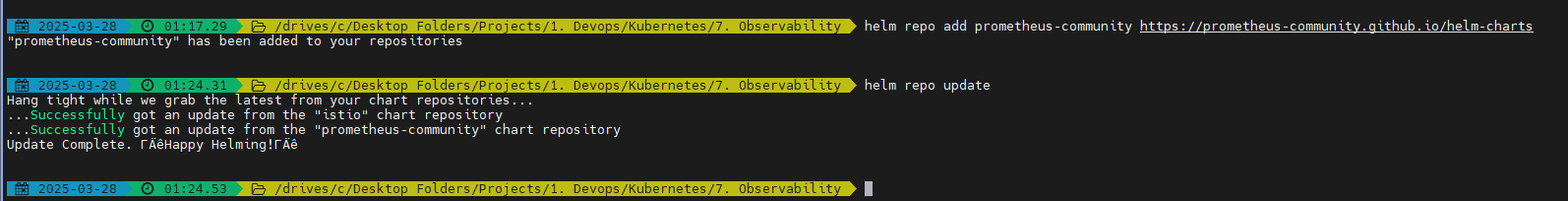
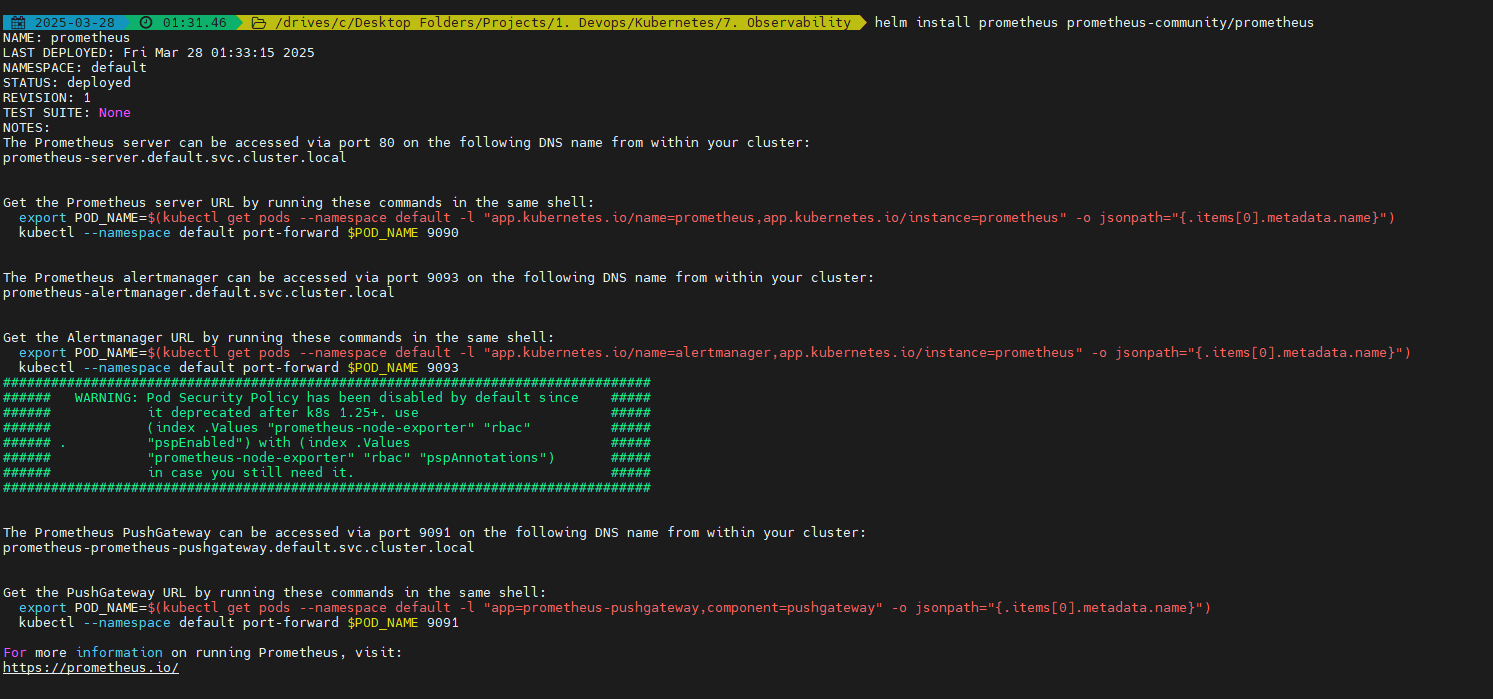
Install Prometheus:

