SCGC-Assignment

Miulescu Cristina-Maria, SCPD

https://github.com/cristinamiu/SCGC-Assignment

Let's create a new working directory:

```
mkdir assignment
cd assignment
```

Task 0: Create the Kubernetes cluster

As pointed in the Kubernetes lab, I used the command kind to create the Kubernetes cluster:

```
sudo kind create cluster
```

Task 1: Deploy the nginx service

First let's add the following line to **/etc/hosts** on the host VM in order to give our service a name:

```
172.18.0.2 nginx promexporter
```

Create **task1** directory:

```
mkdir task1
cd task1
```

1.1 Create a ConfigMap for the HTML content

This ConfigMap will display the html for port **80** within the cluster and port **30080** outside the cluster.

nginx-html-configmap.yaml

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: nginx-html
data:
 index.html: |
    <html>
        <body>
                <h1>Hello from SCGC Assignment!</h1>
                <h3>Miulescu Cristina-Maria, SCPD</h3>
        </body>
    </html>
```

1.2 Create a ConfigMap that will take care of the **stub_status** module on port **8080**:

nginx-configmap.yaml

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: nginx-conf
data:
  default.conf: |
    server {
      listen       8080;
      server_name  nginx;


      location / {
        root   /usr/share/nginx/html;
        index  index.html index.htm;
      }

      location /metrics {
        stub_status;
        allow 1.1.1.1;
      }
    }
```

1.3 Create the Deployment:

This deployment will have 2 containters with nginx image. The first one will have the customized html file and the second one is for the stub_status metrics.

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx
  labels:
    app: nginx
spec:
  replicas: 1
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx-html
        image: nginx:latest
        ports:
        - containerPort: 80
        volumeMounts:
        - name: nginx-html-vol
          mountPath: "/usr/share/nginx/html/index.html"
          subPath: "index.html"
      - name: nginx-metrics
        image: nginx:latest
        ports:
        - containerPort: 8080
        volumeMounts:
        - name: nginx-conf-vol
          mountPath: "/etc/nginx/conf.d/default.conf"
          subPath: "default.conf"
      volumes:
      - name: nginx-conf-vol
        configMap:
          name: nginx-conf
          items:
          - key: "default.conf"
            path: "default.conf"
      - name: nginx-html-vol
        configMap:
          name: nginx-html
          items:
          - key: "index.html"
            path: "index.html"
```

1.4 Create the service file:

nginx-service.yaml

```
apiVersion: v1
kind: Service
metadata:
  name: nginx
spec:
  type: NodePort
  selector:
    app: nginx
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
      nodePort: 30080
      name: "html"
    - protocol: TCP
      port: 8080
      targetPort: 8080
      nodePort: 30088
      name: "metrics"
```

1.5 Apply the configuration files

```
sudo kubectl apply -f nginx-html-configmap.yaml
sudo kubectl apply -f nginx-configmap.yaml
sudo kubectl apply -f nginx-deployment.yaml
sudo kubectl apply -f nginx-service.yaml
```

Output:

The terminal output looks as below:

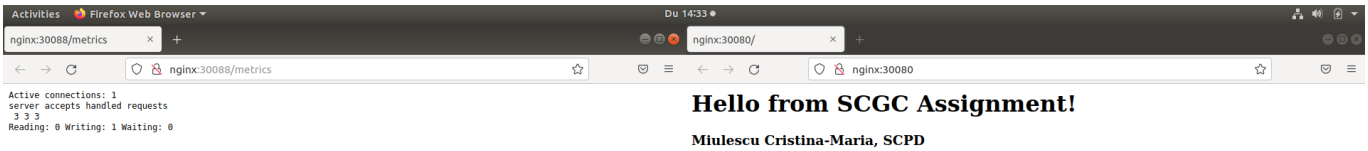
```
student@scgc:~/scgc/assignment/task1$ sudo kubectl get pods
NAME                     READY   STATUS    RESTARTS   AGE
nginx-c9bb45656-b5z5z    2/2     Running   0          24s
student@scgc:~/scgc/assignment/task1$ sudo kubectl get svc
NAME         TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)                       AGE
kubernetes   ClusterIP   10.96.0.1       <none>        443/TCP                       2m21s
nginx        NodePort    10.96.198.183   <none>        80:30080/TCP,8080:30088/TCP   23s
student@scgc:~/scgc/assignment/task1$ curl http://nginx:30080
<html>
    <body>
            <h1>Hello from SCGC Assignment!</h1>
            <h3>Miulescu Cristina-Maria, SCPD</h3>
    </body>
</html>
student@scgc:~/scgc/assignment/task1$ curl http://nginx:30088
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
student@scgc:~/scgc/assignment/task1$ curl http://nginx:30088/metrics
Active connections: 1
server accepts handled requests
 2 2 2
Reading: 0 Writing: 1 Waiting: 0
```

Also, in the browser:

- left side: http://nginx:30088/metrics
- right side: http://nginx:30080



Task 2: Deploy Prometheus

Next we will go to the **/task2** directory:

```
mkdir task2
cd task2
```

2.1 Create a ConfigMap for promexporter

prom-configmap.yaml

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: prom-conf
data:
  default.conf: |
    server {
      listen       80;
      server_name  promexporter;


      location / {
        root   /usr/share/nginx/html;
        index  index.html index.htm;
      }
    }
```

2.2 Create Deployment for promexporter

The promexporter will get its metrics info from http://nginx:8080/metrics :

prom-deployment.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: promexporter
  labels:
    app: promexporter
spec:
  replicas: 1
  selector:
    matchLabels:
      app: promexporter
  template:
    metadata:
      labels:
        app: promexporter
    spec:
      containers:
      - name: nginx-container
        image: nginx:latest
        ports:
        - containerPort: 80
        volumeMounts:
        - name: prom-conf-vol
          mountPath: "/etc/nginx/conf.d/default.conf"
          subPath: "default.conf"
      - name: promexporter-container
        image: nginx/nginx-prometheus-exporter:0.8.0
        args: ["-nginx.scrape-uri", "http://nginx:8080/metrics"]
        ports:
        - containerPort: 9113
      volumes:
      - name: prom-conf-vol
        configMap:
          name: prom-conf
          items:
          - key: "default.conf"
            path: "default.conf"
```

2.3 Create Service for promexporter

The promexporter will have port **9113** within the cluster and port **30081** outside the cluster:

prom-service.yaml

