**Final Task**

By:- Nirmal Singh

Task 1 :- A ) Bootstrap kubernetes cluster on your laptop using kubeadm

**task 1.A.A)** I have created the shell script and put all neccessary commands in that file for creating.

**1.A.A) KUBEADM INIT**

**File name:- kubeadmInit.sh**

#!/bin/sh

**#kubeadm init**

sudo kubeadm init --ignore-preflight-errors=IsPrivilegedUser,preflight

**#create directory**

mkdir -p $HOME/.kube

**#copy admin.conf file**

sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config

**#add to the super user group**

sudo chown $(id -u):$(id -g) $HOME/.kube/config

**# create pod network**

sudo kubectl apply -f https://docs.projectcalico.org/v3.11/manifests/calico.yaml

**# make master node as worker node**

kubectl taint nodes --all node-role.kubernetes.io/master-

#**Create namespace**

kubectl create ns nirmal

**# RBAC**

kubectl create -f traefik-rbac.yaml

kubectl create -f traefik-service-acc.yaml

COMMAND - **$ bash kubeadmInit.sh**

**Task 1.A.B) KUBEADM RESET**

**File name:- kubeadmReset.sh**

#!/bin/sh

**#Reset the kubeadm**

sudo kubeadm reset --ignore-preflight-errors=IsPrivilegedUser,preflight -f

**#flush out the iptables**

sudo iptables -F

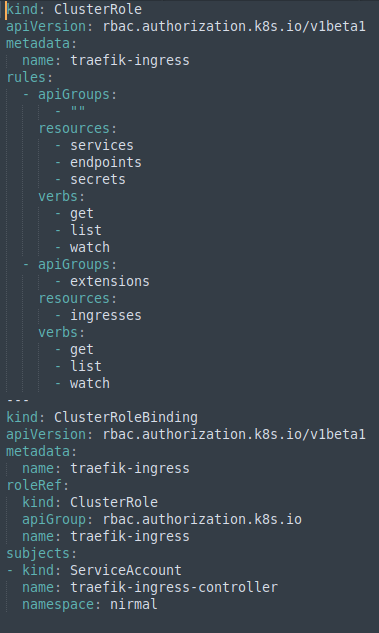
sudo rm /etc/cni/net.d/\*

sudo ipvsadm --clear

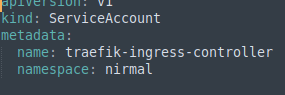
rm /home/nirmalsingh/.kube/config

COMMAND - **$ bash kubeadmReset.sh**

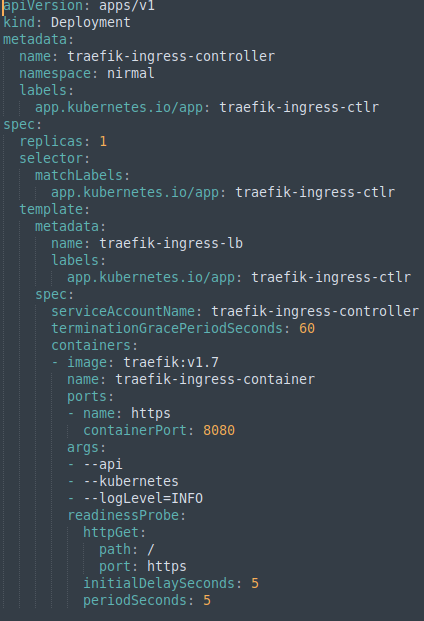
Task 1 . B ) Deploy traefik ingress controller on your K8 cluster (you can use helm for this).