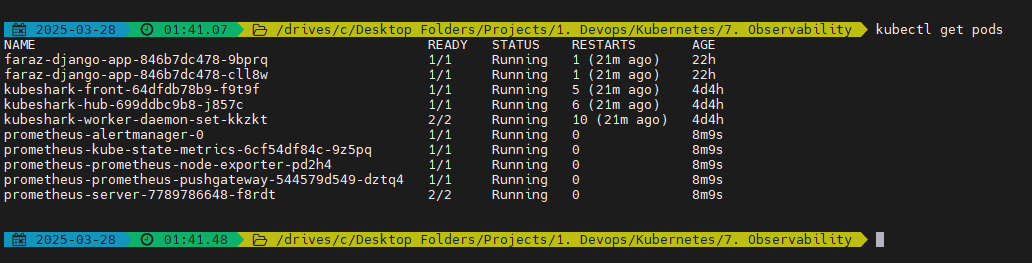
* helm repo add prometheus-community <https://prometheus-community.github.io/helm-charts>
* helm repo update

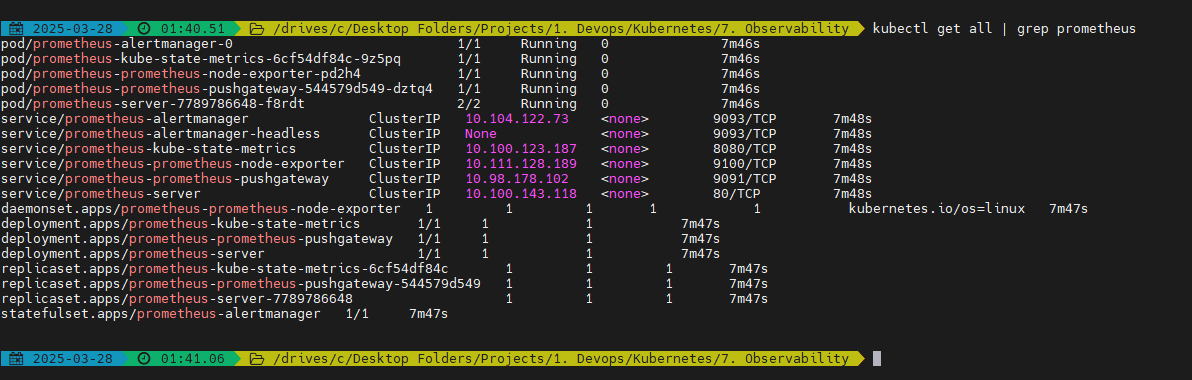


* helm install prometheus prometheus-community/prometheus



Verify all the pods are running:

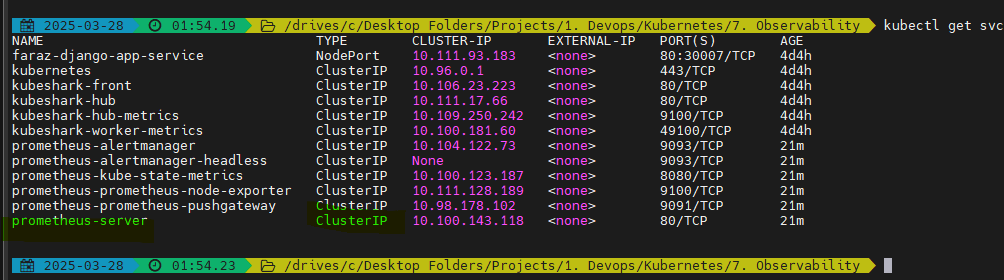




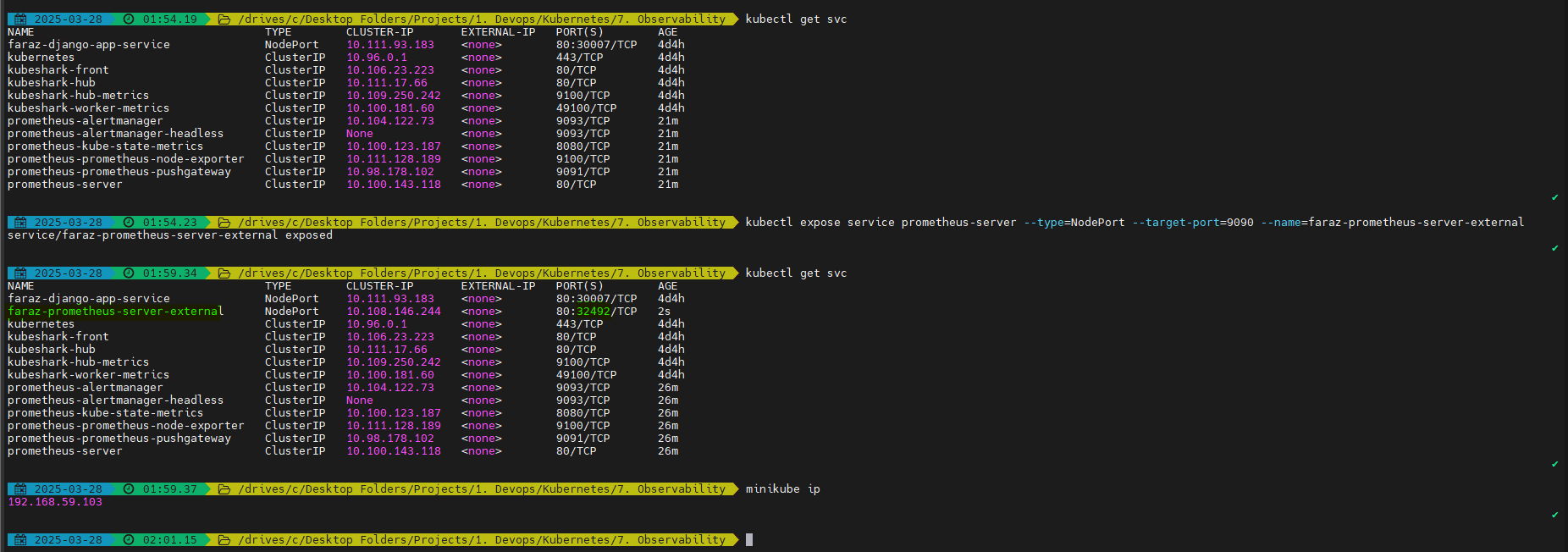
Now as you can see the all the services are by default created using cluster IP mode.