```
apiVersion: v1
kind: Service
metadata:
  name: promexporter
spec:
  type: NodePort
  selector:
    app: promexporter
  ports:
    - protocol: TCP
      port: 9113
      targetPort: 9113
      nodePort: 30081
      name: "html"
```

2.4 Apply the files:

```
sudo kubectl apply -f prom-configmap.yaml
sudo kubectl apply -f prom-deployment.yaml
sudo kubectl apply -f prom-service.yaml
```

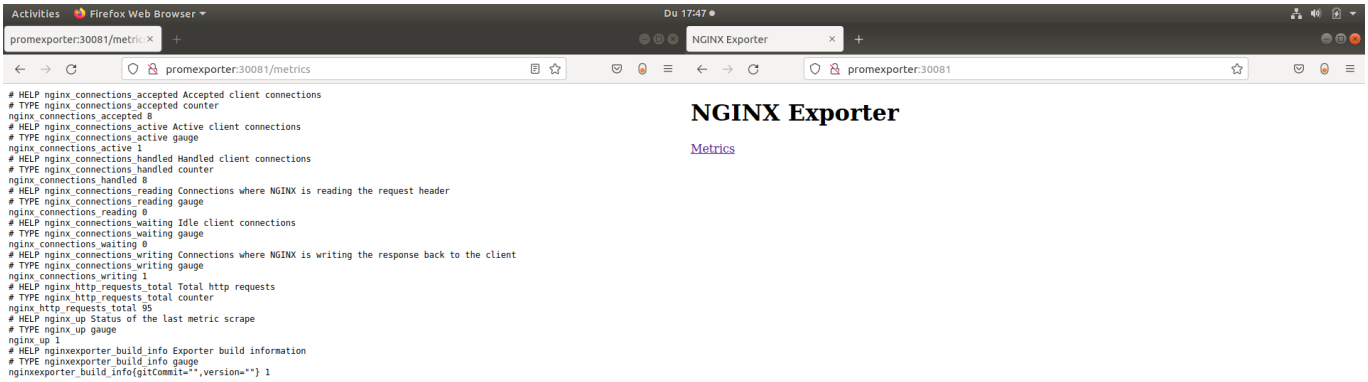
Output:

The terminal output:

```
student@scgc:~/scgc/assignment/task2$ sudo kubectl get pods
NAME                           READY   STATUS    RESTARTS   AGE
nginx-c9bb45656-b5z5z          2/2     Running   0          27m
promexporter-7796c99ff6-flc5k  2/2     Running   0          5m21s
student@scgc:~/scgc/assignment/task2$ sudo kubectl get svc
NAME           TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)                        AGE
kubernetes     ClusterIP   10.96.0.1       <none>        443/TCP                        29m
nginx          NodePort    10.96.198.183   <none>        80:30080/TCP,8080:30088/TCP    27m
promexporter   NodePort    10.96.109.16    <none>        9113:30081/TCP                 5m18s
student@scgc:~/scgc/assignment/task2$ curl http://promexporter:30081
<!DOCTYPE html>
                    <title>NGINX Exporter</title>
                    <h1>NGINX Exporter</h1>
                    <p><a href="/metrics">Metrics</a></p>student@scgc:~/scgc/assignment/task2$
student@scgc:~/scgc/assignment/task2$ curl http://promexporter:30081/metrics
# HELP nginx_connections_accepted Accepted client connections
# TYPE nginx_connections_accepted counter
nginx_connections_accepted 5
# HELP nginx_connections_active Active client connections
# TYPE nginx_connections_active gauge
nginx_connections_active 1
# HELP nginx_connections_handled Handled client connections
# TYPE nginx_connections_handled counter
nginx_connections_handled 5
# HELP nginx_connections_reading Connections where NGINX is reading the request header
# TYPE nginx_connections_reading gauge
nginx_connections_reading 0
# HELP nginx_connections_waiting Idle client connections
# TYPE nginx_connections_waiting gauge
nginx_connections_waiting 0
# HELP nginx_connections_writing Connections where NGINX is writing the response back to the client
# TYPE nginx_connections_writing gauge
nginx_connections_writing 1
# HELP nginx_http_requests_total Total http requests
# TYPE nginx_http_requests_total counter
nginx_http_requests_total 7
# HELP nginx_up Status of the last metric scrape
# TYPE nginx_up gauge
nginx_up 1
# HELP nginxexporter_build_info Exporter build information
# TYPE nginxexporter_build_info gauge
nginxexporter_build_info{gitCommit="",version=""} 1
student@scgc:~/scgc/assignment/task2$
```

The browser output:

- left side: http://promexporter:30081/metrics
- right side: http://promexporter:30081



Task 3:

3.1 Create **monitoring** namespace:

```
sudo kubectl create namespace monitoring
```

3.2 Install prometheus using helm

```
$ sudo helm repo add prometheus-community https://prometheus-
community.github.io/helm-charts
$ sudo helm install prometheus prometheus-community/prometheus -n monitoring
```

3.3 Port-forrward

```
sudo kubectl -n monitoring port-forward services/prometheus-server 30082:80
```

3.4 Open a new terminal and edit the configMap of prometheus-server:

```
sudo kubectl -n monitoring edit cm prometheus-server
```

Add this line:

