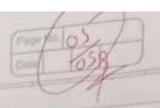
Assignment -2



- (S) Create a REST API with serverless framework.

 - ⇒ set up the environment · Install Node is AWS CUI and serverless framework.
 - · Configure AWS credentials using aws configure.

 - · Create a new directory for the project

 · Use the serverless CLI to generate a new service

 with serverless create template aws node;s

 - · Configure the serverless yml file

 · Define the service , provider (AWS) and functions

 [API endpoints] in serverless yml.
 - Set up the HTTP methods [GET, POST, PUT, DELETE]
 for your REST API paths.

 - · With the APT togic: • In handler is implement functions for handling requests [e.g. fetching, creating, updating of deleting data].
- · Handle different HTTP methods of their corresponding o peploy the service :
- . Deploy the service to AWS using serverless deploy. deployment output:

terrained at pathautes strong strong dashasashat au

- . TEST the APINT TO THE STATE OF THE STATE O
- · Use tools like Postman to test the endpoint [e.g sending GET , POST, PUT and DELETE requests] 20 3 July 19 July spanned to saling the desired

Incelam andaud 3

· Remove the service

· After testing remove the service from AWS using serverless remove to clean up resource.

C- Insquaples A

These steps will guide you through the process of creating, deploying of managing a serverless REST API

32> case study for sonarqube

-ect quality. Use sonarqube to analyze your github cade.

Install sonarlint in your sonarqube Java Intellij ide

and analyze java code. Analyze python project with

sonarqube

> Sonarqube is an open source platform used for conti-nuous inspection of code quality. It detects bugs, code
smalls of security vulnerabilities in project across various
programming languages

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 language with
default being so narway profile comes built in for all
language custom profiles can be created by copying
or extending existing ones. Copying creates an independent
profile, while extending existing ones. Copying creates
an independent profile, while extending in herit rules
from parent profile of reflects future changes automatically
You can activate or deactivate rules, prioritize certain
rules and configure parameter to tailor profile to specific
projects. Permissions to manage quality profile are

restricted to users with administrative privileges. Sonarqube allows for the comparison of two profiles to check for differences in activated rules and users can track changes via event 10g. Quality profiles can also be imported. From other instances via backup of restore to ensure profiles include new rules its important to check against updated built in profiles or use sonarqube rules page.

Using sonarcloud to analyze Github code!

Sonarcloud is cloud-based counterport of sonarqube that integrates directly with Github. BitBucket, Azure and Gitlab repositories. To get started with sonarqube via Github signup via sonarcloud product page of connect your Github organization or personal product page and connect your Github repository. After setting up the organization choose subscribetion plan [free for public repo]. Next, import respositories into your conarcloud organization where each Github repo becomes a sonar (loud project. Define 'new code' to focus on recent changes of choosen bet automatic analysis or (I-based analysis automatic analysis happens directly in sonarcloud, while (I based analysis integrates with your build process once the analysis is complete results can be viewed in both sonar-cloud and Github including security import issue.

Sonarlint in Java IDE:

Sonarlint is an IDE that performs on the fly code

analysis as you write code. It helps developées detect

bugs, security vulnerabilities 4 code smells directly

Prontesi.

in the development environment such as Intellig Idea or Eclipse. To set it up, install the sonarlint plugin, configure the connection with sonarqube or Sonarchoud select the project profile to analyze Java Code. This approach ensures immediates feedback on code quality promoting clean of maintainable code from beginning.

Analyzing python projects with sonarqube:
Sonarqube supports python test coverage, reporting but it requires third party to like coverage py to generate the coverage tool runs before sonar scanner gensuses report file is saved in different path.

- Analyzing Node is project with sonarqube

 For node is project sonarqube can analyze javoscript

 4 typescript code Similar to the python setup, you can
 configure sonarqube to analyze node is projects by
 installing the appropriate plugins & using sonarscannes
 to scan the projects sonarqube will check the code
 against Industry standard rules, best practices,
 therefore issues related to security vulnerabilities flagging issues related to security vulnerabilities bugs of performance optimization.
- 3. At a large organization, your centralized operations team may get many repetitive infrastructure requests, you can use terratorm to build a "self-serve" infrastructure independently. You can create guse terraform modules that codify the standards for deploying 4 managing services in compliance with your organisation's practices. · Terraform cloud can also integrate with ticketing system like environment service now to automatically generate new infrastructure requests.

Implementing a 'self-service' infrastructure model using Terraform can transform how large organizations manage their infrastructure independently, organization can enhance efficiently, reduce bottlenecks of ensure compliance with established needs. with established needs.

· The need for self-service infrastructure:

In large organizations, centralized operations team often fall an overwhelming number of repetitive requests. This can lead to delays in service delivery of frustration among product teams who need to move quickly. A self-service model allows teams to provision of manage their intrastructure without relying on the operations team for every request.

· Benifits of using Terroforms

Nodularity & Reusability:
Terraform modules encopsulate standard configurations
for various components

· Teams can reuse these modules across different projects, reducing redundancy of minimizing the risk of errors.

2> Standardizations:

· By defining best practices with modules, organisation can exist that all deployments comply with iternal policies & standard.

integrity across the organization.

3) Increased Efficiency:
Product teams can deploy services quickly by siusing pre-defined modules, significantly reducing the time spent on infrastructure setup.

4) Integration with Ticketing systems.

• Terraform cloud can integrate with smooth ticketing system like service now to automate the generation of intrastructure requests

·This integration streamlines work blows by allowing teams to initiate requests directly from their ticketing platform, reducing manual intervention.

=> Implementation 8teps:

DIdentify infrastructure components?

Begin by identifying with components of your infras--tructure can be modularized.

2) Develop Terraform moduls:

o Create reusable modules that define the desired configurations & resources.

3) Establish Governance & Best Practices:

· Define quidelines for module usuage, versioning I documentation to ensure darity & mountainbility. 14) Testing & validation:

Functionality before deployment

* Best practices for module: management:

· Utilize the terraform register.

· Version control: Implement versioning for your modules to track changes overtime. This helps manage dependencies effectively of minimize distruption during updates.

This approach not many streamline processes but also enhances agility in responding to changing business environment that supports innovation of growth within

Terraform can now use their module to deploy Ecz instances with

"You can integrate terraform cloud with service Now to automate the infrastructure request process.

"Using terraform's APT driven approach, service Now can trigger Terraform runs based on ticket approvals, automating resource deployment.

Example workflow:

DA product team submits a request in service Now for a

2) The request triggers a Terraform (loud updates the service Now ticket with the status, of resource details.

3) Creating Terraform modules for stat teams define reusable modules for commonly requested resource like

> Networking [VPC, Bubnets]

2> Compute EEC2, Autoscaling Groups

3) Storage (SS, RPS]

4) IAM Roles/ Policies

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

- Contract