If we want to expose this we may edit it to NodePort IP or Load balancer mode

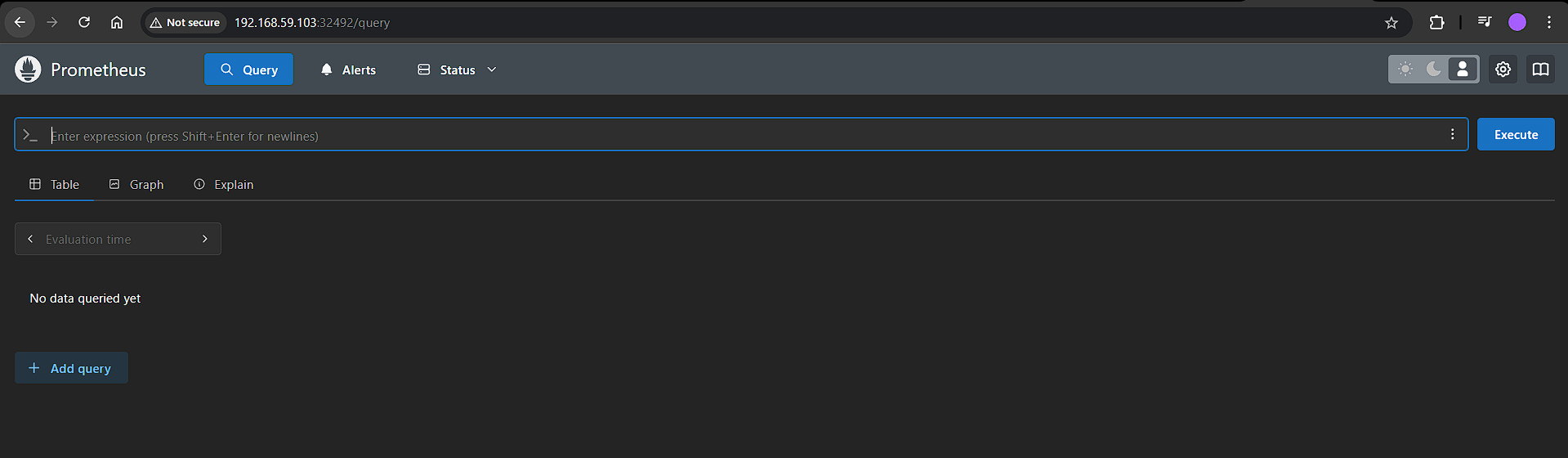
Exposing the prometheus-server:



* kubectl expose service prometheus-server --type=NodePort --target-port=9090 --name=faraz-prometheus-server-external