```
- job_name: prometheus
      static_configs:
      - targets:
        - localhost:9090
    # Add this
    - job_name: promexporter
      static_configs:
      - targets:
        - promexporter:30081
```

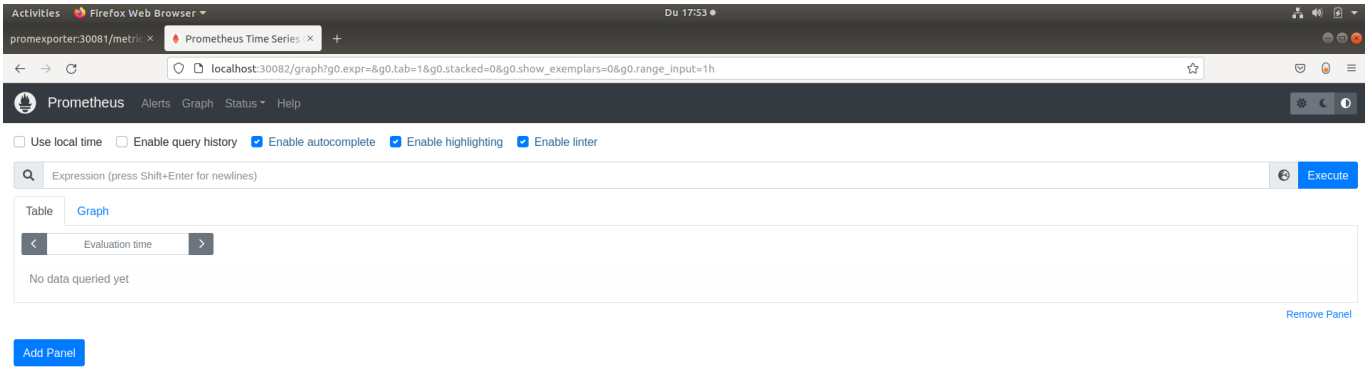
Note: I also tried with *promexporter.default.svc.cluster.local:9113* as indicated but it did not show the metrics when accessing the link.

3.5 Test

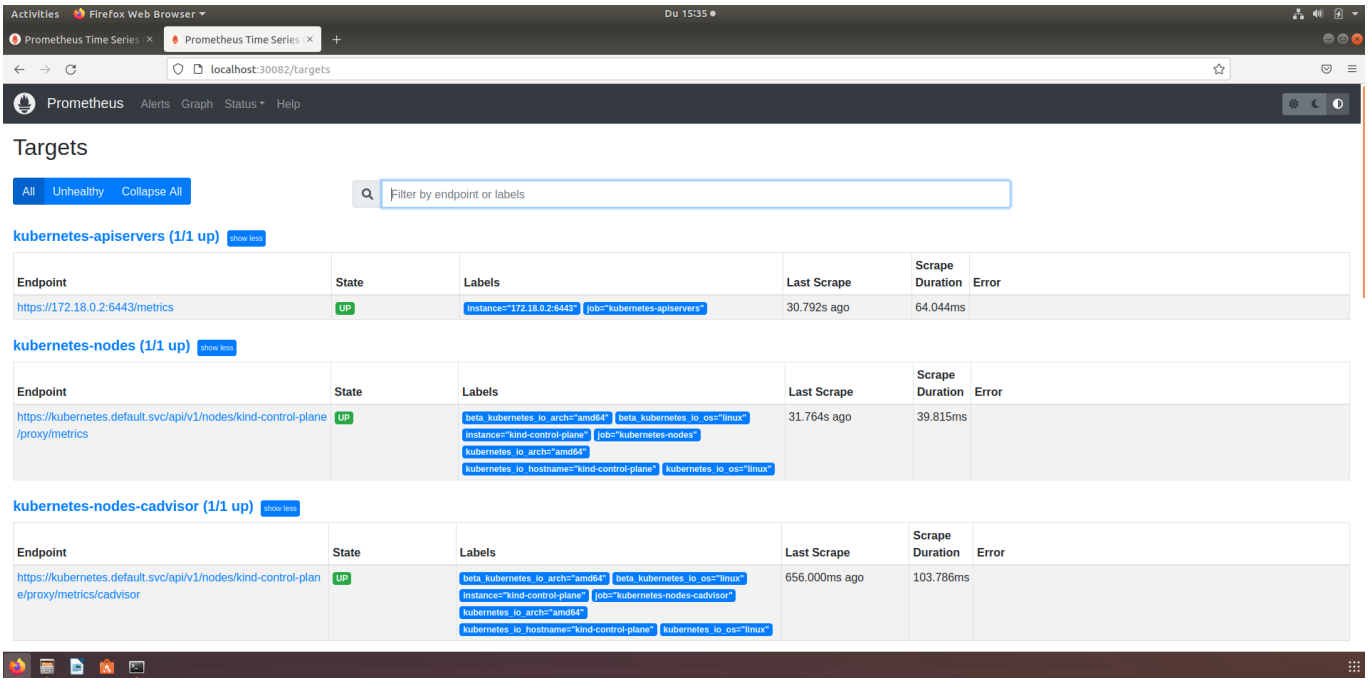
```
curl http://localhost:30082
```

Output:

Going to http://localhost:30082, you shold see the following:



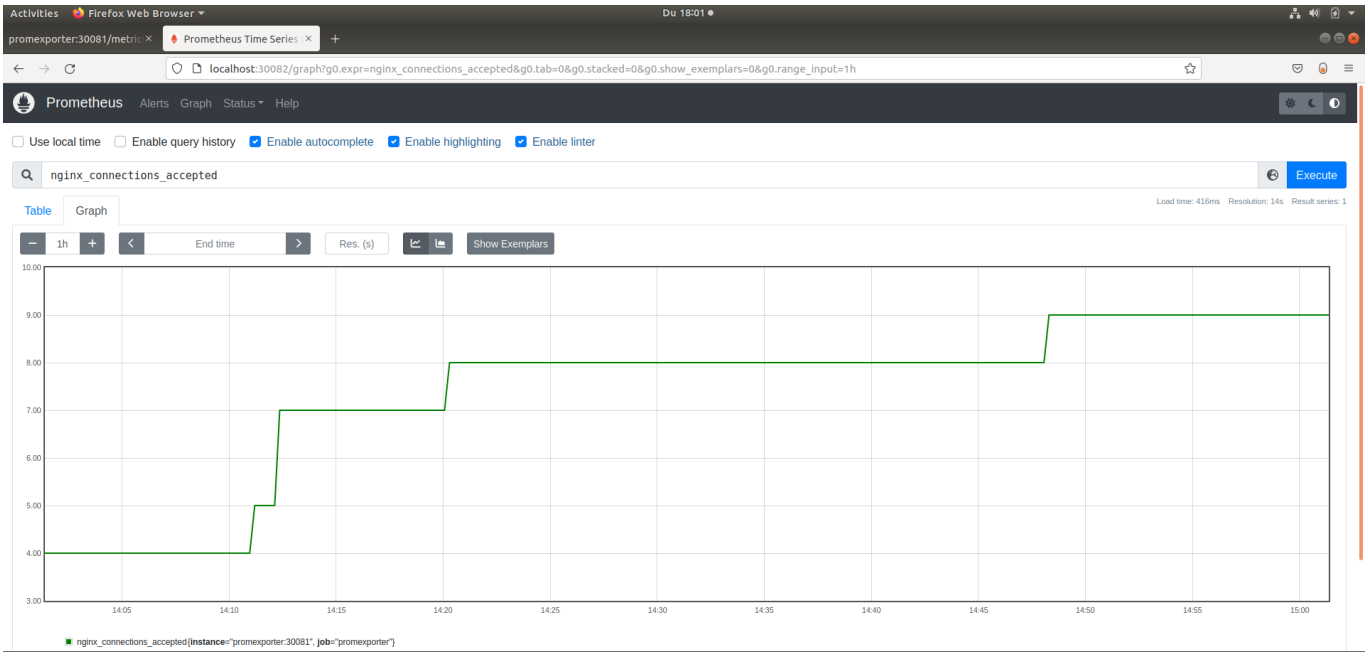
Under **/targets** :



We can find under **/targets** our **promexporter**:

Query a metric:

Also, go to **Graph** and query a metric, like “nginx_connections_accepted”:



Task 4: Grafana

4.1 Install Grafana

Next, I installed Grafana using helm chart on the **monitoring** namespace:

