Name
Roll no - 54

A+ 2 05/05

Assignment - 2

Create a REST API with the serverless framework.

Step 1 : Install serverless framework
make sure you have node.js installed. Then install the serverless framework globally.

```
npm install -g serverless
```

Step 2 : Create a new service.
Navigate to the directory where you want to create your project and create a new service:

```
serveless create --template aws-nodejs --path
    my-service
```

This creates a serverless.yml configuration file.

Step 3 : Define REST API in serverless.yml
In your serverless.yml, define the API endpoints and methods.

Step 4 : Implement API in handler.js.
In the handler.js file, write the logic for the API:

Step 5 : Deploy the service
To deploy your API to AWS Lambda:

```
serverless deploy
```

Teacher's Sign.: ______________

Step 6 : Test the API

After deployment ,serverless will give yo
API endpoints. Use tools like postman
curl to test the endpoints:

```
curl https://your-api-endpoint.amazonaws.c
        /dev/users
```

This will return the appropicte responce based on the method and path.

Step 7 : Cleanup

If you want to remove the API

```
serverless remove
```

This will delete all resources from AWS.

Q2 Case study for SonarQube

Creating your own profile in sonarqube for project quality. we sonarqube to analyze Github code. Install sonarlint in your java i to analyze java code , Analyze python pro with sonarqube.

→ Sonarqube is an open source platform ued continous inspection of code quality. It dete bugs , code smells and security vulnerabilit in project across various programming

1) Profile creation in SonarQube.

Quality profiles in sonarqube are essential tions that define rules applied during cod Each project has a quality profile for every supported language with default being 'Son omes built in for all languages. Custom profiles can be created by copying or extending existing ones. copying creates an independent profile, while extending inherit rules from parent profile and reflects future changes automatically you can activate or deactivate rules, Priotize certain rules and configure parameters to tailor profile to specific projects. Permissions to manage quality profile are restricted to users with administrative priveleges. SonarQube allows for the comparision of 2 profiles to check for differences in activated rules and users can take back hanges via event log. quality profiles can also e imported from other instances via backup and restore. To ensure profiles include new rules ts important to check against updated built n profiles or use sonarqube rules page.

sing sonarcloud to analyze GitHub code:
onarcloud is cloud-based conterpart of sonarqube hat integrates directly with GitHub, BitBucket, Azure nd Gitlab repositories. To directly started with onarcloud via Github signup via sonarcloud roduct page and connect your Github organization are personal account once connect, Sonarcloud irror your Github setup with each project orresponding to Github repository. After setting p the organization choose subscription lon. next implement repositories into your sonarclo

organization where each github repo becomes
sonarcloud project.

3) Sonarlint in Java IDE:
sonarlint is an IDE that performs on-the-
code analysis as you write code. It helps
developers detect bugs, security vulnerabilitie
and code It helps developers detect bugs,
security vulnerabilities Idea or Eclipse. To
set it up, install the SonarLint plugin, confi
the connection with sonarqube or sonarcl
and select the project profile to analyze
Java code. This approach ensures immedia
feedback on code quality, Promoting clean
and maintainable code from beginning.

4) Analyzing python projects with SonarQube
SonarQube supports Python test coverage r
generate the coverage port. To enable cover
adjust your blind process so that covera
tool runs before sonar scanner and e
report file is saved in different part
for setup your tox .ini include configurat
for pytol and coverage to generate cover
report in XML format. The build proces
can also be automated using Github A
which install dependencies, tests and invok
SonarQube scan. Ensure report in cobertu
xML format and plays where scanner c
access it

Teacher's Sign.:

Terraform "self-serve" Infrastructure model

① Terraform Modules for self-serve infrastructure

. create Terraform modules that codify the standards for deploying resources like VPCs, EC2 instance and S3 buckets.

Example module for am EC2 instance.

ec2-module/main.tf.

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

ec2-module/output.tf:

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

eams can now use this module to display EC instances with.

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

② Terraform cloud integration with Servia Now

- you can integrate Terraform Cloud with serv Now to automate the infrastructure request process.
- Using Terraform's API-driven approach, service Now can trigger Terraform runs based on ticket approvals, automating resource deployment.

Example workflow

① A product team submits a request in servi Now for new infrastructure.

② The request triggers a Terraform cloud upd the serviceNow ticket with the status and resource details.

③ Creating Terraform Modules for teams reusable for commonly requested resources

① Networking (VPC, subnets)
② Compute (EC2, Autoscaling Groups)
③ Storage (S3, RDS)
⑤ IAM Roles / policies

By doing this, teams can manages their own infrastructure while maintaing compliance with organizational standards.