1.B.A ) **CREATE** **SERVICE ACCOUNT**

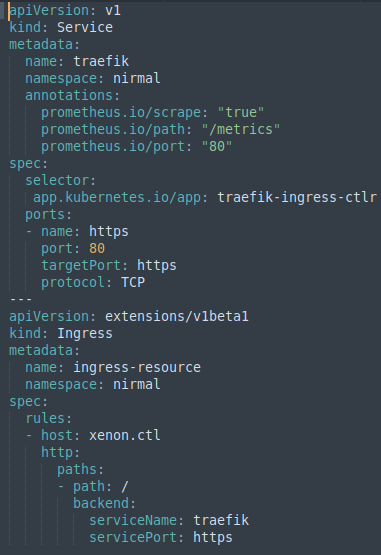
****

**Deployment file for traefik ingress controller**

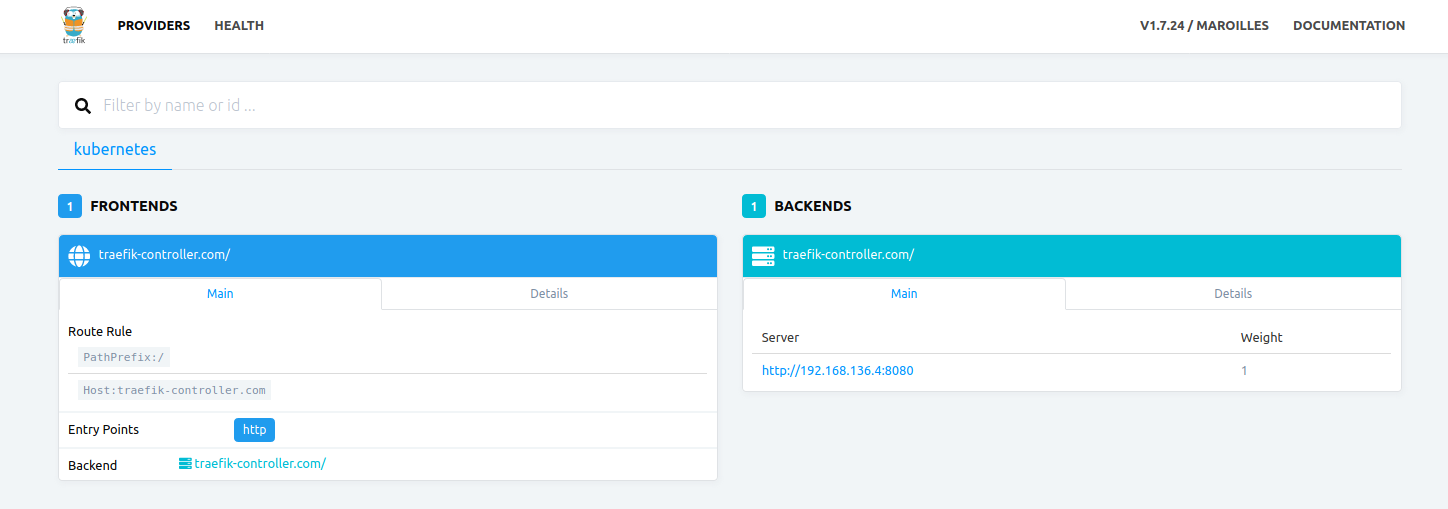
****

COMMAND - **$ kubectl create -f traefik-deployment.yaml**

**Ingress Resource**

****

COMMAND - **$ kubectl create -f traefik-ingress.yaml**

**Verify the cluster/ingress controller is operational or not, once things seems good follow below guidelines: 192.168.1.103:8080**

**Task 2:**

**Dockerize the App mentioned by the URL** [**https://github.com/M1TKO/my-note-webapp**](https://github.com/M1TKO/my-note-webapp) **and deploy it on Kubernetes using following guidelines**

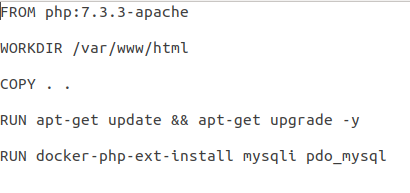
**A. Database should be external (deploy external DB on Kubernetes)**

**B. app should use persistent volumes (hostpath would work here for us)**

**C. ingress name to access via web should be notes.xenon.team**

**D. app should always scheduled by tolerating the taint**

**E. Demonstrate usage of Readiness and Liveness probe via your application**

**Task 2) Create docker file : Dockerfile**

**2.1 Now build and push image to the docker hub**

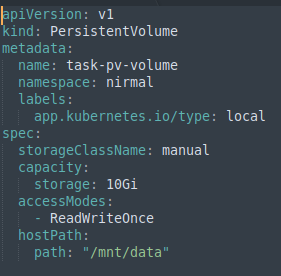
COMMAND - $ docker build -t image-name:version .