```
sudo helm repo add bitnami https://charts.bitnami.com/bitnami
sudo helm install grafana bitnami/grafana
```

```
student@scgc:~/scgc/assignment/task3$ sudo helm install grafana bitnami/grafana -n monitoring
[sudo] password for student:
NAME: grafana
LAST DEPLOYED: Sun May 29 15:51:05 2022
NAMESPACE: monitoring
STATUS: deployed
REVISION: 1
TEST SUITE: None
NOTES:
CHART NAME: grafana
CHART VERSION: 7.8.1
APP VERSION: 8.5.3

** Please be patient while the chart is being deployed **

1. Get the application URL by running these commands:
    echo "Browse to http://127.0.0.1:8080"
    kubectl port-forward svc/grafana 8080:3000 &

2. Get the admin credentials:

    echo "User: admin"
    echo "Password: $(kubectl get secret grafana-admin --namespace monitoring -o jsonpath="{.data.GF_SECURITY_ADMIN_PASSWORD}" | base64 --decode)"
student@scgc:~/scgc/assignment/task3$ sudo kubectl get svc
NAME          TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)                      AGE
kubernetes    ClusterIP   10.96.0.1       <none>        443/TCP                      84m
nginx         NodePort    10.96.198.183   <none>        80:30080/TCP,8080:30088/TCP  82m
promexporter  NodePort    10.96.109.16    <none>        9113:30081/TCP               60m
student@scgc:~/scgc/assignment/task3$ sudo kubectl -n monitoring get svc
NAME                            TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)    AGE
grafana                         ClusterIP   10.96.71.201    <none>        3000/TCP   94s
prometheus-alertmanager         ClusterIP   10.96.209.141   <none>        80/TCP     49m
prometheus-kube-state-metrics   ClusterIP   10.96.64.17     <none>        8080/TCP   49m
prometheus-node-exporter        ClusterIP   10.96.119.89    <none>        9100/TCP   49m
prometheus-pushgateway          ClusterIP   10.96.92.43     <none>        9091/TCP   49m
prometheus-server               ClusterIP   10.96.163.183   <none>        80/TCP     49m
student@scgc:~/scgc/assignment/task3$
```

4.2 Port-forward Grafana

