Adv Devops Assignment 2

Q1. Create REST API with Serverless Framework.

→ Creating REST API with Serverless Framework is an efficient way to deploy serverless application that can scale automatically without managing servers

(i) Serverless Framework: A powerful tool that deploy of services & serverless application across various cloud providers such as AWS, amazon & google cloud.

(ii) Serverless Architecture: This design model allows developers to build application without everrying about underlying infrastructures, enabling focus on code & busniess logic.

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

Steps for creating REST API for Serverless Framework.

1) Install Serverless Framework:
You start by installing serverless framework CLI globally using node package manging (npm). This allow you to manage Serverless application directly from your terminal.

2) Creating a Node.js Serverless project:
A directy is creating for your project, where you will initialize a Serverless service (project) This service will house all your lambda Fun^n Configuration & cloud resources using the command Serverless Create you set up a template

3) For AWS Node.js microservices that will eventually deploy to AWS lambda.

3) Project Structure: The project scaffold creates essential files like handler.js code for lambda functional & structral .yml.

4) Create a ~~React~~ REST API Resource:
In the serverless.yml file you define function th handle post requests of HTTPS.

5) Deploy the service:
With the 'Sls deploy' command serverless frame work packages your application, uploads neccar resources to AWS & set up the infrastructure.

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

7) Storing data in DynamoDB: To store submitted Candidate data you integrate AWS Dynamo D as database.

8) AWS IAM Permission:
you need to ensure that serverless framework is given right permission to interact endpoint API Key, log Strecross.

9) Monitoring & Maintainance.
After deploy Serverless framework provides Servic information like deployed endpoints, API Key log streams.

Case Study for SonarQube.

Creating your own profile in Sonarqube for testing project quality. Use Sonarqube to analyze your github code. Install Sonarent in your Java intellij ide & analyze java code. Analyze python project with Sonarqube.

→ Sonarqube is an open source platform used for continous inspection of code quality. It detects bugs, codes, small & security vulerabilities in Project accross various programming languages.

1) Profile Creation in SONARQube:

Quality profiles in Sonarqube are essential configuration that defines 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 language. Custom profile can be created by coping or extending existing ones. Copying created an independent profile, while extending inherit rule from parent profile & reflect future change automatically. you activate or deactive rule, priotize certain rules & Configure parameter to tailor rules, priotize certain rules & configure Profile to specific projects. Permission to manage quality profile are restricted to user with administrative prevalagies.

2) Using Sonarcloud to analyse code
Sonar cloud is a cloud-based counterpart of Sonar Qube that integrates directly with git-hub, Bitbuc Azure & github lab repositories. To get started with Sonarcloud via github signup via sonarcl product page & connect your github org. or pers acc. once connect, Sonarcloud mirror your github setup with each project corresponding to github repo. After setting up org. choose subs. Coiption plan(Free for public repo). Next, imp repositories into your Sonarcloud, Organization wi each github repo become a Sonar cloud projec Define 'new code' focus on recent changes & choose betn automatic analysis or CI based analysis.

3) Sonarlint in Java IDE:
Sonarlint is an IDE that perform on the fly code analysis as your write code. It ho developers detect bugs, security vulerabilities & code smells directly in development enviro such as Intelli Idea or Eclipse. To set it up install the Sonarlist plugin, configure the c ction with SonarQube or Sonarcloud & sele the project profile to analyze Java code.

4) Analyzing Python project with SonarQube.
SonarQube Support python test coverage reporting but it requires third party too like Coverage. port. To enable coverage.py to generate coverage blind process So that coverage tool runs before Sonar Scanner & ensure report file is Saved in diff path.
For Setup, you can use Tox, Python & coverage. to configure & run test. In your Tox.ini include configurations for pytest & coverage to generate Coverage test report in XML format. Ensure report in Cobeurata XML format & Place where Scanner can access it.

5) Analyzing Node.js projects with SonarQube.
For Node.js project Sonarqube can analyze JS & TS code. Similar to python setup you can configure SonarQube to analyze node.js. Project by installing appropriate plugin & using SonarScanner to scan project. Sonarqube will check code the against industry standard rules & best practice, flogging issue related to security vulerabilities bugs & performance optimization.

At a large scale org., your centralized operation team may get many repetetive, infrastructure req you can use terraform to build "self server" infra independently you can create & use terraform modules that codify standard for deploying & managing. services.
Implementing a "self service" infra. model using terraform how large org. manage their infra. inde dependently.

Need for self service infra.
In large org., centralized operation team often fall an overwhelming no. of repeted requests. This can lead to delay in ser delivery & frustration among product team who need to move quickly

Benefits of using terraform:
1) Modularity & Reusability:
- Terraform modules encapsulate standa configuration for various component.
- Team can reuse these modules accro diff projects, reducing the risk of err

2) Standardization:
By defening best practices with module organization can ensure all deployme comply with internal policies & t standa

3) Increased Efficiency:
Product team can deploy service by using pre-defined modules, significantly reducing time spent on infrastructure ~~reducing~~ setup.

→ Implementation on steps.
1) Identify infrastructure components:
Begin by identifying compnent of your infra. can be modularized.
2) Develop Terraform modules:
- Create reusable modules that define desired configuration & resources.
3) Establish governance & Best practices:
Define guidelines for module usage versioning & documentation to ensure clarity & maintainability.
4) Testing & validation
- Implement a testing framework to validate module functionality before deployment.

* Best practice for module management:
- utilize the terraform register:
- version control: Implement versioning for your modules to track changes overtime

This approach may not streamline process. but also enhances agility responding to

changing business needs.

Q3 Terraform "self service" infrastructure modle.

→ Terraform module for self-serve infra.
· Create terraform modules that codify standard for deploying common resours like VPC's, EC2 instances & S3 bucket-

Ex:

```
variable "instance-type" {
        default = "t2.micro"
}
resource "aws-instance" "example" {
    ami = "ami-12345678"
    instance_type = var.instance-type.
    tags = {
        Name: "example-instance"
    }
}
```

EC2-module/outputs.tf:

```
output "instance_id" {
    value = aws_instance.example.id.
}
```

Team can now use the module to deplo
EC2 instance with

```
module "ec2" {
    source = "./ec2-module"
    instance_type = "t2.medium"
}
```

Terraform can now use this module to deploy EC2 instance with.

- you can integrate terraform cloud with Service Now to automate the infrastructure request process.

Example workflow:

1) Product team submit a request in Service now for new infrastructure.
2) the req. trigger a Terraform cloud update the service now ticket with Status & resource details.
3) Creating Terraform for company team define reusable module for commonly requested resources like reusable module for commonly requested resources like
   1) Networking
   2) Compute EC2
   3) Storage
   4) IAM Role/Policy

By doing this, teams can manage their own infrastructure while maintaining compliance with organisational standard.