Accuknox QA Trainee Practical Assessment Solutions

# Problem Statement 1: Containerisation and Deployment of Wisecow Application on Kubernetes

## Step 1: Dockerization

Create a Dockerfile that defines how to build the image of the Wisecow application.

Example Dockerfile:

FROM python:3.9-slim  
WORKDIR /app  
COPY . .  
RUN pip install --no-cache-dir -r requirements.txt  
EXPOSE 8080  
CMD ["python", "app.py"]

## Step 2: Kubernetes Deployment

Create Kubernetes manifests to deploy the application.

Example deployment.yaml:

apiVersion: apps/v1  
kind: Deployment  
metadata:  
 name: wisecow-deployment  
spec:  
 replicas: 2  
 selector:  
 matchLabels:  
 app: wisecow  
 template:  
 metadata:  
 labels:  
 app: wisecow  
 spec:  
 containers:  
 - name: wisecow  
 image: <YOUR\_CONTAINER\_REGISTRY>/wisecow:latest  
 ports:  
 - containerPort: 8080

Example service.yaml:

apiVersion: v1  
kind: Service  
metadata:  
 name: wisecow-service  
spec:  
 selector:  
 app: wisecow  
 ports:  
 - protocol: TCP  
 port: 80  
 targetPort: 8080  
 type: LoadBalancer

## Step 3: TLS Setup

Create a Kubernetes Secret for TLS and configure the service for HTTPS communication.

Example command for generating certificates:

openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout tls.key -out tls.crt -subj "/CN=wisecow.com/O=wisecow"  
kubectl create secret tls wisecow-tls --cert=tls.crt --key=tls.key

## Step 4: Continuous Integration and Deployment (CI/CD) with GitHub Actions

Create a .github/workflows/ci-cd.yaml file to automate Docker image building and pushing, and deploy to Kubernetes.

Example ci-cd.yaml:

name: CI/CD Pipeline  
on:  
 push:  
 branches:  
 - main  
jobs:  
 build:  
 runs-on: ubuntu-latest  
 steps:  
 - name: Checkout code  
 uses: actions/checkout@v2  
 - name: Set up Docker Buildx  
 uses: docker/setup-buildx-action@v1  
 - name: Log in to DockerHub  
 run: echo ${{ secrets.DOCKER\_PASSWORD }} | docker login -u ${{ secrets.DOCKER\_USERNAME }} --password-stdin  
 - name: Build and Push Docker image  
 run: |  
 docker build -t <YOUR\_DOCKERHUB\_USERNAME>/wisecow:latest .  
 docker push <YOUR\_DOCKERHUB\_USERNAME>/wisecow:latest  
 deploy:  
 runs-on: ubuntu-latest  
 needs: build  
 steps:  
 - name: Set up kubectl  
 uses: azure/setup-kubectl@v1  
 - name: Deploy to Kubernetes  
 run: |  
 kubectl apply -f deployment.yaml  
 kubectl apply -f service.yaml

# Problem Statement 2: Script Development

## 1. System Health Monitoring Script (Bash Example)

#!/bin/bash  
CPU\_THRESHOLD=80  
MEM\_THRESHOLD=80  
DISK\_THRESHOLD=80  
CPU\_USAGE=$(top -bn1 | grep "Cpu(s)" | awk '{print $2 + $4}')  
if (( $(echo "$CPU\_USAGE > $CPU\_THRESHOLD" | bc -l) )); then  
 echo "CPU usage is above threshold: $CPU\_USAGE%" >> /var/log/system\_health.log  
fi  
MEM\_USAGE=$(free | grep Mem | awk '{print $3/$2 \* 100.0}')  
if (( $(echo "$MEM\_USAGE > $MEM\_THRESHOLD" | bc -l) )); then  
 echo "Memory usage is above threshold: $MEM\_USAGE%" >> /var/log/system\_health.log  
fi  
DISK\_USAGE=$(df -h / | grep / | awk '{print $5}' | sed 's/%//g')  
if [ $DISK\_USAGE -gt $DISK\_THRESHOLD ]; then  
 echo "Disk usage is above threshold: $DISK\_USAGE%" >> /var/log/system\_health.log  
fi  
ps aux >> /var/log/system\_health.log

## 2. Log File Analyzer Script (Python Example)

import re  
from collections import Counter  
  
with open('/var/log/nginx/access.log', 'r') as log\_file:  
 logs = log\_file.readlines()  
  
error\_404\_count = 0  
ip\_counter = Counter()  
page\_counter = Counter()  
  
for log in logs:  
 if '404' in log:  
 error\_404\_count += 1  
 ip = re.findall(r'[0-9]+(?:\.[0-9]+){3}', log)  
 if ip:  
 ip\_counter[ip[0]] += 1  
 page = re.findall(r'GET (.\*?) HTTP', log)  
 if page:  
 page\_counter[page[0]] += 1  
  
print(f"404 Errors: {error\_404\_count}")  
print("Most frequent IPs:", ip\_counter.most\_common(5))  
print("Most requested pages:", page\_counter.most\_common(5))