$ docker tag <imageID> docker-repo/image-name:version

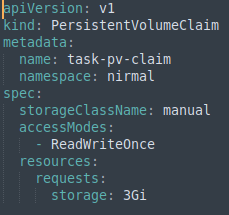
$ docker push docker-repo/php-mysql-image:v 3

**2.2 Create the persistent volume for application**

**1. persistent volume:**



**2. persistent Volume Claim**

****

**a**

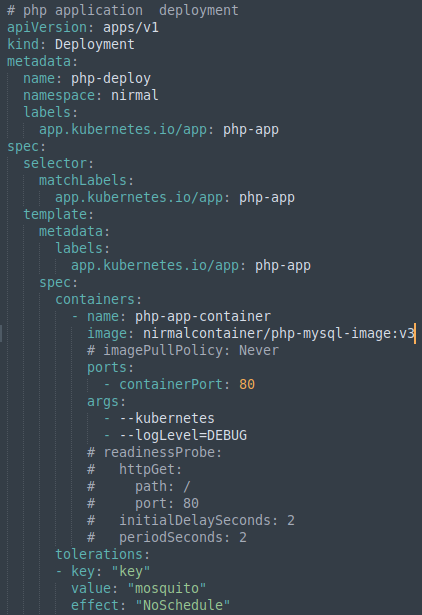
**COMMAND -** $ kubectl create -f pv-volume.yaml

$ kubectl create -f pv-claim.yaml

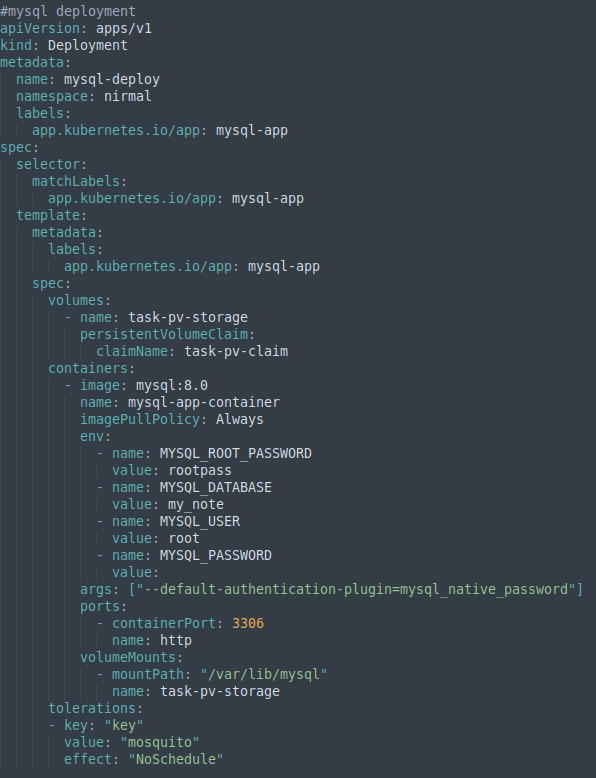
**2.3 Create Deployment for php app, mysql and phpmyadmin in single file :**

webserver.yaml

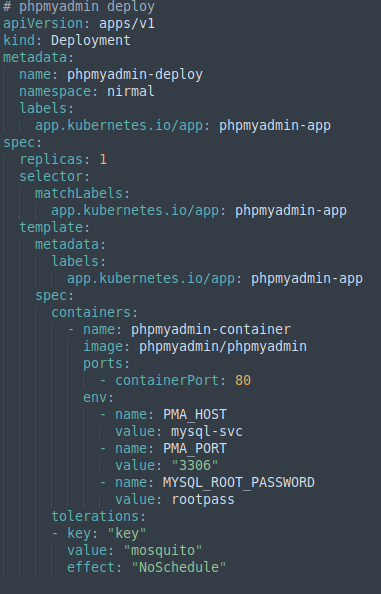
**php-deploy**

****

**mysql-deploy**

****

**phpmyadmin deploy**

****

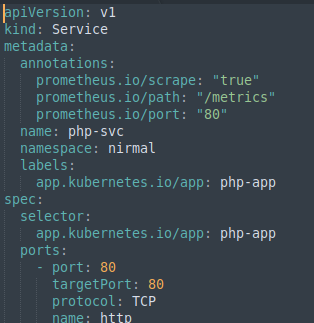
COMMAND **- $ kubectl create -f webserver.yaml**

