi) Testing the API :- once deployed you can test REST API using tools like `curl` or `postman` by making post request to generated API

vii) Storing Data in ~~Database~~ `DynamoDB` :- To store submitted candidate data you integrate AWS `DynamoDB` as database

viii) Adding ~~name~~ functionality:- Adding functional like install candidates get candidates by ID

ix) AWS IAM permissions
you need to ensure that serverless framework is given right permissions to interact with AWS resources like DynamoDB

x) Monitoring and maintenance
After deployment serverless framework provides services information like Deployed endpoints API keys, log streams

Q2) case study for SonarQube
creating your own profile in ~~sonarqube~~ for testing project quality Use sonarqube to analyze your code Install java ~~file~~ ide and analyze java code.

→ sonarqube is an open source platform used for continuous inspection of quality. It detects bug code smells and security vulnerabilities in project across programming languages

i) profile creation in sonarqube
Quality profiles in sonarqube are essential configuration that define rules applied during code analysis. Each project has a quality profile for every supported lang with default being sonar way profile comes built

in for all languages custom profile can be created by copying or extending existing ones copying creates you can activate or deactivate rules prioritize certain rules and configure parameters to profile to specific projects.

- 2) using sonarQube to analyse github code
 sonarQube is a cloud based counter part of sonarQube that triggers directly with github Bit Bucket, and github repositories To get started with sonar cloud via github signup product page and connect your github organization or personal account once connect sonar cloud mirrors your github setup with each project corresponding to the github repos:- After setting up organization where each github repos a sonar cloud project. Define new cloud to focus on recent changes and choose between automatic analysis or CI based analysis Automatic analysis happens directly in sonar cloud while CI based analysis integrates with your build process once the analysis results can be viewed in both sonar cloud and github including security import issue.

3) Sonarlint in Java IDE:-

Sonarlint is an IDE that performs on-the-fly code analysis as you write code. It helps develop in the developing environment such as IntelliJ IDEA or Eclipse. To set it up, install the Sonarlint plugin, configure the connection with SonarQube or SonarCloud, and select the project profile to analyse Java code in code quality, promoting clean & maintainable code from beginning.

4) Analyzing Python projects with SonarQube

SonarQube supports Python test coverage reporting but it requires third party tool like coverage to enable and adjust your build process so that coverage tool runs before SonarScanner and ensures report file is saved in diff path. For setup you can use `tox` and `coverage to py` to configure and run test. In your `tox.ini`: include configuration for py test and coverage to generate report in XML format. The build process can also be automated using GitHub Actions which install dependencies, run test and involves SonarQube scan. Ensure report in XML format and place where Scanner can access it.

5) Analyzing Node.js projects with SonarQube

For node.js project SonarQube can analyze Java script and typescript code. similar to the python setup you can configure SonarQube to analyze node.js project by installing the appropriate plugin and using SonarScanner to scan the 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 centralized operation team may get many repeatable infrastructure request you can use form to build a self services infrastructure mode that lets product team manage their own infrastructure independently you can create and use Terraform modules that codify the standards for deploying & managing services in your organization allowing teams to efficiently deploy services.

→ Implementing a self service infrastructure model using Terraform can transform how large organization manage their infrastructure independently organization can enhance efficiency

reduce benefits and ensure compliance with established needs.

The need for self service infrastructure:- In large organization centralized operators teams often face an overwhelming number repetitive request. This can lead to delay in service delivery and frustration among product teams who need to move quickly. A self service model allows team to provision and manage their infrastructure without relying on the operations team for every request.

• Benefits of using Terraform:-

1. Modularity & reusability
2. Standardization
3. Increased Efficiency
4. Integration with ticketing systems

• Implementation steps

i) Identify Infrastructure components.

• Begin by Identifying which components your infrastructure can be modularized like VPCs, security groups, load balanced.

2) Develop Terraform modules

- create reusable modules that define the desired configuration & resources
- Ensure each module includes input variables for customization and outputs for integration with other modules.

3) Establish Governance and Best practices

- Define guidelines for module usage version and documentation to ensure clarity and maintainability
- Encourage teams to contribute to module development and share improvements.

4) Testing and Validation

- Implement a testing framework to validate functionality before development.
- Best practices for module management
 - utilize the terraform registry
 - leverage existing community modules from the Terraform Registry to avoid reinventing solutions and ensure adherence to best practices.