

## Advance Devops

### Assignment 2

Q1

Create a REST API with serverless framework

→ Creating REST API with serverless framework is an efficient way to deploy serverless applications that can scale automatically without managing servers.

(i) serverless framework : A powerful tool that deployment of services and serverless applications across various cloud providers such as AWS, Azure and Google Cloud.

ii) serverless applications across various clouds. This design model allows developers to build applications without worrying about underlying infrastructure, enabling focus on code and business logic.

iii) REST API : Representational state transfer is architecture style for designing network applications.

Steps for creating REST API on serverless framework:

1) Install serverless framework

You start by installing serverless framework UI globally using node package manager (npm). This allows you to manage serverless applications directly from your terminal.

2) Creating a Node.js serverless project

A directory is created for your project, where you will initialize a serverless service.

3) Project structure

The project scaffold creates essential files like

handler.js (which contains code for lambda functions) and serverless.yml

- a) Create a REST API Resource:  
In the serverless.yml file you define function that handles post requests of http.
- 5) Deploy the service:  
with the 'sls deploy' command serverless framework packages your application, uploads necessary resources to AWS and sets up the infrastructure
- 6) Testing the API : Once deployed you can test REST API using tools like curl or postman by making post requests to generated API.
- 7) Storing data in DynamoDB : To store submitted candidate data you integrate AWS DynamoDB as a database
- 8) Adding more functionalities : Adding functional like list all candidates, get candidate by ID
- 9) AWS IAM Permissions : You need to ensure that serverless framework is given right permissions to interact with AWS
- 10) Monitoring and maintenance : After deployment serverless framework provides monitoring information like deployed endpoints, API key log streams

(Q2)

case study for sonarqube  
Creating your own profile in sonarqube for testing project quality. Use sonarqube to analyze your github code. Install sonarlint in your java intellij ide and analyze java code. Analyze python project with sonarqube.

Sonarqube is an open source platform used for continuous inspection of code quality. It detects bugs, code smells and security vulnerabilities in project across various programming languages.

### 1) Profile creation in Sonarqube:

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' profile or come built in for all languages. Custom profiles can be created by copying or extending inheritance rules from parent profile. They are restricted to users with administrative privileges. Sonarqube allows for the comparison of two profiles to check for differences in activated rules and users can track changes via event log. Quality profile of basic.

### 2) Using SonarCloud to analyze Github code

SonarCloud is cloud-based counterpart of SonarQube that integrates directly with Github, BitBucket, Azure and GitLab repositories. To get started with SonarCloud via github organization or personal account.

Once connected, SonarCloud mirrors your GitHub setup with each project corresponding to a GitHub repository. Next repositories into your SonarCloud organization took where each GitHub repo becomes a SonarCloud project. Define 'new code' to focus on recent changes and choose between automatic analysis or I-based analysis.

### 3) Sonarlint in Java IDE:

Sonarlint is an IDE that performs on-the-fly code analysis as you write code. It helps developers detect bugs, security vulnerabilities and code smells directly in the development environment such as IntelliJ IDEA or Eclipse. To set it up, install the Sonarlint plugin, configure the connection with SonarQube on SonarCloud and select the project profile to analyze the Java code.

### 4) Analyzing Python Projects with SonarQube

~~SonarQube supports Python test coverage, reporting but it requires third party tool like coverage To enable coverage adjust your build process so that coverage tool runs before Sonar scanner and ensures report file is saved in different path.~~

For setup you can use Tox, Pytest and coverage to configured and run test. In your tox.ini include configurations for pytest and coverage to generate coverage report in XML format. The build process can also be automated using GitHub Actions which install dependencies, tests and invokes SonarQube.

3) Analyzing Node.js projects with SonarQube.  
 For Node.js project 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 plugin and using SonarScanner to scan the project. SonarQube will check the code against Industry

3 At a large organization, your centralized Operations team must get many repetitive infrastructure requests. You can use Terraform to build a "self-service" infrastructure. You can use Terraform modules that codify the standards for deploying and managing services in compliance with your organization's practices. Terraform cloud can also integrate with ticketing system like ServiceNow.

→ 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.

The need for self-service infrastructure  
 In large organizations, centralized operations often face an overwhelming number of repetitive requests. This can lead to delays in service and frustration among product teams who need to move quickly. A self-service model allows teams to provision and manage their infrastructure without relying on the operations team for

- Benefits of using Terraform modularity and reusability
- Terraform modules encapsulate standard configurations for various infrastructure components.
- Teams can reuse those 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 deployment comply with internal policies and standards

## 3) Increased Efficiency:

- Product teams can deploy services quickly by using pre-defined modules, significantly reducing the time spent on infrastructure setup

## 4) Integration with Ticketing Systems.

- Terraform cloud can integrate with ticketing systems like ServiceNow to automate the general streamlining workflows by allocating teams to individual

## → Implementation steps

### 1. Identify Infrastructure Components

Start by identifying which components of your infrastructure can be modularized. VPCs, security groups, load balancers.

- 2) Develop Terraform modules
- Create reusable modules that define the desired configurations and resources
- 3) Establish governance and Best Practices
- Define guidelines for module usage, versioning, and documentation to ensure clarity and maintainability
  - Encourage teams to contribute to module development and share improvements

#### 4) Testing and validation

- Implement a testing framework 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 ensure adherence to best practices
- Version control: Implement versioning for your modules to track changes over time
- Documentation: maintain comprehensive documentation for each module including usage examples, input descriptions, and dependencies

Encourage collaboration: Foster a culture of collaboration by sharing modules across teams to efficiently manage their own infrastructure while ensuring compliance with established

standard. This approach not only streamlines processes but also enhances agility in responding to changing business needs.