**2.4 Create Service for each deployment :**

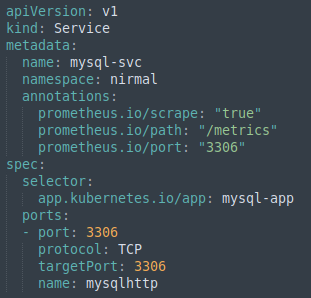
webserver-svc.yaml

COMMAND - **$ kubectl create -f webserver-svc.yaml**

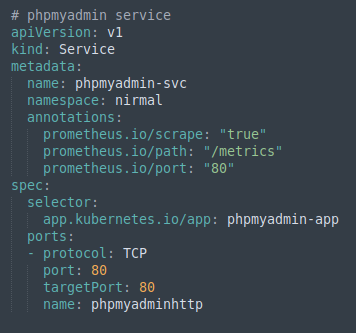
**php-svc**

****

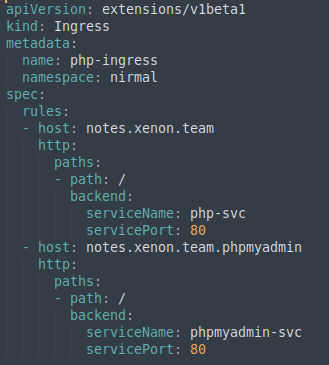
**mysql-svc**

****

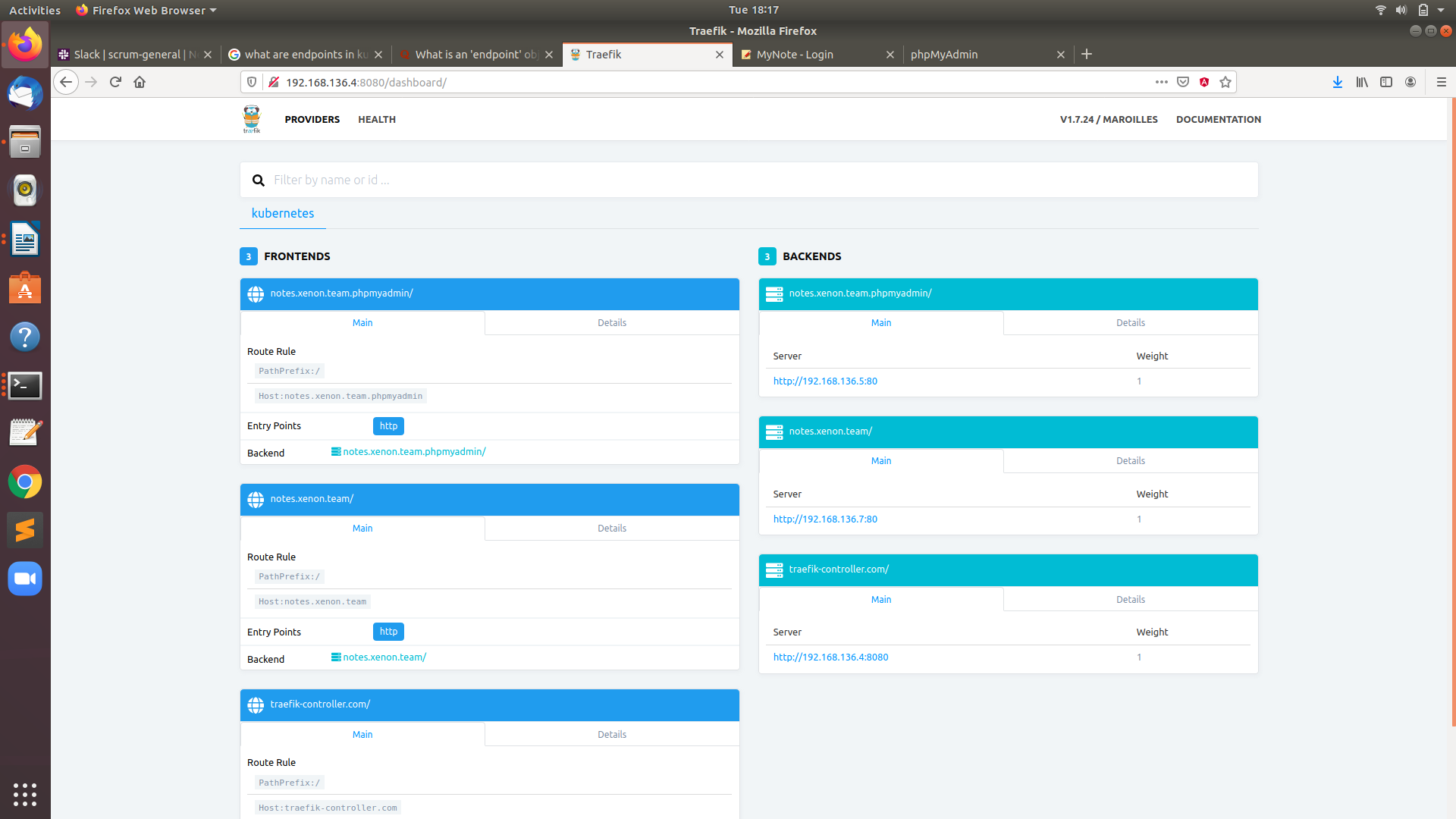
**phpmyadmin-svc**

****

**2.5 