```
sudo kubectl -n monitoring port-forward svc/grafana 30085:3000
```

4.3 Grafana UI

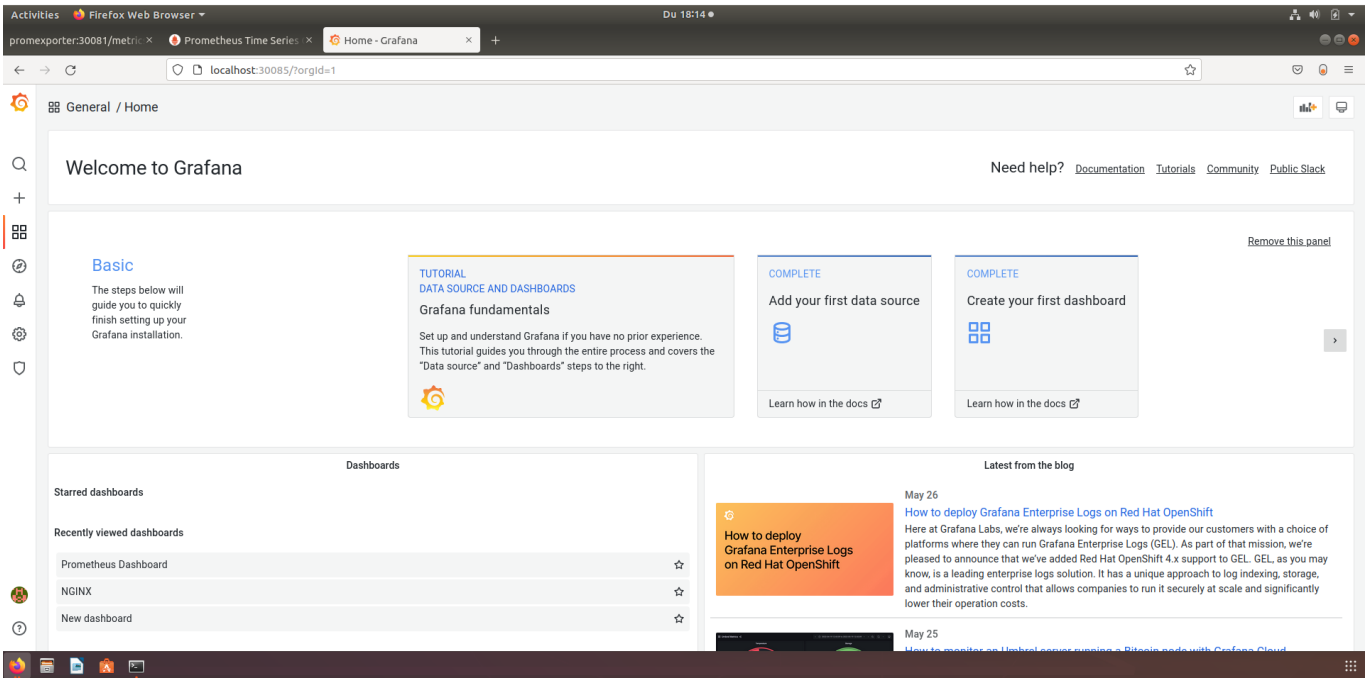
We can access Grafana UI on http://localhost:30085, the port we forwarded to.

First we have to find out the password in order to login:

