In this document I want to share the issues regarding the Ingress and LoadBalancer.

**LoadBalancer**: A LoadBalancer service is the standard way to expose a service to the internet. With this method, each service gets its own IP address.

The desired state was getting minikube ip/vote and minikube ip/result exposed into the network by the LoadBalancer.

Configuration files can be found on branch test.

Ingress used was nginx ingress: https://kubernetes.github.io/ingress-nginx/

Issues:

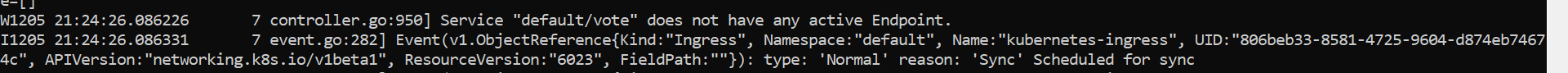
1. Service doesn’t have an endpoint.

Possible fix: Upgrade nginx controller image

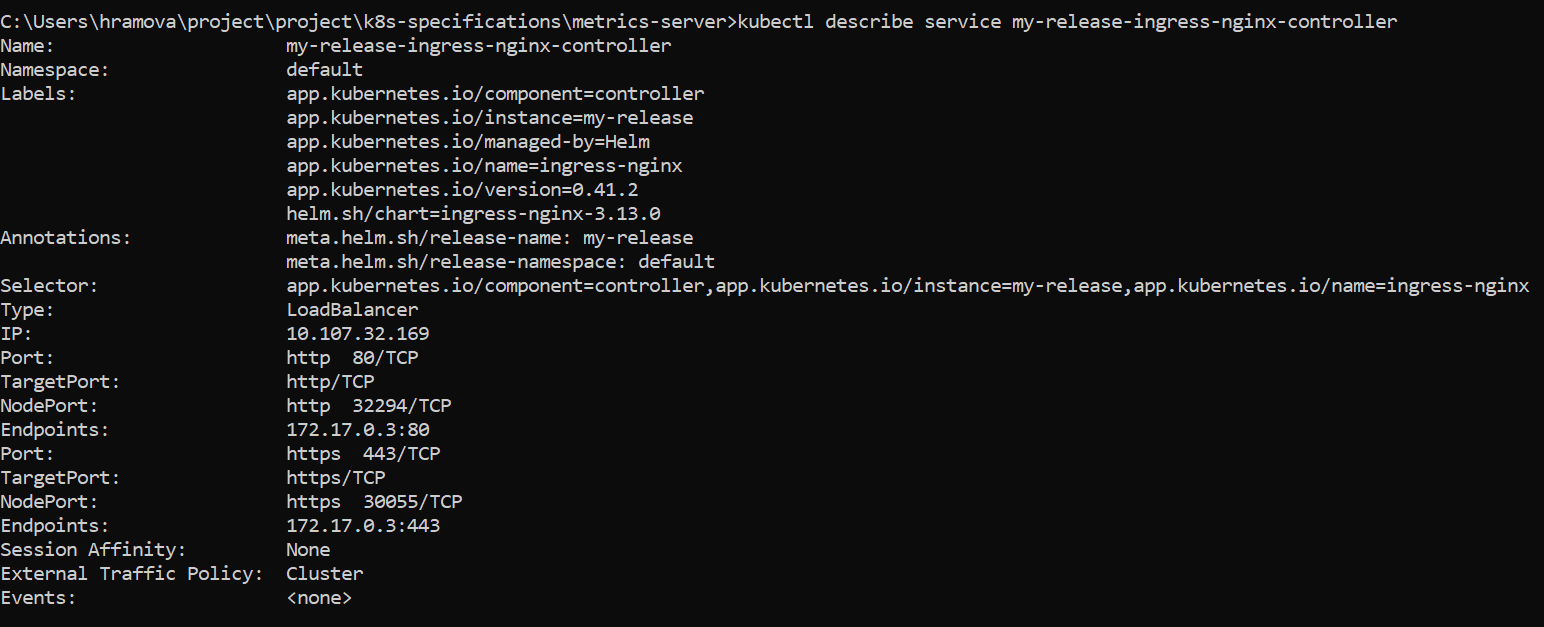
kubectl set image deployment/nginx-ingress-controller \

nginx-ingress-controller=k8s.gcr.io/ingress-nginx/controller:v0.34.1@sha256:0e072dddd1f7f8fc8909a2ca6f65e76c5f0d2fcfb8be47935ae3457e8bbceb20 \

-n ingress-nginx



1. LoadBalancer Ip is pending





Possible fix:

It looks like you are using a custom Kubernetes Cluster (using minikube, kubeadm or the like). In this case, there is no LoadBalancer integrated (unlike AWS or Google Cloud). With this default setup, you can only use [NodePort](https://kubernetes.io/docs/concepts/services-networking/service/" \l "nodeport) or an Ingress Controller.

With the [Ingress Controller](https://kubernetes.io/docs/concepts/services-networking/ingress/" \l "ingress-class) you can setup a domain name which maps to your pod; you don't need to give your Service the LoadBalancer type if you use an Ingress Controller.

<https://minikube.sigs.k8s.io/docs/handbook/accessing/#using-minikube-tunnel>

Services of type LoadBalancer can be exposed via the minikube tunnel command.

minikube tunnel



Fix was not found!

As a result I just did set up the services as **NodePort**:

A NodePort service is the most basic way to get external traffic directly to your service. NodePort, as the name implies, opens a specific port, and any traffic that is sent to this port is forwarded to the service.