Create ingress-Resource file**

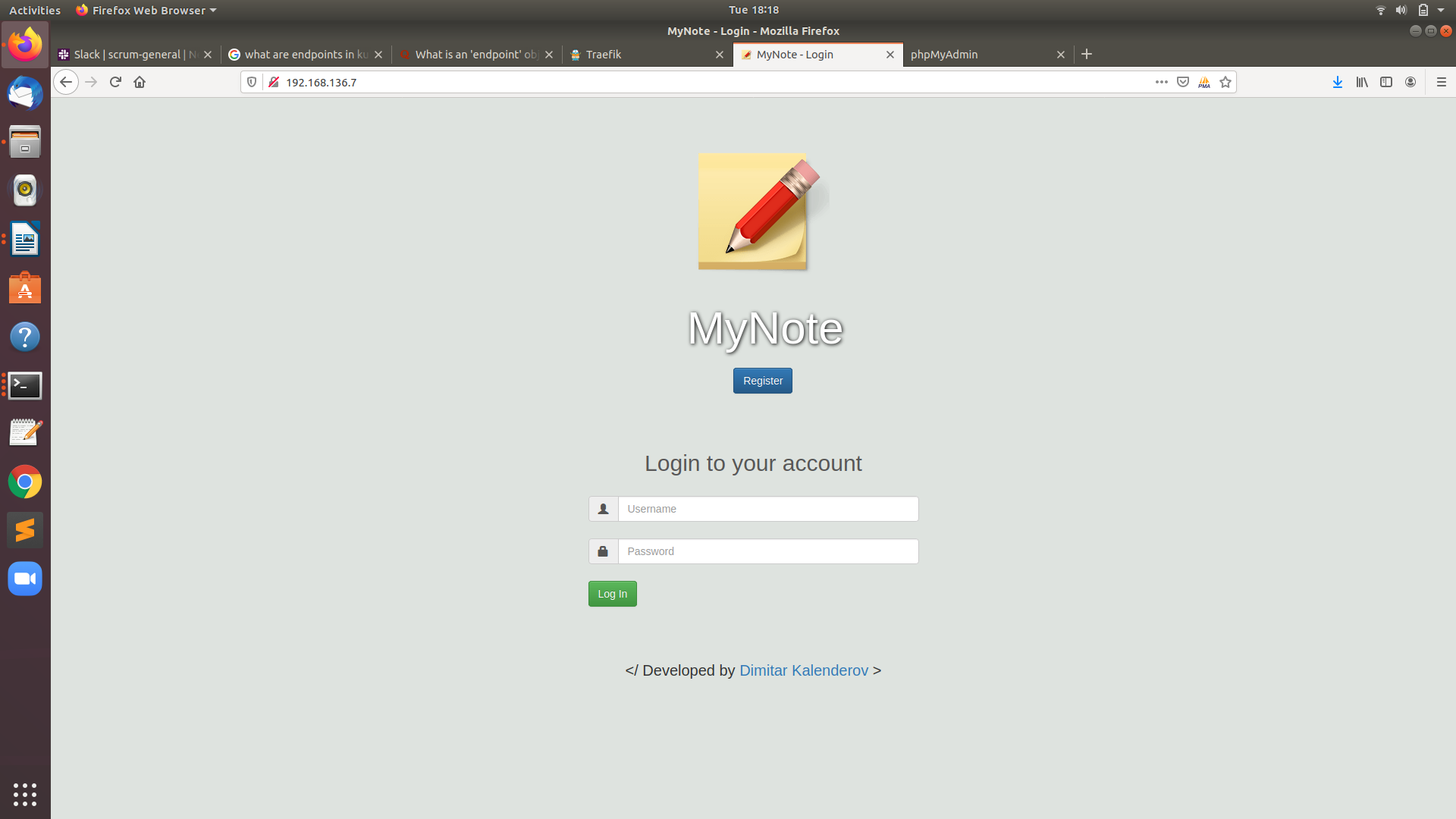
****

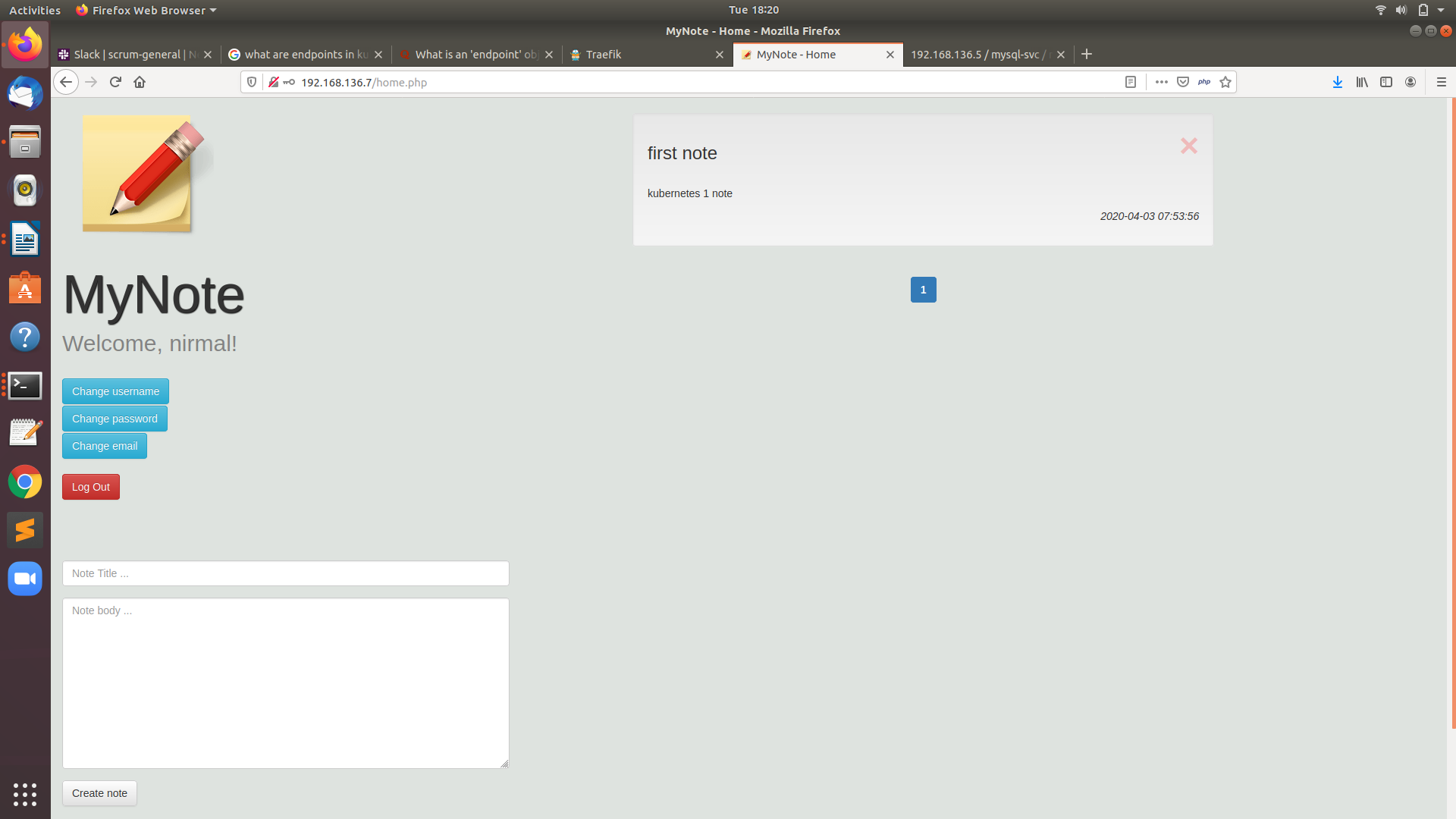
COMMAND - **$ kubectl create -f php-mysql-ingress.yaml**