```
student@scgc:~$ echo "Password: $(sudo kubectl get secret grafana-admin --
namespace monitoring -o jsonpath="{.data.GF_SECURITY_ADMIN_PASSWORD}" | base64 --
decode)"
[sudo] password for student:
Password: 2kOvyXZklL
```

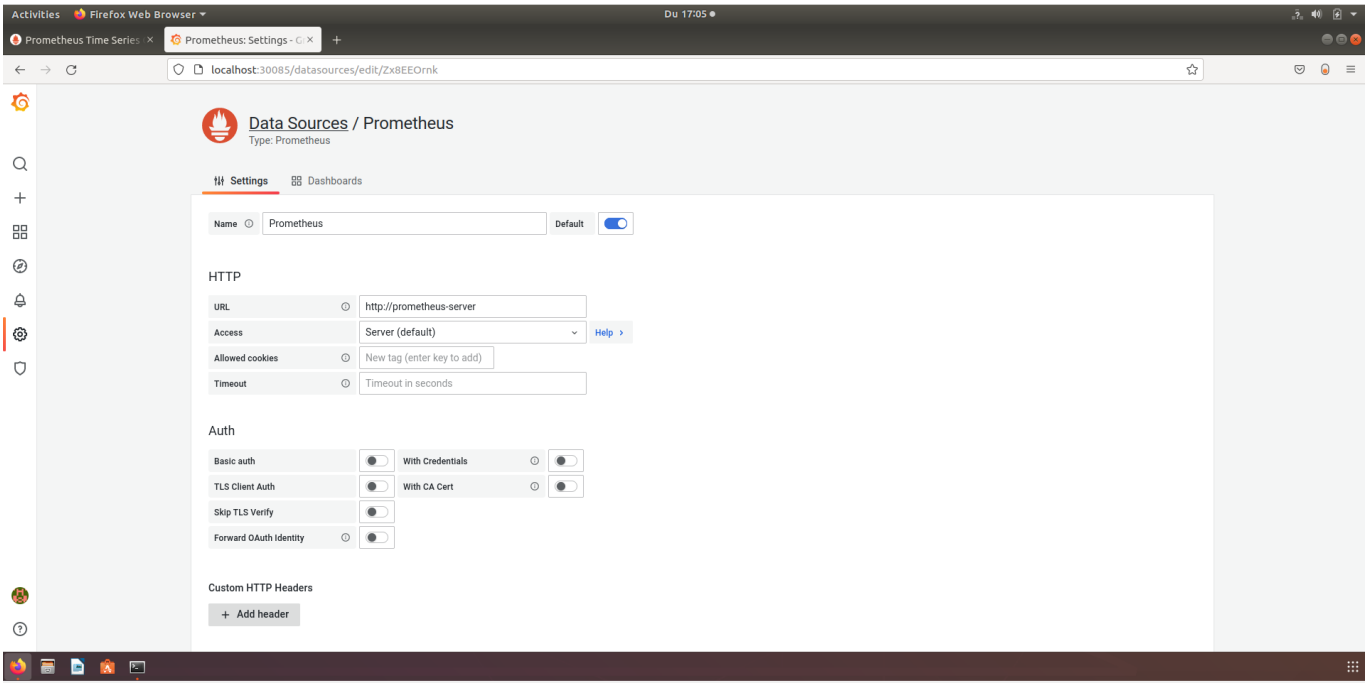
So the credentials are:

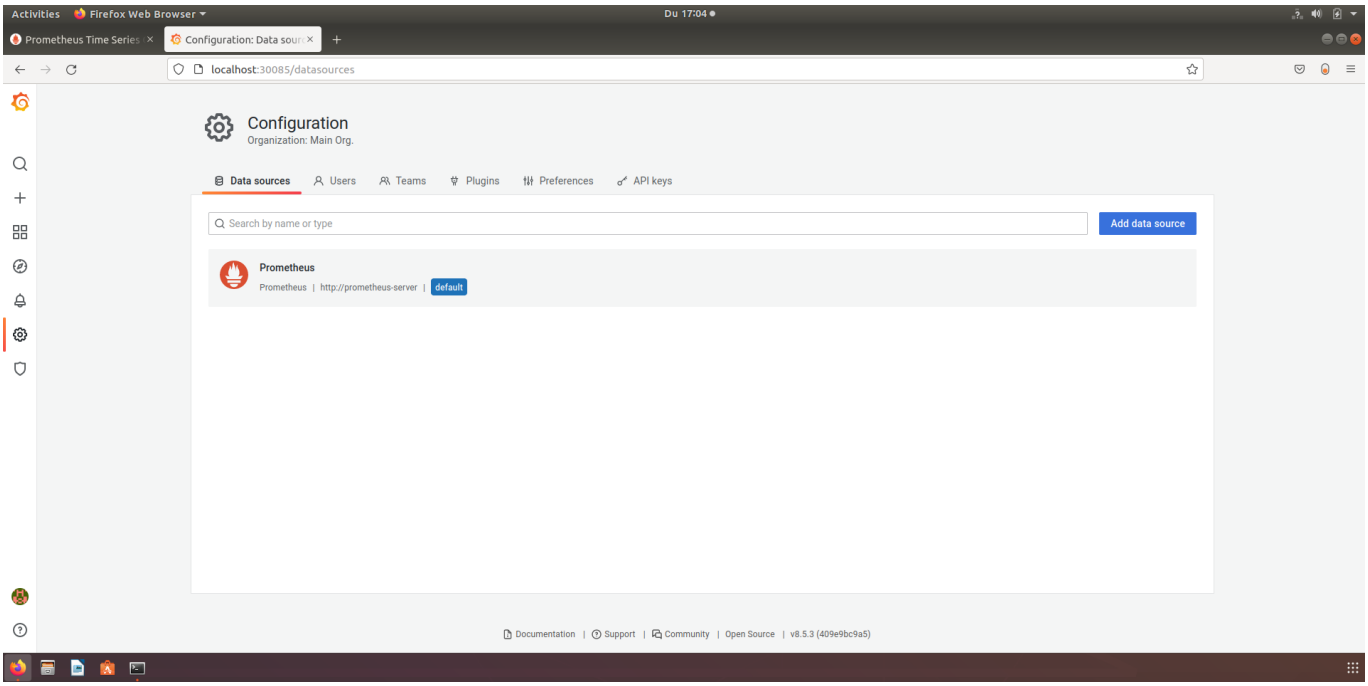
- Username: admin
- Password: 2kOvyXZklL



4.4 Create Prometheus DataSource

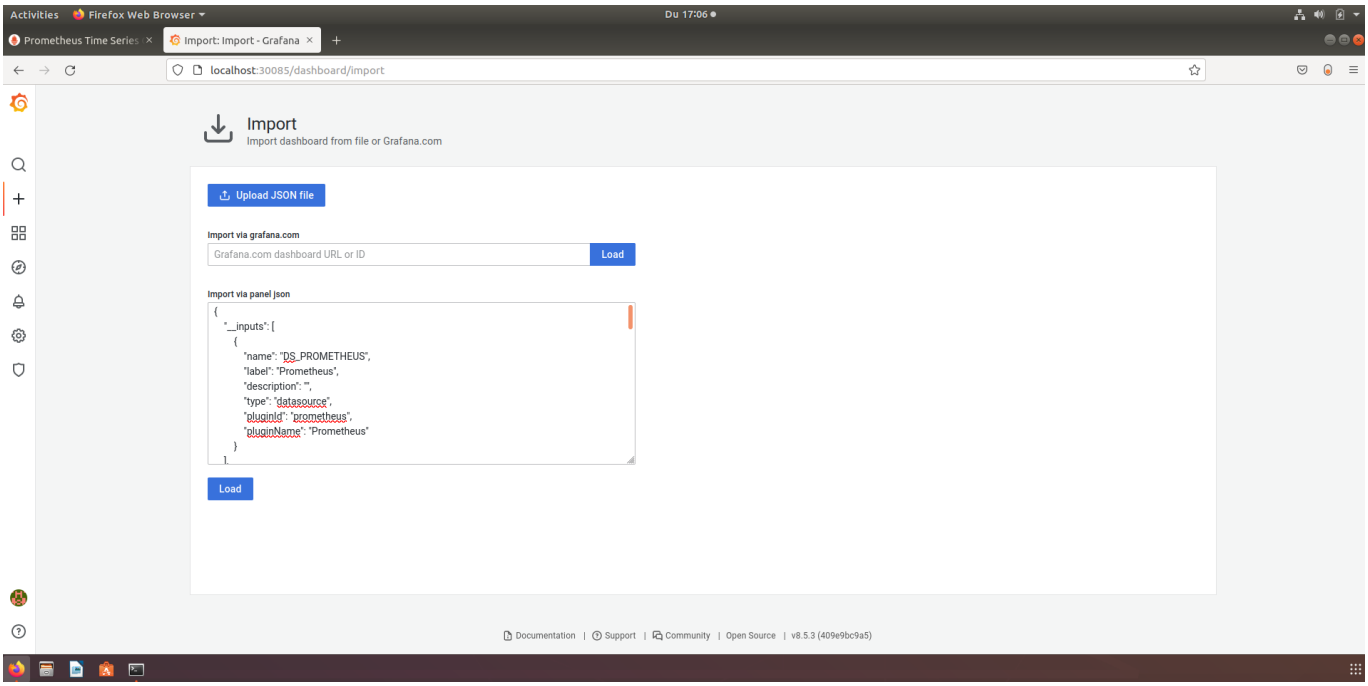
The only field modified is the URL field: http://prometheus-server .

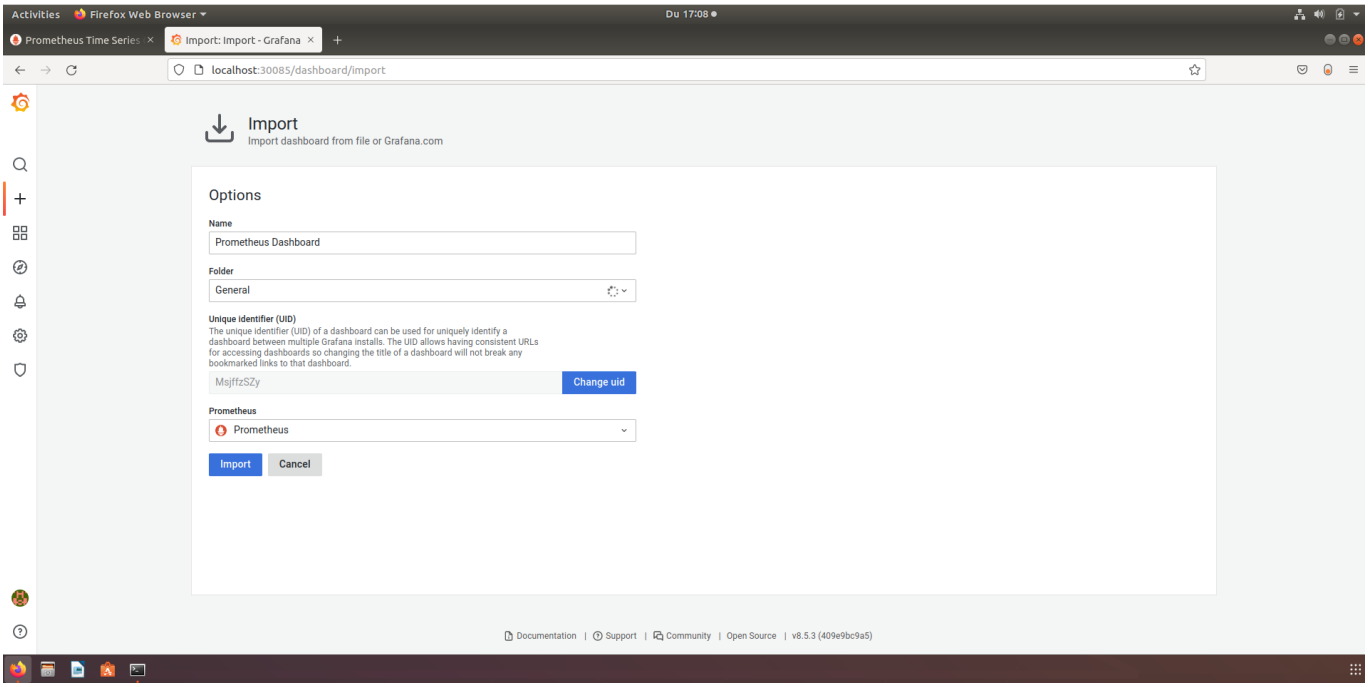




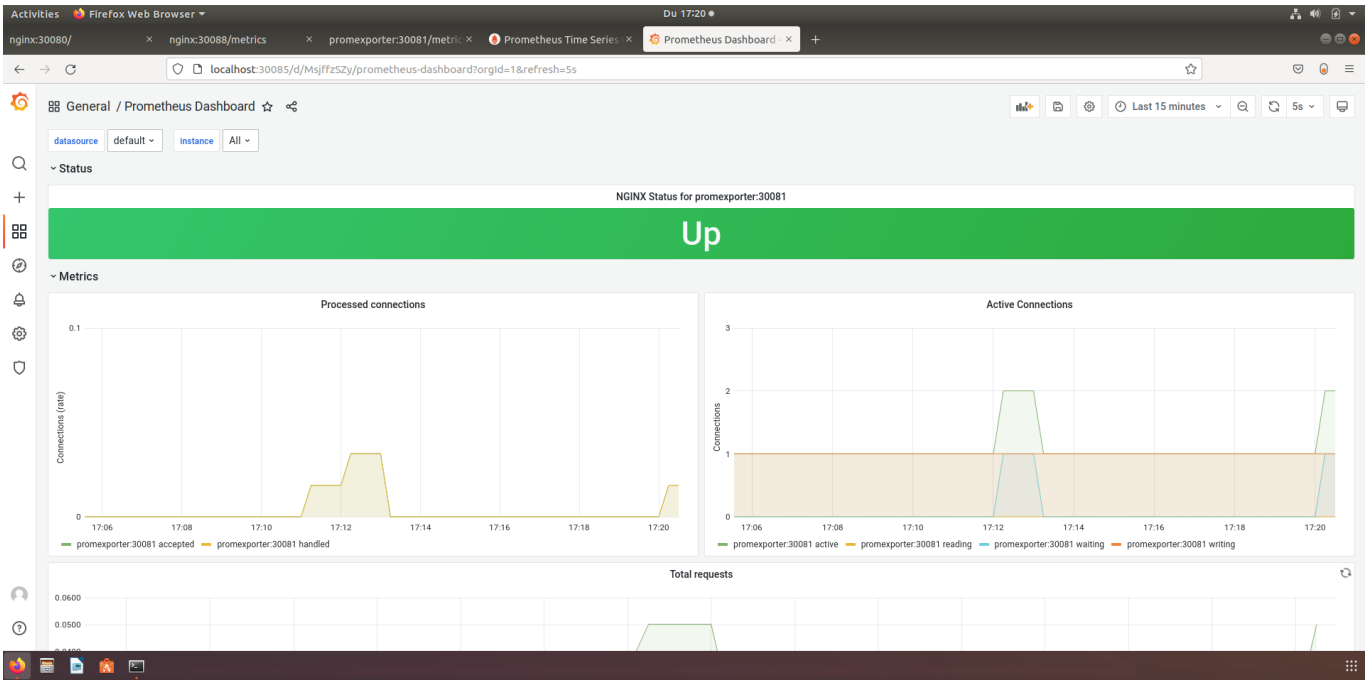
4.5 Create Prometheus Dashboard:

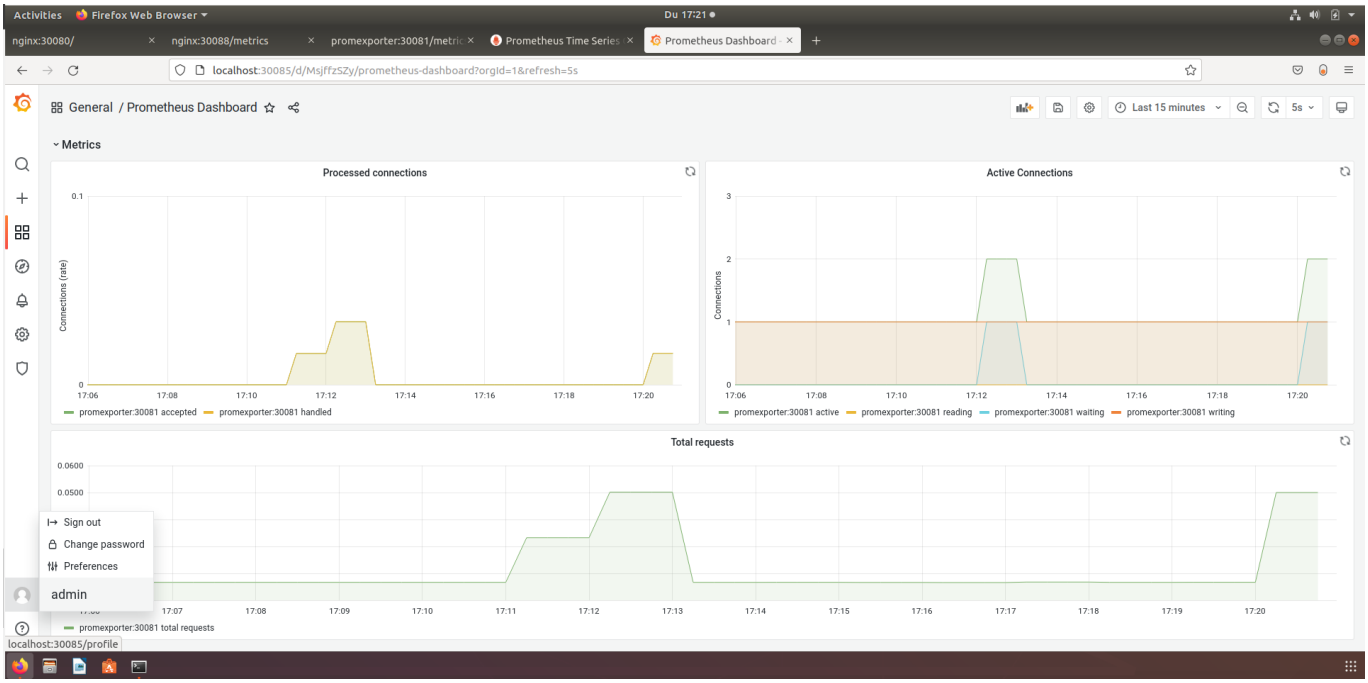
I copied the content from https://github.com/nginxinc/nginx-prometheus-exporter/blob/main/grafana/dashboard.json .


Activities
Firefox Web Browser
Du 17:06
Prometheus Time Series
Import: Import - Grafana
localhost:30085/dashboard/import
Import
Import dashboard from file or Grafana.com
Upload JSON file
Import via grafana.com
Grafana.com dashboard URL or ID
Load
Import via panel json
