

AdDev Ops Assignment 2

1

- 1) Create a REST API with the Serverless Framework.

"Serverless Framework" refers to the user not needing to manage and maintain their own servers.

Steps :-

- 1) Install serverless framework (CLI globally) using 'node package manager' (npm).
- 2) Create node JS serverless project, inside a directory intended for it.  
This service will house all your Lambda functions, configurations and cloud resources.
- 3) The project scaffold creates essential files like handler.js which contains code for Lambda functions and serverless.yml.
- 4) Create a REST API resource that handles HTTP requests.

- 5) Deploy the Service and upload new resources to AWS and set up the infrastructure.
- 6) Once deployed, test the REST API tools like curl or Postman by making post requests to generated API.
- 7) To store submitted data you integrate AWS DynamoDB as a database.
- 8) Add functionalities like list all candidates by ID.
- 9) You need to ensure that serverless framework is given right permissions to interact with AWS resources like Dynamo DS.
- 10) After deployment serverless framework provides service information like deployed endpoints, API key, log streams.

## Case Study for SonarQube

- Creating ~~new~~ profile in SonarQube for testing project quality.
- SonarQube is an Open Source platform used for continuous inspection of quality.
- SonarQube detects bugs and code smells along with vulnerabilities in projects across various programming languages.

### Profile Creation:

Quality profiles in SonarQube are essential configurations that define rules applied during code analysis. Each project has a quality profile for every supported language with default being 'Sonar Way'.

Custom profiles can be created by copying or extending existing ones.

Permissions to manage quality profiles are restricted to users with administrator privileges. Permissions can also be imported from other instances via backup and restore.

## 2.) SonarCloud to analyze GitHub code:

SonarCloud is a cloud-based counter of SonarQube that integrates directly with GitHub, BitBucket, Azure and GitHub repositories.

To get started with SonarCloud via GitHub setup with each project corresponding to GitHub repositories. Automatic analysis happens directly in SonarCloud while CI based analysis integrates with your build process so the analysis is complete results can be viewed in both SonarCloud and GitHub including including security import issue.

## 3.) SonarLint in Java IDE:

SonarLint is an IDE that performs on-the-fly code analysis as you write code. It helps developers in development environments, such as IntelliJ IDEA or Eclipse. This approach ensures immediate feedback on code quality, promoting clean & maintainable code from the beginning.

## ➤ Analyzing Python projects with SonarQube:

SonarQube supports Python test coverage reporting but it requires third party tools like coverage. For setup, you can use .Travis and Coverage. PyTest configure and run tests in your tox.ini.

The build process can also be automated using GitHub Actions, which installs dependencies, runs tests and invokes SonarQube scan. Ensure report is in XML format.

## ➤ Analyzing Node projects with SonarQube:

You can ~~configure~~ SonarQube to analyze Node JS projects by installing the appropriate plugin and using SonarScanner 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.

Q.3 At a large organization, your central operations team may get many repetitive infrastructure requests. You can create and use Terraform modules that codify the standards for deploying & managing services in your org. Terraform cloud can also integrate with ticketing system like Jira.

Ans) Implementing a 'self-service' infrastructure model using Terraform can transform how large organizations manage their infrastructure independently.

Organizations can enhance efficiency, reduce bottlenecks, and ensure compliance with established needs.

→ Overwhelming number of repetitive requests leads to delays in service deliveries.

→ Benefits:

- Modularity and Reusability
- Standardization
- Increased Efficiency
- Integration with ticketing systems

## \* Implementation Steps :

- 1) Identify Infrastructure components
  - Begin by identifying which components of your infrastructure can be modularized
- 2) Establish Governance and Best Practices to ensure clarity and maintainability.
- 3) Develop Terraform modules to include input variables for customizations and outputs for integration with other modules.
- 4) Testing and Validation of best modules for module management, utilize the Terraform registry.
- 5) Version Control for modules to track changes over-time. This helps manage dependencies effectively and minimize disruptions during updates.
- 6) Documentation for each module, usage examples, input/output descriptions and any dependencies.

• Encourage collaboration and promote consistency in deployments and facilitate knowledge within the organization. Their approach not only streamlines processes but also enhances agility in responding to changing business needs.

Ultimately it leads to a more responsive IT environment that supports innovation and growth within the organization.