****

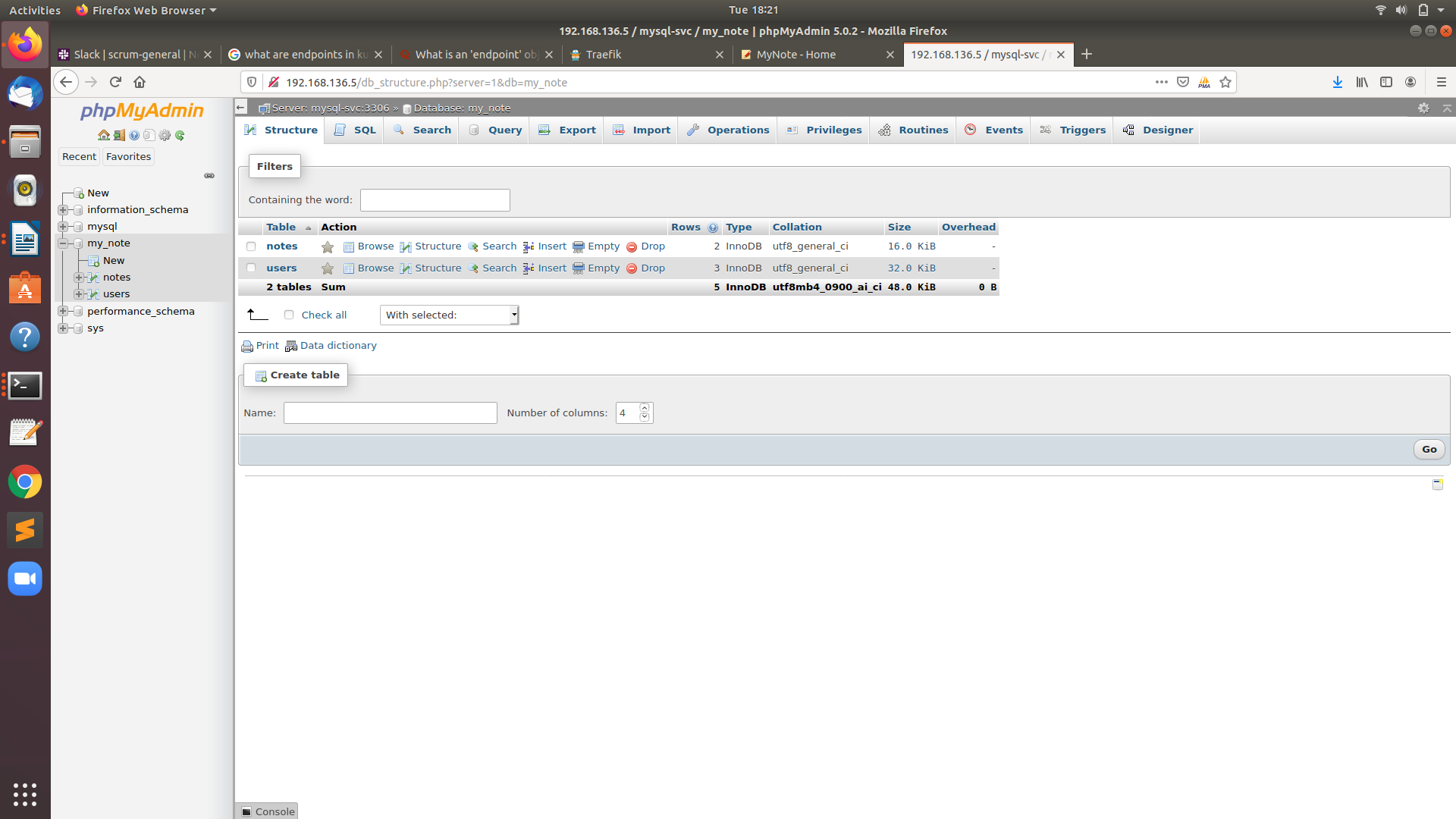
**SCREENSHOTS – traefik-ingress-controller**