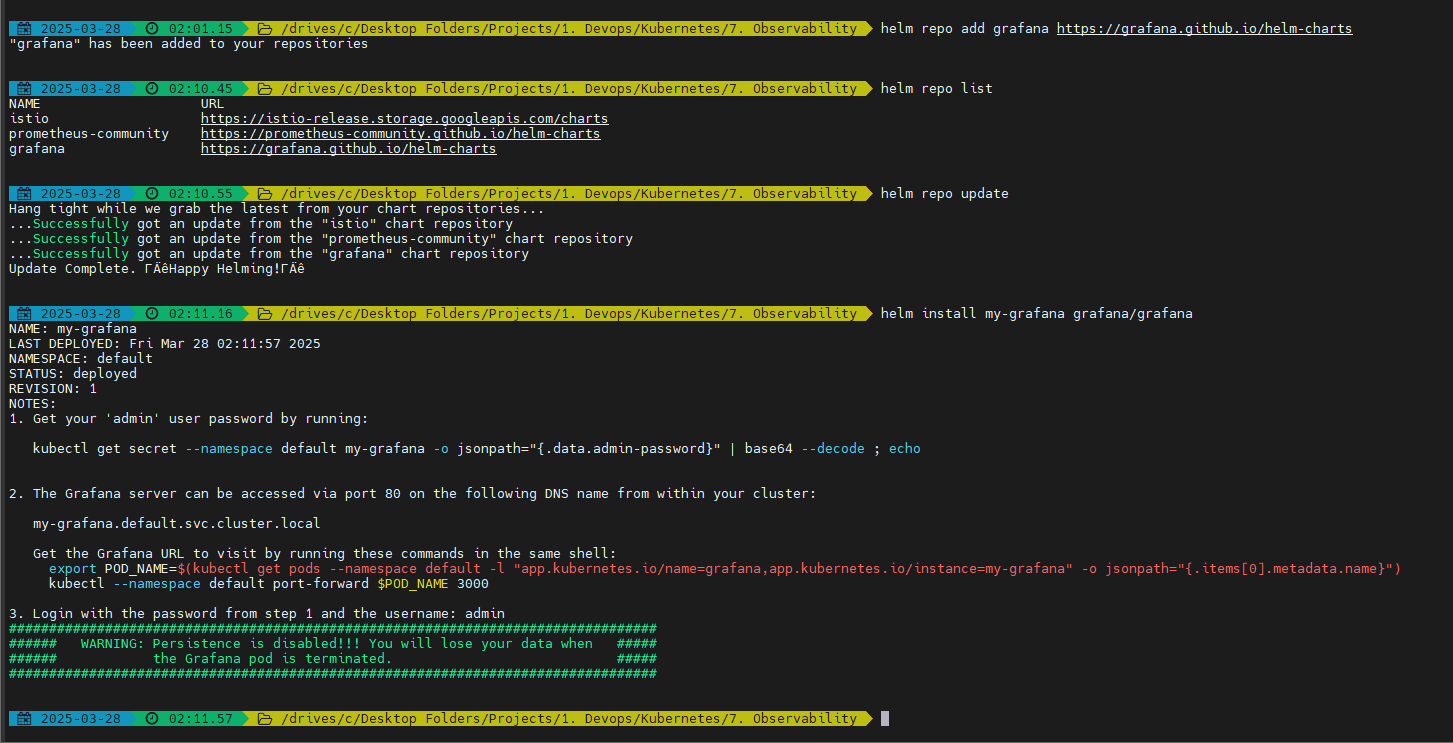
You will be able to view on:



* By default we may get information of metrics exposed by your K8s API server
* If we have an application and we want to get extra information then developers need to write metrics server using Prometheus Matrix Library and expose metrics end points. We can scrape these metrics inside config maps.

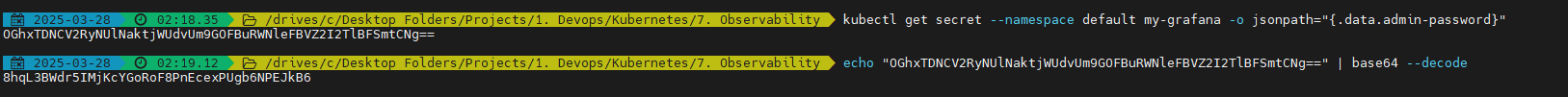
Now install Grafana:

* helm repo add grafana <https://grafana.github.io/helm-charts>
* helm repo list
* helm repo update
* helm install my-grafana grafana/grafana



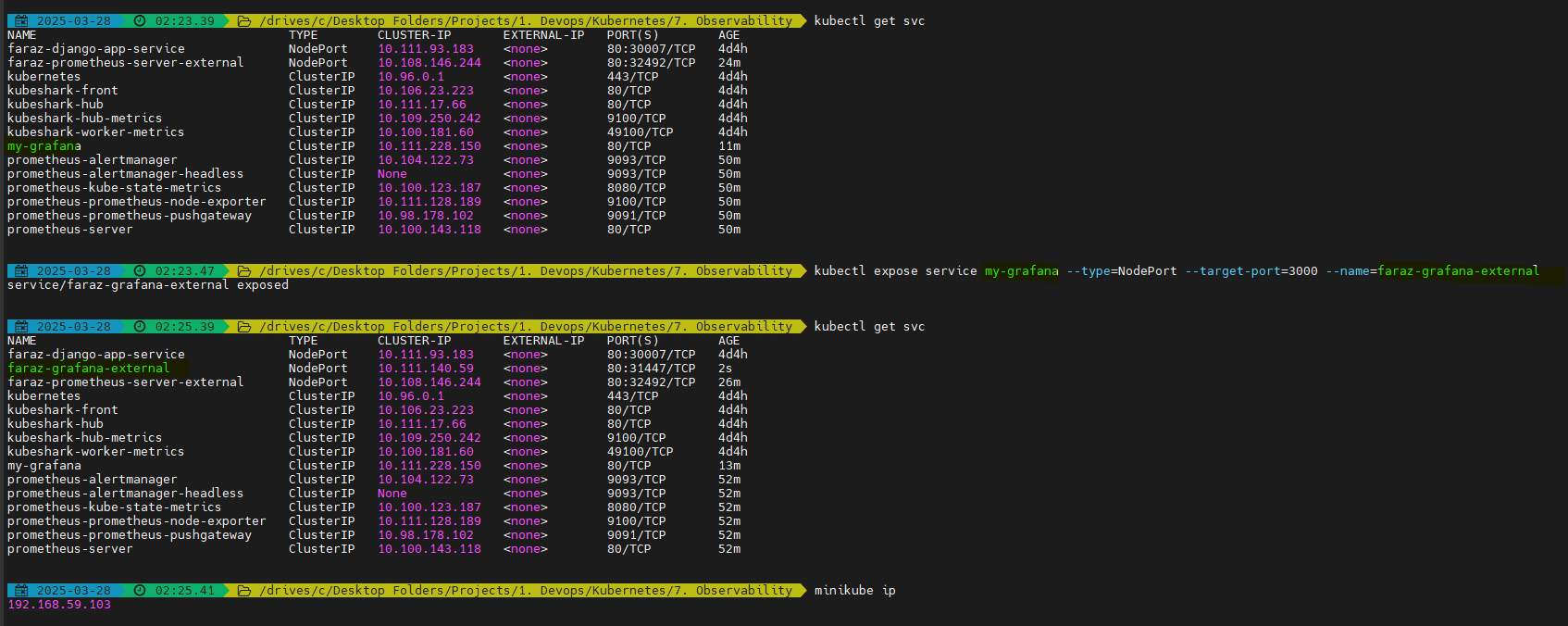
Get the password to login to Grafana:

