ADV. DevOps
Roll.No .42
NAME: Pranav Pol STD.:________ DIV.: D15A

04/05

DATE :
PAGE :

Assignment No. 2.

Q.1 Create a REST API with the serverless Framework.

The serverless Framework is an open source that simplifies the deployment and management of serverless applications. It allows developers to build and deploy application on Cloud platform Like AWS without having to manage the underlying infrastructure.

Step 1 : Install the serverless Framework.
First ensure you have Nodejs and npm installed. Then ensure install the serverless Framework globally

```
npm install -g serverless
```

Step 2 : Create a new serverless service.

```
serverless create --template aws-nodejs
--path my-service cd my-service.
```

Step 3 : Define Function in serverless.yml
open serverless.yml file in project directory.
In this file we define our service configuration including the functions and their triggers / events
This file contains service, Provider it name and runtime environment. It also contains Functions which contains Create, read, update and delete methods.

NAME:________________ STD.:______ DIV.:______ DATE : PAGE :

Step 4: Implement lamda function.
Create a Handler.js file in your project directory. In this file we write the Code for our lambda functions.
This file Contains export function of modules of various CRUD methods.

Step 5: Deploy your service and AWS Config
• Configure AWS Credentials using
aws Configure.
• Deploy your Service to aws using
serverless deploy.
This Command packages our services, uploads it to AWS, and Setsup necessary infrastructure.

Step 6: Testing our API.
After deployment, the Serverless framework will provide us with the endpoint for our API. We Can test these endpoint using tools Like Postman or CURL.
For example, To test the Create function we Can send the POST request to the provided URL

NAME:________________ STD.:______ DIV.:________ DATE : PAGE :

Q.2. Case Study for SonarQube.

- Create your own profile in SonarQube for testing project Quality.

Open sonarqube and log into as an administrator Then ~~to~~ Go to administration of Quality profiles Create and name your profile. Based on project requirements add or remove rules and save your new quality Projects.

- Using SonarCloud for Github Code Analysis

Go to SonarCloud website and sign up then Link your github account to SonarCloud. ~~go to~~ Then Choose the ~~repositories~~ to analyse then Follow the prompts to set up and run the analysis. By linking our Github repositories directly we achieved Continous Code quality checks with each Commit. The main Challenge was Configuring the permissions Correctly to ensure all repositories were accessible.

- Install Sonarlint in your Intellij IDE and analyse your Java Code.

For installing Sonarlint in Intellij Go to file settings ~~and~~ then navigate to plugins and click on market place. In marketplace search

NAME:______________ STD.:______ DIV.:______ DATE : PAGE :

For SonarLint and install it. Then we have restart Intellij IDEA to activate the plugin. Installing SonarLint in our IDEs provided us real-time feedback on Code quality issues as we developed.

- Analyse a python Project with SonarQube

For this we need sonar-scanner-cli we are installing it as a docker image Create sonar-Project.properties in your Project directory and ~~to~~ add the properties this includes project key, host URL, and login token, then analyze python project via Command line and integrate this step into our CI pipeline.

- Analyze Node.js project with SonarQube

As we have ~~it~~ already got image then we need to install npm ~~as~~ package of Sonnar Scanner in our Nodejs project. The main Challenge was sonar-project.properties file to ensure all necessary parameter were included.

NAME:________________ STD.:________ DIV.:__________ DATE : PAGE :

Q.3.

A self serve infrastructure model allows individu
product teams within an organisation to
provision and manage their infrastructure without
the need for a centralised operations team. This
is done by creating reusable Terraform modules
that encapsulates the organisation's standard
and best practices. Terraform enables Teams to
define their infrastructure as code, automating the
provisioning and life cycle management of Cloud
resources.
Terraform modules are reusable, versioned, and
shareable components that define infrastructured
resources. They encapsulate best practices and
standards for deploying and managing services.
For example, an organisation may require all its
resources to have specific tagging, use certain
certain group, or enforce encryption on storage
resources. By using Terraform modules, these
requirements can be codified, ensuring compliance
Terraform Cloud is a managed service that provid
for running Terraform commands remotely,
enabling infrastructure as code workflows
with integration into ticketing services system
like serviceNow, infrastructure request can be
automatically generated based on service
tickets, streamlining the provisioning process.

NAME:__________ STD.:______ DIV.:______ DATE : PAGE :

The ticketing system Integration Works as
A user submits a request for infrastructure
through a Service Now Catalog form item. The
Catalog form item include fields for necessary
details such as the type of infrastructure,
Configuration option. Service now automatically
generates a ticket based on the User request
This ticket goes through the approval workflow
It Can be automated based on predifined
rules. once approved the ticket trigger a
Terraform run. Service Now uses the
information in ticket to execute the Corresponding
Terraform module. After this terraform make
provision for requested modules. as Specified
in Ticket. If the infrastructure is Successfully
provisioned the ticket is marked as Completed.
If there are error the ticket is updated with
the error details for further action.