**php-app UI screen shots**

****

****

**Mysql Database**

****

**Task to be done via Helm3**

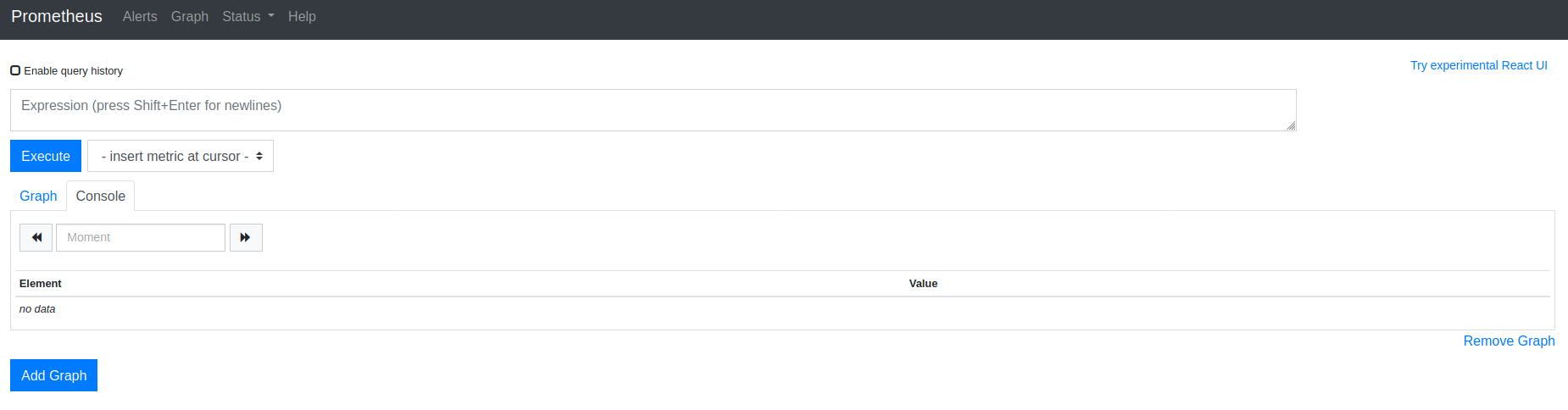
**Step 1 –** Create Persistent volume first.

**Step 2** – create helm chart : **$ helm install promgraf promgraf/ -n nirmal**

**Step 3 -**  Set prometheus at nodeport 32322 and grafana at 32323

Next page consist screenshots of this

**Prometheus**



**grafana**

username – admin

password - nirmal