* kubectl get secret --namespace default my-grafana -o jsonpath="{.data.admin-password}"
* echo "<output from above>" | base64 --decode



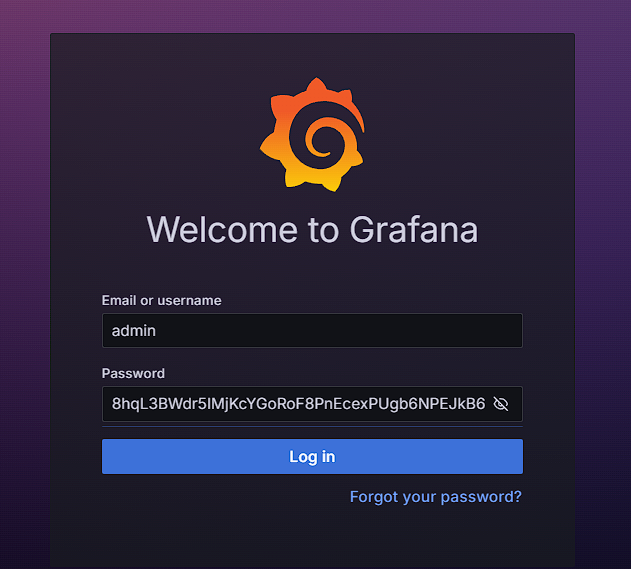
Now expose the service to NodePort mode (In production we may not do this):