{
"__inputs": [
{
"name": "DS_PROMETHEUS",
"label": "Prometheus",
"description": "",
"type": "datasource",
"pluginId": "prometheus",
"pluginName": "Prometheus"
}
],
Load
Documentation | Support | Community | Open Source | v8.5.3 (409e9bc9a5)


Activities
Firefox Web Browser
Du 17:08
Prometheus Time Series
Import: Import - Grafana
localhost:30085/dashboard/import
Import
Import dashboard from file or Grafana.com
Options
Name
Prometheus Dashboard
Folder
General
Unique identifier (UID)
The unique identifier (UID) of a dashboard can be used for uniquely identify a dashboard between multiple Grafana installs. The UID allows having consistent URLs for accessing dashboards so changing the title of a dashboard will not break any bookmarked links to that dashboard.
MsjffzSZy
Change uid
Prometheus
Prometheus
Import
Cancel
Documentation | Support | Community | Open Source | v8.5.3 (409e9bc9a5)

Output


Activities
Firefox Web Browser
Du 17:20
nginx:30080/
nginx:30088/metrics
promexporter:30081/metric
Prometheus Time Series
Prometheus Dashboard
localhost:30085/d/MsjffzSZy/prometheus-dashboard?orgId=1&refresh=5s
General / Prometheus Dashboard
Last 15 minutes
5s
datasource
default
instance
All
Status
NGINX Status for promexporter:30081
Up
Metrics
Processed connections
Connections (rate)
promexporter:30081 accepted
promexporter:30081 handled
Active Connections
Connections
promexporter:30081 active
promexporter:30081 reading
promexporter:30081 waiting
promexporter:30081 writing
Total requests


Activities
Firefox Web Browser
Du 17:21
nginx:30080/
nginx:30088/metrics
promexporter:30081/metric
Prometheus Time Series
Prometheus Dashboard
localhost:30085/d/MsjffzSZy/prometheus-dashboard?orgId=1&refresh=5s
General / Prometheus Dashboard
Last 15 minutes
5s
Metrics
Processed connections
Connections (rate)
promexporter:30081 accepted
promexporter:30081 handled
Active Connections
Connections
promexporter:30081 active
promexporter:30081 reading
promexporter:30081 waiting
promexporter:30081 writing
Total requests
Sign out
Change password
Preferences
admin
promexporter:30081 total requests
localhost:30085/profile