Kunal Punjabi
DISA
44

Fdu DevOps - Assignment 2.

04/05

Create a Rest API with Serverless framework.

-> creating Rest APIs with Serverless framework is an efficient way to deploy Serverless application that can scale automatically without managing servers

(i) Serverless framework:
A Powerfull tool that deployment of Services and Serverless application across various cloud providers such as AWS, Azure and Google Cloud.

ii) Serverless architecture: This design model allows developers to build application without worrying about underlying infrastructure enabling focus on code and business logic.

iii) Rest API: Representation State transfer is architecture style for designing application.

Steps for creating REST API for Serverless framework.

1) Install Serverless framework:
You start by installing Serverless framework globally using Node Package Manager (NPM) this allow you to manage Serverless application directly from your terminal.

creating a Node.js Serverless project.
A directory is created for your project where you initialize a Serverless Service. using the command Serverless Create you set up a template for AWS node.js microservices that will automatically deploy to AWS Lambda.

Project Structure: The project creates essential files like handler.js (which contains) code for Lambda functions and Serverless.yml.

Create a Rest API Resource.

5) Deploy the Service:
with the sls deploy commands Serverless framework packages your application uploads, ~~at~~ necessary resources on Aws and Setup the infrastructure.

6) Testing the API:
Once deployed you can test REST API using tools like cu or Postman by making post request to generated API

7) Storing data in Dynamo DB: To store Submitted Candidate data you integrate AWS DynamoDB as a data

8) Adding more functionalities like list all Candidates get Candidate by ID

9) AWS IAM Permission
You need to ensure that Serverless framework is given rig Permission to interact with Aws resources like Dynamodb.

10) Monitoring and maintance.
After Deployment Serverless framework provides Service informa like deployed endpoints, API Key, log Structure.

case Study for SonaQube.

-> SonarQube is an open Sourc platform used for continous inspection of quality to detect bugs, code smells and security vulnerabilities in proje across various programming langvage.

1) Profile creation in SonarQube.

Quality profiles in SonarQube are essential configuration that define rules applied during code analysis. Each project has a quality profile for every supported language with default being 'Sonar way' Profile comes built in for all languages. Custom profiles can be created by copying or extending existing ones. Copying creates an independent profile, while 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 prioritize certain rules to Specific projects. Permissions to manage quality profile are restricted. To ensure profiles include new rules its important to check against updated built in profiles or use Sonarqube rules page.

2) using Sonar Cloud to analyze GitHub code:

Sonar Cloud is Cloud-based counterpart of SonarQube that integrates directly with GitHub, Bitbucket, Azure and Gitlab repositories To get Started with Sonar-Cloud via GitHub Signup via Sonar Cloud product page and connect your GitHub Organization or personal account once connected, Sonar Cloud mirrors your GitHub setup with each project corresponding to GitHub repository. After setting up the organization Choose Subscription plan Next, import repositories into your SonarCloud Organization where each

repo becomes a Sonar Cloud proje. Defin 'new cod' o
focus on recent changes and choose between automat
analysis or CI-based analysis Automat analysis happ
directly in Sonar Cloud, while CI based analysis inte
with your build proces once the analysis is comple
results can be viewed in both Sonar-Cloud and
GitHub including Security impor issue.

3) Sonarlint in JAVA IDE:
Sonarlint is an IDE that peforms on-the fly code
analysis on your writ code. It helps developers dete
bugs, Security vulnerabilites and Code Smells direct
in the development environment Such as INTELLIJ idea
Eclipse. To Set it up install the Sonarlint Plugin,
Configure the connection with Sonar Qube or Sonar Cloud
and Select the Projec Profile to analyze Java Code.
This approach ensures immediate feedback on code
quality, Promoting clean and maintainable code
from beginning.

4) Analyzing Python Projects with SonarQube:
SonarQube Supports python but Coverage, reporting but
requires third Party tool like Coverage.Py to generate
Coverage reports in XML format. The build Process c
also be automated using GitHub Actions, which inst
dependencies, runs, tests and invoke SonarQube Scan.
Ensure report in Cobertura XML format and Place
where Scanner can access it.

Analyzing Node.js Projects with SonarQube

For Node.js projects SonarQube can analyze JavaScript and TypeScript code. Similar to the Python setup, you can configure SonarQube to analyze Node.js projects by installing the appropriate plugins and using Sonar Scanners to scan the projects. SonarQube will check the code against industry standard rules and best practices, flagging issues related to security vulnerabilities, bugs and performance optimization.

At a large organization your centralized operations team may get many repetitive infrastructure requests. You can use Terraform to build a self service infrastructure model that lets product teams manage their own infrastructure independently. You can create and use Terraform modules that codify the standards for deploying infrastructure. Organizations can enhance efficiency, reduce bottlenecks and compliance with established needs. Terraform Cloud can also integrate with ticketing systems like ServiceNow to automatically generate new infrastructure requests.

Implementing a self service infrastructure model using Terraform can transform how large organizations manage their infrastructure independently. Organizations can enhance efficiency, reduce bottlenecks and compliance with established needs.

The need for self-service infrastructure: In large organizations operating teams often face an overwhelming number of repetitive requests this can lead to delay in service delivery and move quickly.

A self service model allows teams to provision and manage their own infrastructure without or the operations te for each request.

Benefits of using Terraform.

1) modularity and Reusability.
Terraform modules encapsulate standard configuration for various infrastructure components networks databases, compute resources
Teams can reuse these modules across different projects reducing redundancy and minimizing the risk of errors.

2) Standardizations:
By defining best practices within modules organizations can ensure that all deployments comply with natural policies and standards
This consistency helps maintain security and operational integrity across the organization.

4) Integration with existing System Terraform Cloud can integrate with existing systems like ServiceNow. to automate the generation their existing platform, reducing manual intervention.

Implementation Steps

1. Identify infrastructure components Begin by identifying
which components of your infrastructure will comprises of
eg VPCs, Security group, load balancers.

2. Develop Terraform modules.
Create reusable modules to define the desired configuration
and resources.
Ensure each module includes input variable for
customization and output for integration with other

Establish Guidelines and Best practices:
Define guidelines for module usage, versioning and
documentation to ensure clarity and maintainability.

Testing and validation

Implement a testing to validate module
functionality before deployment.

Best practices for Module management

Utilize the terraform registry.

Leverage existing community modules from the
Terraform Registry to avoid reinventing solutions
and Sure to best Practices.

Version control:

Implement versioning for your modules & tools
Overview this helps many dependency efficiency
minimize disruption during updates.

Ensuring collaboration: fosters a culture of collaboration
Sharing Modules across teams. This promotes consistency
in deployments and By adopting a self-service
infrastructure model with transform organization
can empower product teams to efficiency more
approach Not only streamlines process but also enhance
agility in responding to changing business needs.
ultimately it leads to a more responsive IT
environment that supports innovation and growth within the
organization.