### General instructions read carefully:

1. The approach of solving the Problem solely depends on the Candidate
2. Make sure to have Draw.io diagrams for the workflowns and application architecture
3. Every configuration, code written should be pushed on git (Private Repo if Possible)
4. Your are not permitted to share the doc with anyone, even with your colleagues
5. There should be proper Documentations for the Project
6. Use helm3.x when using helm for deploying applications
7. You have to save each and every configuration as the cluster access given to you is for experiments and may get down as people here are experimenting constantly.
8. There is namespace nirmal and you have full admin access to that NS

Sample Labels to be associated with every Kubernetes objects

app.kubernetes.io/name: wordpress

app.kubernetes.io/instance: wordpress-abcxzy

app.kubernetes.io/version: "4.9.4" - Req - Defaults to 0.1 if not available

app.kubernetes.io/managed-by: helm|manifest

app.kubernetes.io/contact: Gursimran

app.kubernetes.io/workload: xenonstack-website

app.kubernetes.io/env: dev|uat|prod

app.kubernetes.io/tier: frontend|backend|ml|de

app.kubernetes.io/project: devops-intelligence

## Problem

**Task 1:**

A ) Bootstrap kubernetes cluster on your laptop using kubeadm

(use master cluster only to get the worloads scheduled on it )

B ) Deploy traefik ingress controller on your K8 cluster (you can use helm for this)

Verify the cluster/ingress controller is operational or not, once things seems good follow below guidelines

**Task 2:**

Taint your node with key as key and value as mosquito and taint effect NoSchedule

Dockerize the App mentioned by the URL <https://github.com/M1TKO/my-note-webapp> and deploy it on Kubernetes using following guidelines

* Database should be external (deploy external DB on Kubernetes)
* app should use persistent volumes (hostpath would work here for us)
* ingress name to access via web should be notes.xenon.team
* app should always scheduled by tolerating the taint
* Demonstrate usage of Readiness and Liveness probe via your application

**Taks-3**   
  
Task to be done via Helm3

There should be a monitoring Stack for the application, So you can use pre built helm chart for Grafana and Prometheus

but the Problem is you have to create a single helm chart combining the existing helm chart of Grafana and Prometheus and then deploy the monitoring stack.

*Caution here* - we do not need NODE and kube system monitoring only application monitoring is required to be done.