Installeren prometheus:

Voor het monitoren van mijn kubernetes cluster ga ik gebruik maken van prometheus. In het volgende bestand ga ik toelichten welke stappen ik precies heb ondernomen om dit voor elkaar te krijgen.

Het downloaden van de prometheus manifest files

git clone https://github.com/techiescamp/kubernetes-prometheus

Create a namespace & cluster role

1. Kubectl create namespace monitoring
2. Create file name clusterRole.yaml (see K8s folder)
3. Create role
    a. Kubectl create -f clusterRole.yaml

Create a Config Map To Externalize Prometheus Configurations

All configurations for Prometheus are part of `prometheus.yaml` file and all the alert rules for Alertmanager are configured in `prometheus.rules`.

1. `prometheus.yaml`: This is the main Prometheus configuration which holds all the scrape configs, service discovery details, storage locations, data retention configs, etc)
2. `prometheus.rules`: This file contains all the Prometheus alerting rules

1. Create a file called config-map.yaml (see k8s folder for file)
2. Execute file
    a. kubectl create -f config-map.yaml

Create a Prometheus Deployment

1. create a file named Prometheus-deployment.yaml (see k8s for file)
2. kubectl create -f Prometheus-deployment.yaml
    a. `kubectl create -f prometheus-deployment.yaml`
3. `check if the deployment works`
    a. `kubectl get deployments --namespace=monitoring`

Connecting To Prometheus Dashboard

The first method for pc will be port forwarding:

Step 1: First, get the Prometheus pod name.

```
kubectl get pods --namespace=monitoring
```

The output will look like the following.

```
➜  kubectl get pods --namespace=monitoring
NAME                                    READY     STATUS    RESTARTS   AGE
prometheus-monitoring-3331088907-hm5n1   1/1       Running   0          5m
```

Step 2: Execute the following command with your pod name to access Prometheus from localhost port 8080.

> **Note:** Replace prometheus-monitoring-3331088907-hm5n1 with your pod name.

```
kubectl port-forward prometheus-monitoring-3331088907-hm5n1 8080:9090 -n
monitoring
```

Step 3: Now, if you access `http://localhost:8080` on your browser, you will get the

Creating a service:

1. create file Prometheus-service.yaml
    a. kubectl create -f prometheus-service.yaml --namespace=monitoring
    b.

Using an ingress controller

Method 3: Exposing Prometheus Using Ingress

If you have an existing ingress controller setup, you can create an ingress object to route the Prometheus DNS to the Prometheus backend service.

Also, you can add SSL for Prometheus in the ingress layer. You can refer to the Kubernetes ingress TLS/SSL Certificate guide for more details.

Here is a sample ingress object. Please refer to this GitHub link for a sample ingress object with SSL

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: prometheus-ui
  namespace: monitoring
  annotations:
    kubernetes.io/ingress.class: nginx
spec:
  rules:
  # Use the host you used in your kubernetes Ingress Configurations
  - host: prometheus.example.com
    http:
      paths:
      - backend:
          serviceName: prometheus-service
          servicePort: 8080
```

Installing Grafana

1. apply Grafana-datasource-config.yaml
    a. kubectl apply -f `https://github.com/bibinwilson/kubernetes-grafana/grafana-datasource-config.yaml`
2. apply deployment.yaml
    a. kubectl apply -f `https://github.com/bibinwilson/kubernetes-grafana/deployment.yaml`
3. apply service.yaml
    a. kubectl apply -f `https://github.com/bibinwilson/kubernetes-grafana/service.yaml`
    b.