04/05

Advance devops (assignment no 2)

Create a REST API with serverless framework

Creating REST API with serverless framework efficient way to deploy serverless application that can scale automatically without managing servers. A powerful tool that deployment of services and serverless application across various cloud periods such as AWS, and google cloud.

ii) Serverless architecture: This design model allows developers to build application without waring about underlying infrastructure enabling focus on code & bussiness logic.

iii) REST API: Representational state transfer is architecture style for designing network application directly from your terminal.

Steps for creating RESTART for serverless framework

Install Serverless framework:

You start by installing Serverless framework CLI globally using node modules. This allows you to manage serverless application directly from your terminal

Creating a node js serverless project:

A directory is created for your project, where you will initialize a serverless service. This service will house all your lambda functions configuration and could services. Using the command serverless create your AWS

Node js microservices that will eventually deploy to AWS lamda.

3) Project structure:-
The project scoldfold creates essential files like handler.js (which contains code for lambda function and serverless.YML.

4) Create a REST API Resource:
In the serverless yml file you define function that handles port request of HTTP.

5) Deploy the service:
with the SLS deploy commands: serverless framework packages your application, uploads necessary resource to AWS and set up the infrastructure

6) Testing the API: once deployed you can test REST API using tools like Postman by making POST request to generated API.

7) Storing Data in Dynamods: To store submitted candidate data you integrate AWS DynamoDB as database

8) Adding more functionality: Adding functionalities like install candidates get candidates by ID.

9) AWS IAM permissions
you need to ensure that serverless framework is given right permissions to interact with AWS resources like DynomoDB.

» Monitoring and maintainance.
fter deployment Serverless framework provides servic
formation like deployed endpoints, API Key, log
reams.

Case study for sonarqube.

reating your own profile in Sonarqube for testing
roject quality. Use sonarqube to analyze your code.
Install java ide and analyze java code

Sonarqube is an open source platform used for
continous inspection of quality. It detects bug, code
smells and security vulnerabilities in project acro
programming languages.

» Profile Creation in Sonarqube
Quality profiles in sonarqube are essentials
configuration 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
Custom profiles can be created by copying or
extending existing ones copying creates you can
activate or deactivate rules, prioritize certain
rules and configure parameter to profile to
specific projects

2) Using Sonarcloud to analyse github code:
Sonarcloud is a cloud based counter part of sonarqube that triggers directly with github, Bit Bucket, and github repositories. To get started with sonarcloud via github signup product page and connect your github organization or personal account. Once connect, sonarcloud mirrors your gith setup with each projet corresponding to the github repos:- After setting up organization choose subscription plan. Next input repositories into your sonarcloud organization where each github repo to a sonarcloud project. Define new cloud to focus on recent changes and choose between automatic analysis or CI based analysis. Automatic analysis happens directly in sonarcloud. 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 developers in the developing environment such as Intellic 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 begining.

Analyzing python projects with Sonarqube.

Sonarqube supports Python test coverage reporting but it requires third party tool like coverage you need to enable and adjust your build process so that coverage tools runs before Sonnar scanner and ensures report file is saved in diff path. for set up, 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 runs test and invokes Sonarqube scan. Ensure report in XML format and place where scanner can access it.

Analyzing Node.js projects with SonarQube.

For nodejs project Sonarqube can analyse Java script and typescript code. Similar to the python setup you can configure Sonarqube to analyze nodejs project by installing the appropriate plugin and using sonar scanner to scan the projects. Sonarqube will check the code against industry standard rules and best practices. flagging issues related to securing vulnerabilities bugs and performances optimization

3) At a large organization your centralized operation team may get many repatabile Infrastru requests you can use form to build a self service infrastructure modes that lets product team manage their own infrastructure independently you can create and use Terraform modules that codify the standards for deploying & managing servers in your organization, allowing teams to efficiently deploy services

Ans Implementing a self service infrastructure model using Terraform can transform how large organizati manage their infrastrudure independently organizatio can enhance efficiency reduce benefits and ensure compliance with established needs.

- The need for self service infrastructure:

In large Organization, centralized operations teams often face an overwhelming number of repetitude requests. This can lead to deploy the service delivering and frustration among the product teams who need to move quality. A self service models allows teams to provision and manage their infracstructure without replying on the operations team for every request.

- Benefits of wing terraform.

1. Modularity & Reusability:

Terraform modules encapsulate standard configuration for various infrastructure components. Teams can reuse these modules across different projects reducing redundancy and minimizing the risks of errors.

Standarizations
By identifying and practices within modules organization can ensure that all deployment comply with internal policies and standards.
This conticiency helps maintain security and operational integrity accross the organization.

Increased Efficiency:
Product teams can deploy services quality by using predefined modules, significantly reducing the time spent on infrastructure setup.
This allows them to focus on developing features rather than mangging infrastructure.

Integration with ticketing systems.

Terraform cloud can integrate with ticketing system like service now. to automate the generation of infrastructure requests.
This integration streandines workflow by allowing teams to initiate request directly from their ticketing platform, reducing manual intervention

Implementation steps:

1. Identify infrastructure components
Begin by identifying which components of your infrastructure can be modularized.

2. Develop Terraform modules.
- Create reusable modules that define the desired configuration and 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, versions and documentation to ensure clarity and maintainability.
- Encourage teams to contribute to module develop and share improvements

4. Testing and validation

- Implement a testing framework to validate module functionality before development.
- Best practices for module management
- Utilize the terraform register.
- leverage existing community modules from the Terraform register.