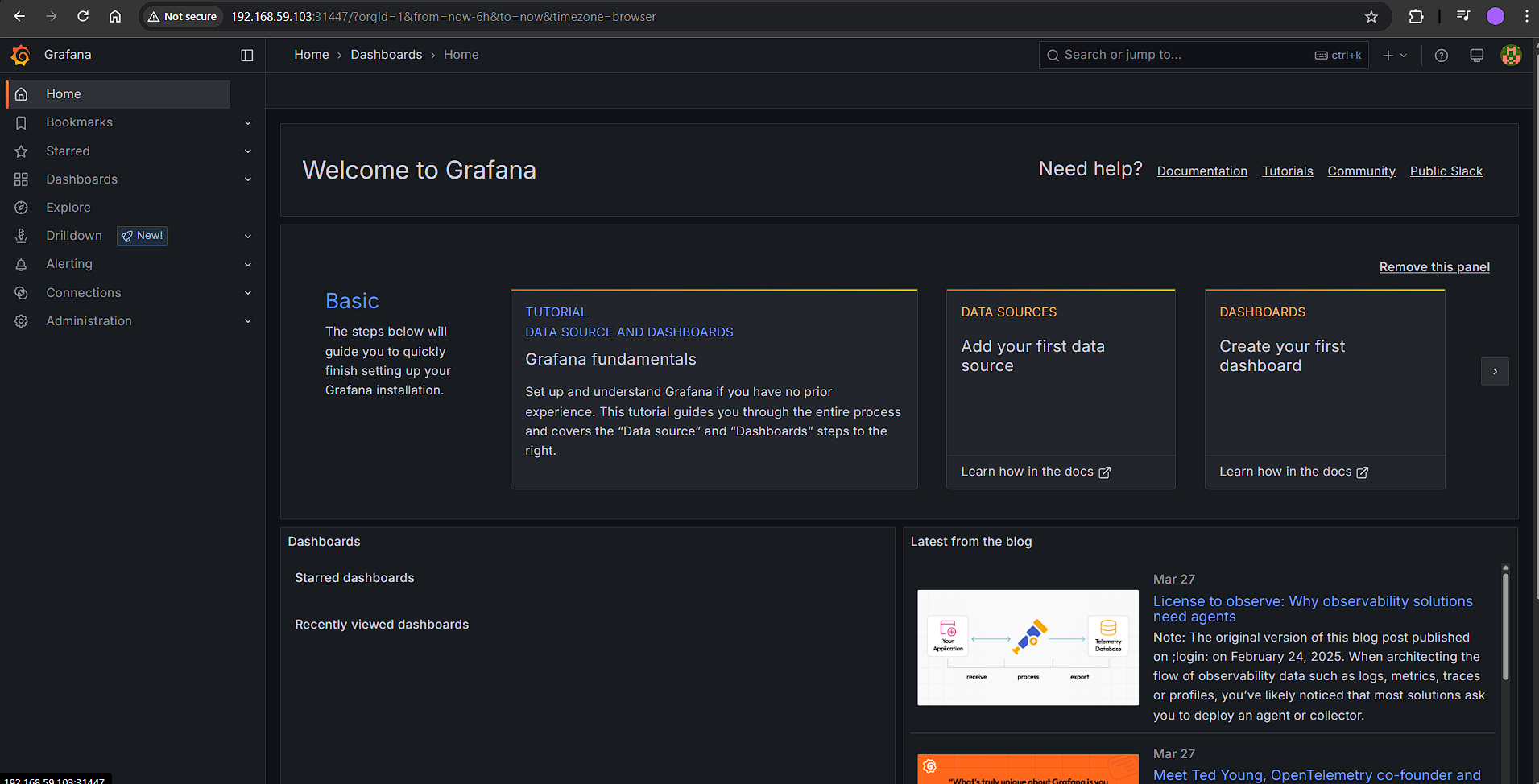
* kubectl expose service my-grafana --type=NodePort --target-port=3000 --name=faraz-grafana-external



Now provide the username and password to Grafana:

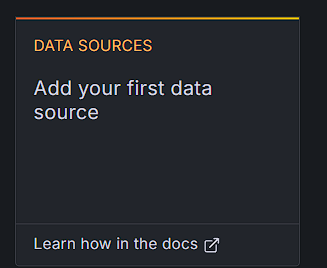






Now create Prometheus as a data source for Grafana

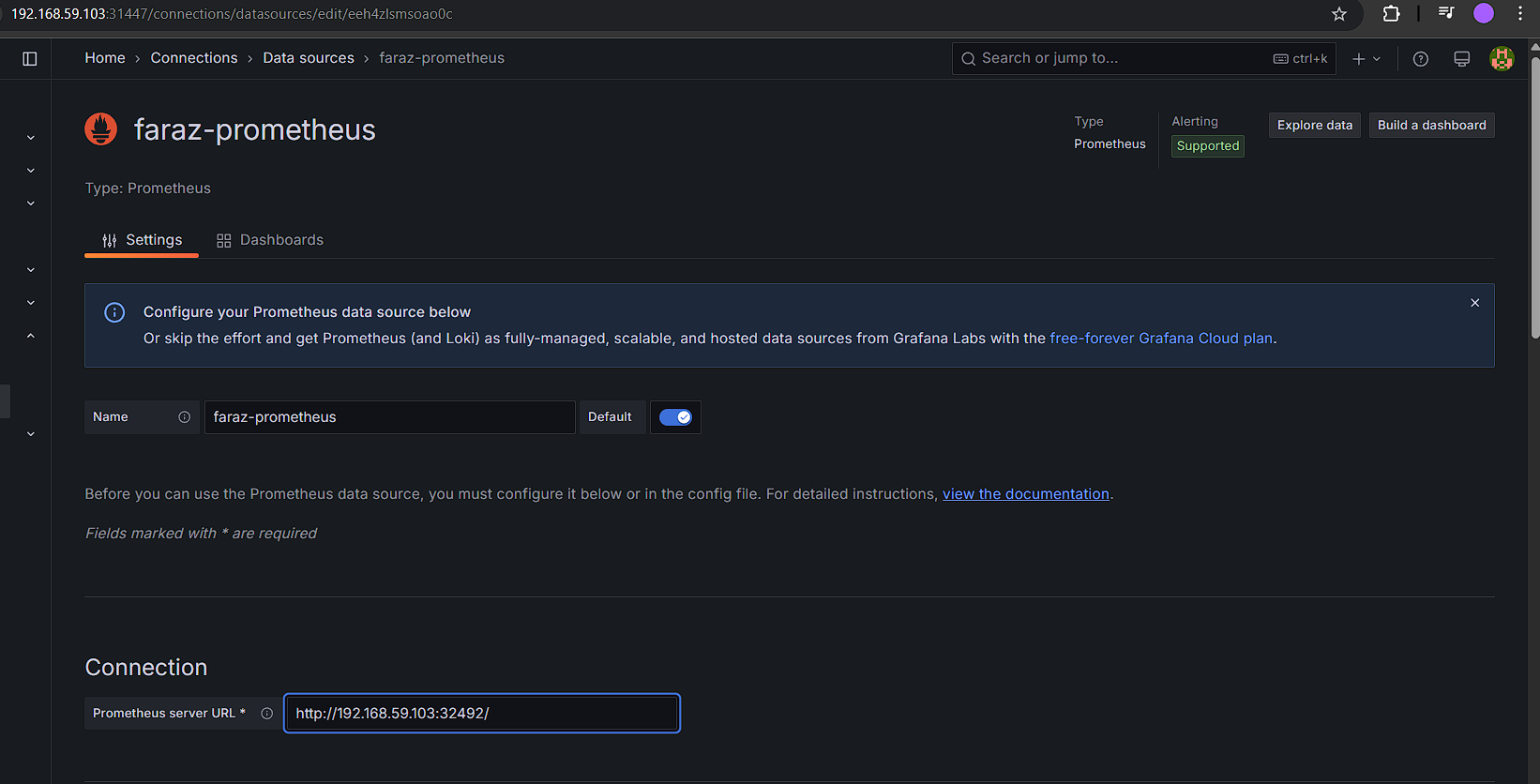
Click on Data source:



