Installing And Setting up Grafana

This lesson focuses on the Installation and setting up part of Grafana.

WE'LL COVER THE FOLLOWING

- Install and set-up `Grafana`
  - Login into `Grafana`
  - Set `Prometheus` as a data source

Install and set-up `Grafana`

You probably know what's coming next. We Google "Grafana Helm" and hope that the community already created a Chart we can use. I'll save you from the search by revealing that there is `Grafana` in Helm's *stable* channel. All we have to do is inspect the values and choose which ones we'll use.

```
helm inspect values stable/grafana
```

I won't go through all the values we could use. I assume that, by now, you are a Helm ninja and that you can explore them yourself. Instead, we'll use the values I already defined.

```
cat mon/grafana-values-bare.yml
```

The **output** is as follows.

```
ingress:
  enabled: true
persistence:
  enabled: true
  accessModes:
  - ReadWriteOnce
  size: 1Gi
resources:
```

```
  limits:
    cpu: 20m
    memory: 50Mi
  requests:
    cpu: 5m
    memory: 25Mi
```

There's nothing special about those values. We enabled Ingress, we set `persistence`, and we defined the `resources`. As the name of the file indicates, it's a very bare setup without anything fluffy.

All that's left is to install the Chart.

```
GRAFANA_ADDR="grafana.$LB_IP.nip.io"

helm install grafana \
    stable/grafana \
    --namespace metrics \
    --version 4.1.3 \
    --set ingress.hosts="{$GRAFANA_ADDR}" \
    --values mon/grafana-values-bare.yml

kubectl -n metrics \
    rollout status deployment grafana
```

Now we can open `Grafana` in your favorite browser.

```
open "http://$GRAFANA_ADDR"
```

Login into `Grafana`

You are presented with the login screen. Just as with many other Helm Charts, the installation comes with the `admin` user and the password stored as a Secret.

```
kubectl -n metrics \
    get secret grafana \
    -o jsonpath="{.data.admin-password}" \
    | base64 --decode; echo
```

Please go back to the `Grafana` login screen, type *admin* as the *username*, and paste the output of the previous command as the *password*.

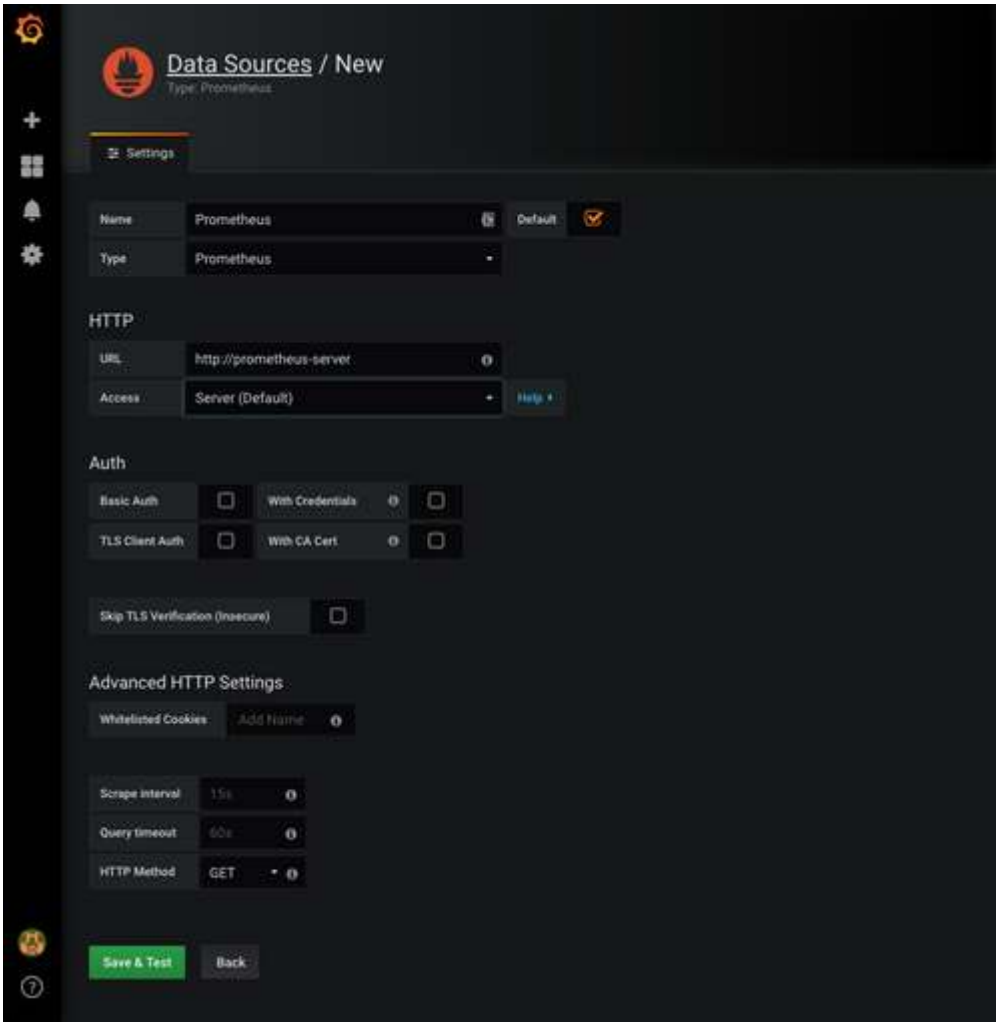
Set `Prometheus` as a data source

`Grafana` does not collect metrics. Instead, it uses other sources of data, so our first action is to set `Prometheus` as the data source.

Please click the *Add data source* icon and click the *Select* button inside *Prometheus*.

Type `Prometheus` as the *Name*. We'll let `Grafana` connect to it through the Kubernetes Service `prometheus-server`. Since both are in the same Namespace, the *URL* should be set to `http://prometheus-server`. All that's left is to *Save & Test*.

> 🔍 The outputs and screenshots in this chapter are taken from Docker For Desktop. There might be slight differences between what you see here and what you can observe on your screen.



Grafana's new datasource screen

Q

`Grafana` doesn't use other sources of data. Instead, it collects metrics, so our first action is to set `Prometheus` as the data source.

COMPLETED 0%

1 of 1

> 🔍 We'll have more screenshots in this chapter than usual. I believe that they will help you replicate the steps we'll discuss.

In the next lesson, we will see how to import and customize the pre-made dashboards.