Nayaab Jindani
D15C
20

Page No.
Date 05/09/2

Advance Devops - Assignment 2

1. Create a REST API with serverless framework.

→ Steps to create a REST API using the serverless framework on AWS by using AWS Lambda with AWS API Gateway:

1. Sign in to your AWS account.
2. In services search for Lambda and select it.
3. click on create function and choose the "Author from scratch" radio button.
4. In the Basic information section enter a function name, choose a runtime (eg. Node.js, Python) from the dropdown list and lastly select create a new role with basic lambda ~~permissions~~ permissions and select create function.
5. On the next screen which appears, scroll down to the code source section wherein you have to write the logic of your function.

Here's a simple python sample code:

```
import json
def lambda_handler(event, content):
return {
    'statusCode': 200,
    'body': json.dumps({'message': 'this is advance devops
    assignment'})}
```

6. click on the deploy button to save your changes
7. Now to setup the API Gateway search for "API Gateway" in the console and select it.
8. click on create API and select the Build button for REST API.
9. Choose new API and enter an API name, choose create API.

Teacher's Sign.: _____

10. To create a resource select the create resource button on the Resources section that appears.
11. Enter a resource name and select create resource.
12. Now we will create a method. For that select create method button and select 'Get' for method type, Lambda function for integration type, check the Lambda proxy integration and choose the previously created function
13. Once method is created, click on Deploy API.
14. Select an existing stage or create a new one by entering a stage name and select deploy.
15. Copy the invoke URL shown after deployment.
16. Append the resource path to this URL and paste in your browser.
17. You will be able to see the output in the form of JSON response on your browser.

2. case study for sonarqube.

→ • create own profile in sonarqube for testing project quality:

1. Log in to SonarQube
2. Navigate to quality profiles on the top.
3. click on create and choose a language for which you want to create a custom profile.
4. Name the profile and click create.
5. After creating the profile you can add or remove rules based on project needs on the rules page.

• Use sonar cloud to analyze your github code:

1. go to SonarCloud.io and sign up using your github account.
2. Once logged in, click on create new project.

Teacher's Sign.: _____

3. Select github as the source and choose the repository you want to analyze.
4. Follow the guided setup to configure your project.
5. Finally setup sonar cloud github action to run analysis automatically on every commit.

- Install SonarLint in your Java IntelliJ Idea or Eclipse IDE and analyze your java code:
  1. To install sonar lint in eclipse, go to help > eclipse marketplace.
  2. search for sonarLint, click install and follow the prompts to complete the installation.
  3. After ~~installation~~ installation, configure SonarLint to connect to your SonarQube / SonarCloud server.
  4. Go to file > settings > ~~tool~~ tools > SonarLint > Connections
  5. Add your SonarQube or Sonar cloud server, provide necessary credentials and associate the project.
  6. To analyze java code: open a java file and look for SonarLint suggestions in the SonarLint issues window.
  7. SonarLint will analyze your code in real time and provide issues ~~for~~ with suggestions on how to resolve them.

- Analyze python project with SonarQube:
  1. create a sonar-project.properties file in the root directory of your python project.
  2. Run SonarQube scanner in project's root directory.
  3. Scanner will analyze the python code and send results to SonarQube dashboard.

- Analyze Node.js project with SonarQube.
  1. Similar to the steps followed in analyzing the python

Teacher's Sign.: _____

in root dir of ~~node.js~~ proj

project, create a Sonar-project.properties file and run the scanner to get the analysis on the Sonar Qube dashboard.

3. At a large organisation, 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 teams to efficiently deploy services in compliance with your organization's practices. Terraform cloud can also integrate with ticketing systems like ServiceNow to automatically generate new infrastructure requests.

→ In large organizations where the centralized operations team often handles numerous repetitive infrastructure requests, using Terraform to build a ~~self~~ self-serve infrastructure model can significantly improve efficiency.

1. Self-serve infrastructure with terraform modules: Reusable Terraform modules allow product teams to independently deploy ~~infr~~ infrastructure like VPC or databases while adhering to organizational standards. This reduces burden on IT teams and so speeds up provisioning by reusing pre-built modules.
2. compliance and ~~Standards~~ Standardization: Modules enforce consistent use of standards like instance types or security policies.
3. Terraform cloud integration: Terraform cloud integrates with ticketing systems like ServiceNow automating infrastructure requests.
4. Automation and collaboration: Terraform cloud's

Teacher's Sign.: _____

workspaces allow teams to collaborate on infrastructure for different environments. Automated planning and application of changes reduce manual effort and minimize human errors

Overall this ~~app~~ approach allows product teams to manage their own infrastructure in a controlled, compliant manner while freeing up the centralized operations team to focus on more strategic tasks.