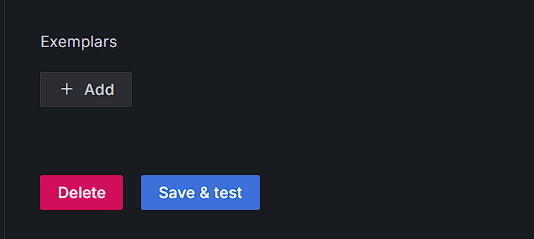
Choose the data source:

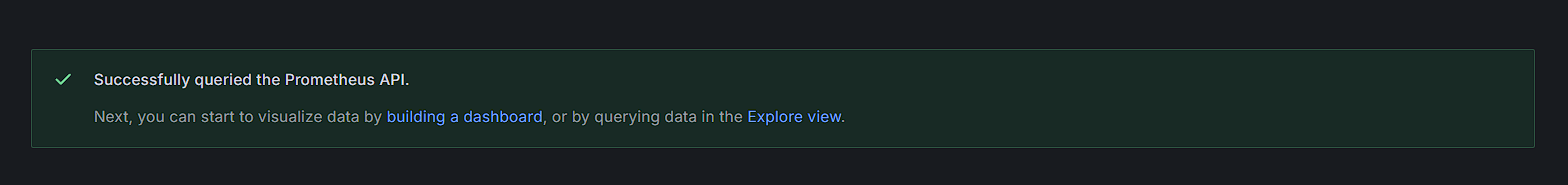


Add the IP address and give it any name:

