Nikita Tuadani
55
DISC

Advance DevOps Assignment 2

Create REST API with the serverless framework:

Steps to create a REST API with serverless work:

Install serverless Framework globally using the following command on the terminal:

```
install -g serverless
```

This command installs the serverless your machine globally using npm.

Create a new services with Aws Nodejs serverless

```
create template aws-nodejs
```

This command initialises a new serverless service called rest-api. Using Nodejs on Aws Lambda

Navigate to the project directory:

```
cd rest-api
```

This command changes directory into the newly created directory to manage files

Initialise Nodejs project and install dependencies

```
npm init -y
npm install express serverless http
```

dependency build the REST API and serverless.

Edit the serverless.yaml file to include service

```
rest api
provider
name - aws
runtime : nodejs 14.x
stage : dev
region : us - east-1
```

Funktion :

app:
handler : handler app
events :
- http:
  Path: /
method: any.

This configuration specifies the service name AWS provides setting and defines lambda function with HTTP events.

6) Edit handler js to add the Express app.
const express requires (express)
const Serverless requires (Serverless : http)
app - get (' /hello world) (req - res) ⇒ res jion ({ mes
'Hello world' }).

7) Deploy the service
serverless the service
Deploy the API to AWS setting up resources li
Lambda and API

8) Test the deployed API
Curl https : // <api - id > ececute - api (region) . ama
com / dev / hello world

9) Redeploy after update :
serverless deploy
After modifying the code redeploy it to up
API with service

10) Remove the service
serverless remove
The above command removes all AWS resource

associated with the API ensuring that there are charges for unused services.

- 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 Intelly Eclipse IDE and analyse your Java code
Analyse python project with Sonarqube.
Analyse Nodejs project with Sonaqube

Create your own Profile in Sonarqube :-
Download and install sonarqube from the official website unzips the file and start the server by running : bin/ cos window x 86-64 / start sonar.
Log in Sonarqube using the default credentials name : admin, password admin). After logging in, change the password.
Navigate to project tap, click on create new project assign a project by and name generate project token

Use Sonarcloud to analysis your Github code :-
Sign up for Sonarcloud from official website, using your GitHub account.
An Sonarcloud under project > create project, choose

your GitHub account.

3) Add a Sonar project, properties file in the root of your repository with following code
Sonar Project key < your project - key >
Sonar organisation < your - organisation >

4) Use Sonar Scanner to analysis the code by running thee following command, sonar Scanner

☆ Install SonarLint in your Java Intellit :-

1) Install SonaLint by going out Intellit or Eclipse going to plugin / market place.
2) In the IDE, Configure SonaLint by linking to your SonarQube or Sona cloud project.
3) Open a Java project and use SonarLint to analyse it. It will display issue directly in the IDE while coding.

☆ Analyse python project with SonarQube :-

1) Set up a python code in a project and ensure that SonaQube is running locally.
2) Download and configure Sona Scanner from it offical website and in Sonar Project, Properties
3) Run the analyise of the project by executi from the project directory. Sonar, scanner The result will be pushed to your local sonarqube server.

* Analyise Node.js project with SonarQube:-

1) Set up a Node.js project.
2) In SonarQube ensure that all Java script / Type script plugins have been installed. Plugins can be installed from the Marketplace tab in Sonarqube.
3) Create Sonar.project, properties file in your project root and include following in it.
   Sonar project key node, project
   Sonar language js.
   Sonar Sources.
4) Run the analyise of the project by executing the Sonar Scanner command.
   SonarQube will analyise the Node.js project and show result on the dashboard, highlighting code, quality, bugs and vulnerabilities.

> At a large organisation your centralised operation, may get many repetitive infrastructure request. You can see terraform to build a self serve infrastructure model that lets product teams manage their infrastructive independenty. Terraform cloud can also integrate with ticketing system like service Now generate new infrastructure requests.

> Creating a self service infrastructure model using Terraform for a large organisation involves

for a large orangisation involves the following steps:-

steps 1 :- Define infrastructare standard:-
Istablish clear standard and best practices for infrastrue deployment including naming conventions, resources tupp, tagging policies and security complaince.

step-2 :- Create Terraform modules:-
Develop revsable Terraform modules based on your Oranganisation, standard.

step-3:- Set up Terraform cloud or Interprise
Use teraform cloud or interprise for centralised management of configuration and state files Location and access control for infrastructure

Step 4:- Configure Version Control:-
Integrate teraform modules with version control. This tracks changes facilities collaboration and ensure proper versioning.

step-5 - Integrate with service. Now or other ticketing systems:-
Integrate with systems like services. Now to automate infractructure request. This triggers.

Step-6 :- Provide Documentation and Training :-
Create documentation and training for using Terraform modules and submitting request, helping teams understand best practices.

Step-7 :- Monitor and Support :-
Monitor the usage of the self service model provide ongoing support to users. Gathering feedback help identify pain points and area.

Step-8 :- Iterate and Improve :-
Regularly rewew and update Terraform modules mentation and policies based on feedback and changing ment organisational needs continous.
Security and compliance