Siddhant Sathe
D15A/51

Assignment 2

1) Create a REST API with serverless framework.
Serverless framework is a powerful tool that deployment of sources and serverless applications across various cloud providers.
REST API stands for representational state transfer is architecture style for designing network applications

Steps
S-1) Install serverless framework
Install serverless framework CLI globally using npm.
This allows you to manage serverless applications directly from your terminal.

S-2) Creating a node.js serverless project.
This project will house all your lambda functions, configurations, and cloud services.

S-3) Project Structure
The project creates essential files like handler.js (contains code for lambda functions) and serverless.yml

S-4) Create a REST API Resource
In the serverless.yml file define function that handles post requests of HTML

S-5) Deploy service
sls deploy commands serverless packages your application uploads necessary resources to AWS and set up the infrastructure.

Teacher's Sign.:

S-6) Testing API
Once deployed, test REST API using tools like curl or postman by making request to generated API.

S-7) Storing data in DynamoDB
To store submitted data, integrate dynamoDB as a databa

S-8) Adding AWS IAM permissions
Ensure that serverless framework is given right permissions.

S-9) Monitoring and maintainace
After deployment serverless frameworks provides services like deployed endpoints, API key, log streams

2) Case Study for Sonarqube
i) Create your own profile in sonarqube for testing project quality.
ii) Use sonarcloud to analyze your github code
iii) Install sonarlint in your Java intellij IDE or eclipse IDE and analyze your Java code
iv) Analyze python project with sonarqube
v) Analyze nodejs project with sonarqube

→ S-1) Create the sonarqube profile for testing project quality.
S-2) Open Intelli-j setting, find Tools > Sonarlint - entry and select '+' to open connection wizard.
S-3) Enter a name for this connection, select sonarcloud or sonarqube.

S-4) Choose the authentication method
a) Generate token on sonarqube or sonarcloud
b) Username + Password
S-5) For sonarcloud only select organisation that you want to connect.
S-6) Sonarqube and sonarcloud can push notification to developers
S-7) Validate the connection creating by selecting finishing at the end of the wizard
S-8) Save the connection in global setting by clicking ok
S-9) Bind python project to sonarqube
a) Select sonarlint > Bind project to sonarqube
b) Choose the correct project from sonarqube
S-10) ~~Cho~~ Analyze the project (python project)
a) Trigger an analysis by going to code > analyze code > sonarlint.
S-11) Analyze project (nodejs)
Make sure your nodejs project is properly configured with sonar-project.properties file for analysis to run.

Q23) At large organisation your ~~cred~~ centralized operation team may get many repititive infrastructure requests. You can use Terraform to build a 'self-serve' infrastructure model that let's product teams manage their own infrastructure independently. You can create and use Terraform modules that codify the standards for deploying and managing services in your organization, allowing teams to effectively deploy services in compliance with your organization's practices. Terraform Cloud can also integration with ticketing systems like servicenow to automatically generate new infrastructure request.

→ Self serve infrastructure model with terraform MODULES:

At a large organization, implementation a self serve infrastructure model using terraform can significantly streamline the process of managing infrastructure across diffrent teams.
This approach allows product teams to manage their own infrastructure independently while adhering to organizational standards and best practices.
Key aspects of this self serve model include
a) Standardization through terraform Modules.
By creating and utilizing terraform modules organizations can codify their infrastructure deployment and arrangement standards. These modules serve as reusable packages of terraform configurations that encapsulate common pattern.

b) Compliance
By using predefined modules, teams, ensure that their deployments comply with the organisation's established practices and security guidelines

d) Automation
The use of terraform modules promotes automation, reducing manual intervention and potential human errors in infrastructure management

~~e)~~

e) Version Control
With modules stored in version control system, teams can track changes, collaborate on improvements and maintain a history of infrastructure configurations

INTEGRATION WITH TICKETING SYSTEMS
Terraform cloud offer integration capabilitees that further ~~eta~~ enhance the self serve model

a) Automatic Infrastructure Request
Terraform cloud can integrate with ticketing system like ServiceNow to automatically generate new infrastructure requests. This automation streamlines the process of submitting and tracking infrastructure changes.

b) Centralized Management
By centralizing infrastructure management through terraform cloud organisation can maintain better clard over who can request and approve infrastrure changes