Accuknox DevOps Trainee Practical Assessment

Problem Statement 1:

Title: Containerisation and Deployment of Wisecow Application on Kubernetes

Project Repository: https://github.com/nyrahul/wisecow Wisecow App

Objective: To containerize and deploy the Wisecow application, hosted in the above-mentioned GitHub repository, on a Kubernetes environment with secure TLS communication.

Requirements:

Dockerization:

- Develop a Dockerfile for creating a container image of the Wisecow application.

Kubernetes Deployment:

- Craft Kubernetes manifest files for deploying the Wisecow application in a Kubernetes environment.
- The Wisecow app must be exposed as a Kubernetes service for accessibility.

Continuous Integration and Deployment (CI/CD):

- Implement a GitHub Actions workflow for:
    - Automating the build and push of the Docker image to a container registry whenever changes are committed to the repository.
    - Continuous Deployment *[Challenge Goal]*: Automatically deploy the updated application to the Kubernetes environment following successful image builds.

TLS Implementation *[Challenge Goal]*:

- Ensure that the Wisecow application supports secure TLS communication.

Expected Artifacts:

- A private GitHub repository containing:
    - The Wisecow application source code.
    - The Dockerfile for the application.
    - Kubernetes manifest files for deployment.
    - The CI/CD pipeline configuration.

- A GitHub Actions workflow file for facilitating Continuous Build and Deployment (CI/CD)

Access Control:

- The GitHub repository should be set to private.
- Repository access should be granted to the following GitHub IDs: nyrahul, SujithKasireddy, and divyansh-accuknox.

End Goal: The successful containerisation and deployment of the Wisecow application to the Kubernetes environment with an automated CI/CD pipeline and secured with TLS communication.

Problem Statement 2:

Please choose any two objectives from the list below and attempt to achieve them using either Bash or Python.

1. System Health Monitoring Script:

   Develop a script that monitors the health of a Linux system. It should check CPU usage, memory usage, disk space, and running processes. If any of these metrics exceed predefined thresholds (e.g., CPU usage > 80%), the script should send an alert to the console or a log file.

2. Automated Backup Solution:

   Write a script to automate the backup of a specified directory to a remote server or a cloud storage solution. The script should provide a report on the success or failure of the backup operation.

3. Log File Analyzer:

   Create a script that analyzes web server logs (e.g., Apache, Nginx) for common patterns such as the number of 404 errors, the most requested pages, or IP addresses with the most requests. The script should output a summarized report.

4. Application Health Checker:

   Please write a script that can check the uptime of an application and determine if it is functioning correctly or not. The script must accurately assess the application's status by checking HTTP status codes. It should be able to detect if the application is 'up', meaning it is functioning correctly, or 'down', indicating that it is unavailable or not responding.