Assignment - 2

Create REST API Serverless framework.
Install the serverless Framcwork
Make sure you have node.js installed, then install serverless globally

```
npm install -g serverless
```

Create a Service
Create a new Serveless service (project) with the following Command.

```
Serverless create --template aws - node.js --path
rest-api-serverless
cd rest-api-serverlees
```

Define API Endpoints in serverless.yml

```
Service: rest-api-serverless
Provider:
    name: aws
    runtime: node 14.x
    region: us-east-1
functions:
  Create User:
      handler: handev.create User
      events:
           - http:
               Path: users
               method: post
  delete User:
      handler: handler. deleteuser
```

4. Implement handler in handler.js
Open handler.js file and add logic for handli
request.

5. Deploy the service
Deploy the REST API to AWS Lambda using following command.

```
serverless deploy
```

6. Test API
Use tools like postman or CURL to test APS:

For Post man or thunderclient you have to provide a request body in json format and the URL which you are testing.

Q.2. Case study for Sonar Qube
Sonarqube is a popular Static code analysis tool that helps in detecting bugs code smells and security vulnerablities.

1. Create your Profile in SonarQube for testing Project Quality
Download and install SonarQube
Create new account
Navigate to create New Projects and generate a unique token for project

Use Sonar Cloud to analyze your Github code:

steps:

1 Sign up at Sonar Cloud and link it to your Github account.

Import a project from github into SonarCloud

Configure Sonar Cloud with your projects quality.

Use Sonar cloud Github action in your CI pipeline or run Sonar Scanner locally to analyze project.

Install Sonar Lint in your java Intelji IDE

Install SonarLint in your IntelliJ or Eclipse from plugin marketplace.

Configure SonarLint to bind with SonarQube or SonarCloud instance.

Write or open a Java project in your IDE.

Review the suggestions provided by Solar Lint and refactor code accordingly.

Analyze Python project with Sonar Qube

steps:

Install Sonar Scanner or use a CI pipeline to integrate SonarQube analysis into your python project

Create a python project in SonarQube or Sonar-cloud.

Configure the sonar-project.properties file for your Python project.

5. Analyze Node.js project with Sonar Qube
steps:

1. Install Sonar Scanner or integrate Sonar Qube into your CI pipline for Node.js projects.
2. Create a sonar project .properties file for your Node.js project:
3: Run Sonar Scanner to analyze your Node.js project:
sonar-scanner
4. Review the results in Sonar Qube to identify any code quality issues like code smells, vulnerablities issues in your Node.js code.

3. Terraform and Self-Serve Infrastructure Model f
Large Organizations

1. Using Terraform for self-serve Infrastructure.
In large organizations, the operations team of gets repetitive infrastructure requests. A solu to streamline this is by using Teraform to build a self-aware infrastructure model. with Terraform, product teams can independently ma their infrastructure with original standards.

Benefits:

> Allow decentralized teams to deploy services efficiently.
> Ensures compliance with best practices and standards through reusable Terraform moduels.
> Reduces the workload on the central operation
> Improves scalablity and flexiblity in manging inf

Terraform modules: By Creating reusable Terraform modules, you can encapsulate best practices for deploying services like database , VMs, or containers making it eaiser to manage resources.

Terraform Cloud and Ticketing System Integration

Terraform Cloud integrates with ticketing system like ServiceNow to automate infrastructure requests.

ServiceNow - Terraform Integration: when a new infrastructure request is generated in ServiceNow Terraform Cloud can automatically create and provision the requested infrastructure using predefined terraforms modules.