Senior DevOps Engineer Challenge

Task 1

We want to design and deploy the infrastructure for deploying a Java based application in one of our PROD AWS account. This AWS account is currently empty. The Java application is based on Spring Boot and can be deployed by running `java -jar <filename.jar>` . It exposes port **8080.**

The deployment of the Java application will be performed by publishing messages to an AWS SNS topic using a **Lambda** subscription. The following requirements must be considered:

- when the SNS message is **default,** the application will be deployed from the following URL: https://s3.eu-central-1.amazonaws.com/nvplayground/demo-0.0.1-SNAPSHOT.jar
- when the SNS message is **not default,** is should contain a different S3 application URL and that application should be deployed
- when the SNS message is **public,** we want the application to be deployed in a public subnet and be exposed using an **EIP**

Please share with us the Terraform code and the Lambda function code.

Notes

- the Lambda function code can use any language the candidate is comfortable with.
- there is no need to create a cloud account or run the code into an existing cloud account for this task.
- the main focus of this task is to review the infrastructure design and the infrastructure code, we will not validate the code into a cloud account.

Task 2

We want to deploy multiple instances of a sample Go application that will run in our production Kubernetes cluster. The sample application is available at https://github.com/ybkuroki/go-webapp-sample and should not use web server mode during deployment.

For deployment, we need to create a container image for the sample application and a Helm chart for deploying inside the Kubernetes cluster.

The following requirements must be considered when creating the Helm chart:

- the application is exposed using the **test.navvis.com/release_name** URL
- data volume is mounted at **/data** - our storage is external and already supports dynamic provisioning
- permissions to list / read other pods in the same namespace is assigned to the application
- traffic is only allowed from pods with the **k8s-app=ingress-nginx** label
- the manifests for **zaplogger.k8s.yml** and **application.k8s.yml** are created

Please share with us the Dockerfile and the Helm chart.

Notes

- there is no need to create all resources in an existing Kubernetes cluster.
- the Docker image does not have to be uploaded to a repository, any random value can be used in the Helm chart for the container image.
- the main focus of this task is to review the design and the code, we will not validate the code into a Kubernetes account.