Assignment - 2

1. Create a REST API with the serverless framework.

→ Install Node.js and NPM from node.js website

steps:

1. Install serverless framework

The serverless framework is a CLI tool that helps deploy serverless applications, managing resources like Lambda, API Gateway and more.

Command: npm install -g serverless

2. Create a New Serverless Project.

The serverless Framework provides templates to scaffold a new serverless service. This will generate the folder structure and basic configuration

Command: serverless create --template aws-nodejs --path my-rest-api

cd my-rest-api

3. Configure AWS Credentials

The serverless framework requires AWS credentials to deploy your functions and resources (Lambda, API Gateway etc.) on AWS

Command: AWS configure

4. Define API routes and resources in serverless.yml

The serverless.yml file defines your entire application functions, events (such as HTTP end points), and resources. It's the core configuration file for the serverless framework.

command : Modify serverless.yml as follows

```
service : my-rest-api
provider:
  name : aws
  runtime : nodejs 14.x
  region : us-east-1
  Stage : dev
functions:
  createItem:
    handler : handler.createItem
    events :
      -http :
        path : item
        method : post
  getItem :
    handler : handler.getItem
    events :
      -http :
        path : item/{id}
        method : get
  updateItem:
    handler : handler.updateItem
    events :
      -http :
        path : item/{id}
        method : post
  deleteItem:
    handler : handler.deleteItem
    events :
```

```
- http:
    path: item/{id}
    method: delete
```

Write Lambda function in handler.js

Lambda function handle the logic for each API route. They get triggered when a request is made to the API gateway.

command: In handler.js, write functions like the following

```
'use strict';
module.exports.createItem = async (event) => {
  const body = JSON.parse(event.body);
  return {
    statusCode: 200,
    body: JSON.stringify({
      message: "Item created successfully!",
      item: body
    }),
  };
};
module.exports.getItem = async(event) => {
  const id = event.pathParameters.id;
  return {
    statusCode: 200,
    body: JSON.stringify({
      message: "Item retrieved successfully!",
      item Id: id
    });
  };
};
```

6. Deploy the REST API
once your configuration and code are setup, deploy the service to AWS :
command : serverless deploy

7. Test the API
You can use tools like Postman or curl to REST API
command : curl -X POST https://<your-api-url>/dev/
-d '{"name": "Book"}'
curl https://<your-api-url>/dev/item/

8. Monitor and Logs
Its important to monitor your lambda functions for debug and performance analysis.
command: serverless logs -f createItem -t

Q2. Case Study for Sonarqube
- Create your own profile in sonarqube for testing project quality.
- Use sonarcloud to analyse your Github code
- Install sonarlint in your Java intellij ide or eclipse ide and analyze your java code
- Analyze python project with sonarqube
- Analyze node js project with sonarqube

→ Sonarqube helps developers identify issues like bugs, security vulnerabilities, and code smells.

1. The first step is creating personal profile, which allows you to manage and monitor code quality across different projects.

- Sign in to sonarqube and set up project preferences
- connect your project to the profile for analyse
SonarCloud integrates with Github, offering continuous analysis for code hosted in repositories.
- Link your Github repository to SonarCloud after signing up.
- SonarCloud automatically analyses code, providing insights into bugs, security vulnerabilities and maintainability.
SonarLint is a plugin that offers real time code quality checks while coding in an IDE.
- In IntelliJ, install sonarlint from settings > plugins
in Eclipse, use Help > Eclipse Marketplace
- Configure sonarlint to sync with sonarQube or sonar cloud.
- SonarQube supports Python projects. To analyse:
- Setup the Python project with a sonar project properties file.
- Run the sonarScanner to analyze the code:
sonar-scanner
- View the detailed report on SonarQube's dashboard.
Analysis for Node.js
- create a sonar-project-properties file in the project root.
- use sonarscanner for analysis
- Review the report on sonarqube, focusing on code security and optimization.

At a large organization your centralized operations team may get many repetitive infrastructure requests. You can use Terraform to build a "self-serve" infrastructure model that lets 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 team to efficiently deploy services in compliance with your organizations' practices. Terraform cloud can also integrate with ticketing systems like service now to automatically generate new infrastructure requests.

Self-serve Infrastructure with terraform

- By using terraform, you can build reusable modules that encapsulate the best practices and compliance standard of your organization. These modules can be made available to various product teams, empowering them to provision infrastructure themselves.
- This reduces the bottleneck that a centralized operations team may face, as the product teams no longer need to wait for manual approvals or setup. Instead, they can handle their own infrastructure needs within the boundaries set by the organization.

Terraform Modules for standardization.

- Terraform modules are reusable templates that simplify the process of deploying infrastructure by codifying the standards for managing resources. Product teams can leverage these modules to deploy resources that comply with organizational policies ensuring consistency and compliance

→ Integration with Ticketing Systems like service Now
- Terraform cloud or Enterprise can integrate with tools ServiceNow, automating the process of generating new infrastructure requests.
- For instance, when a product team submits a ticket f infrastructure resources, Terraform Cloud can autom handle provisioning based on pre-approved templates, re the need for manual intervention from the operations team
- This model optimizes operational efficiency by de infrastructure standards, and automating the requ process through integration.