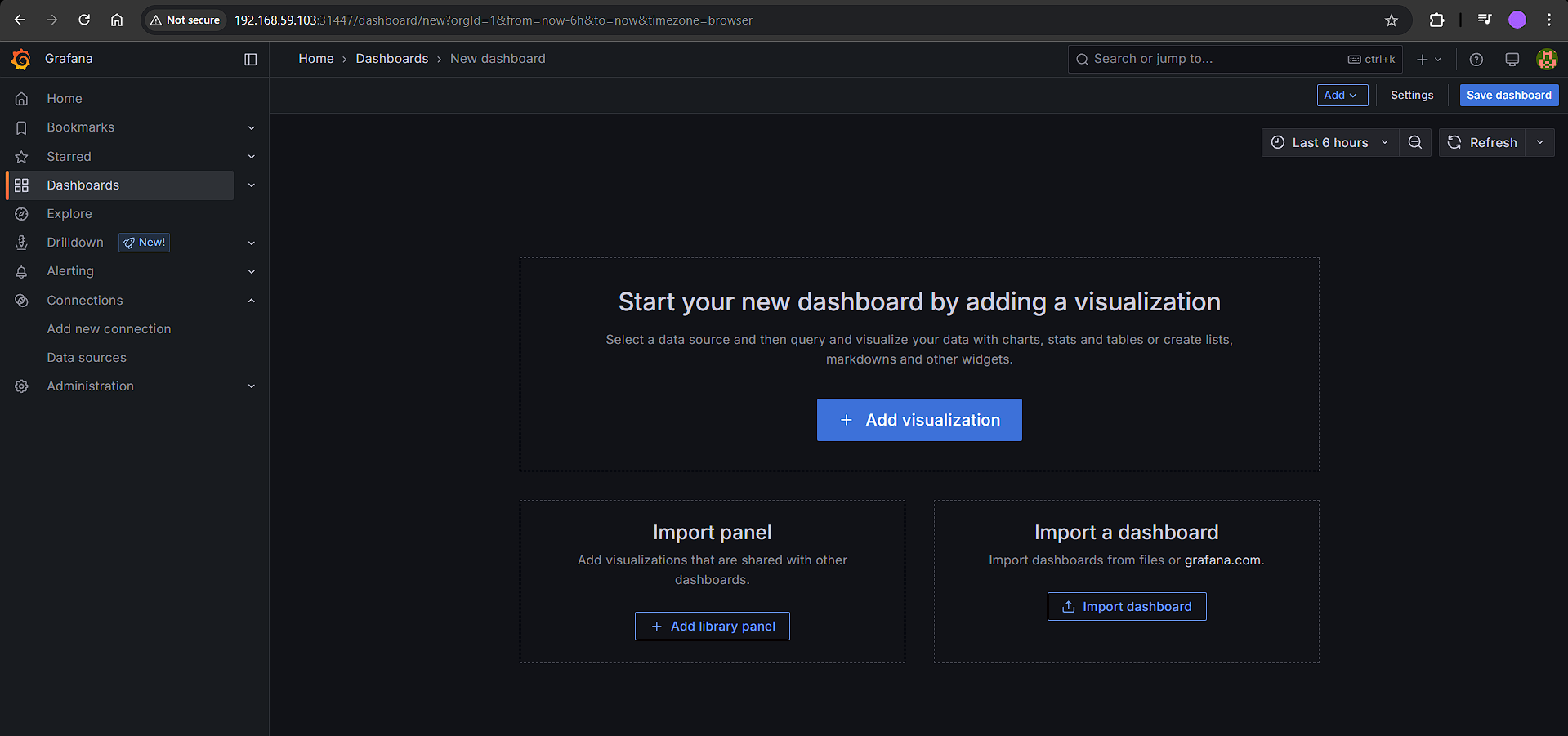


Click on save and test:

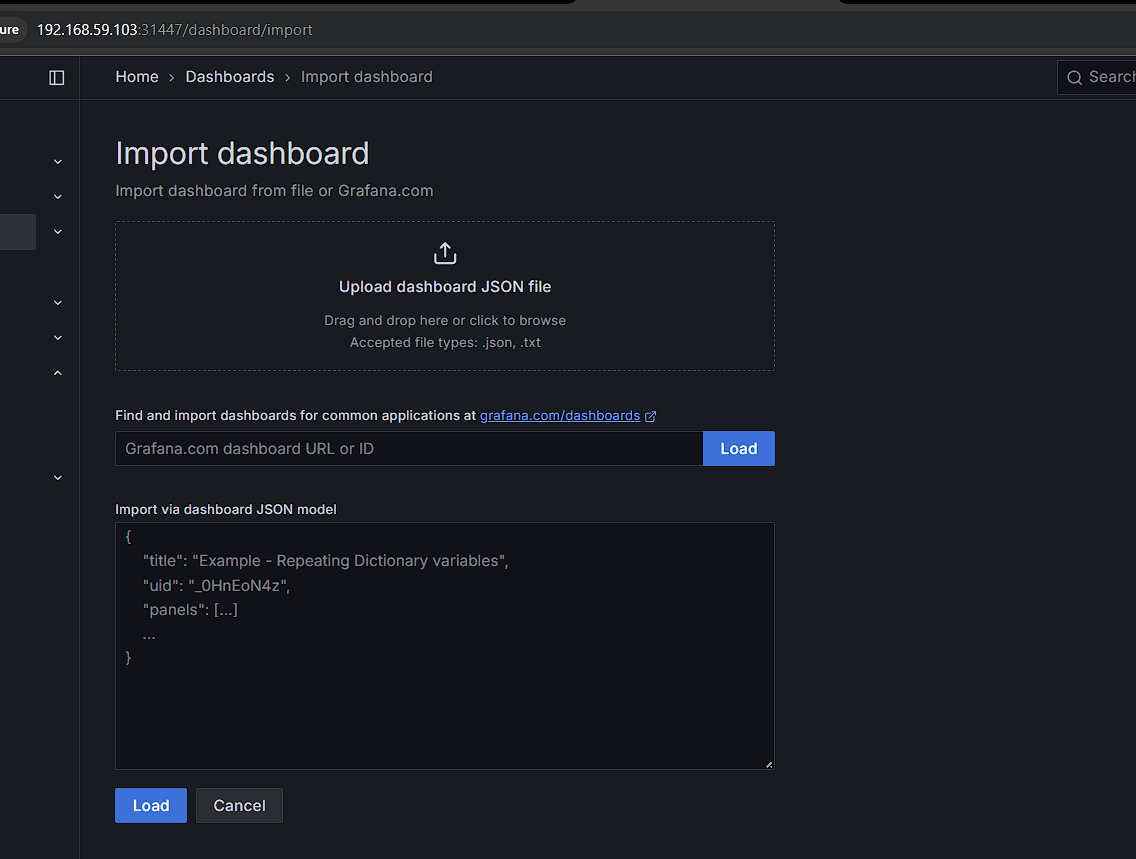




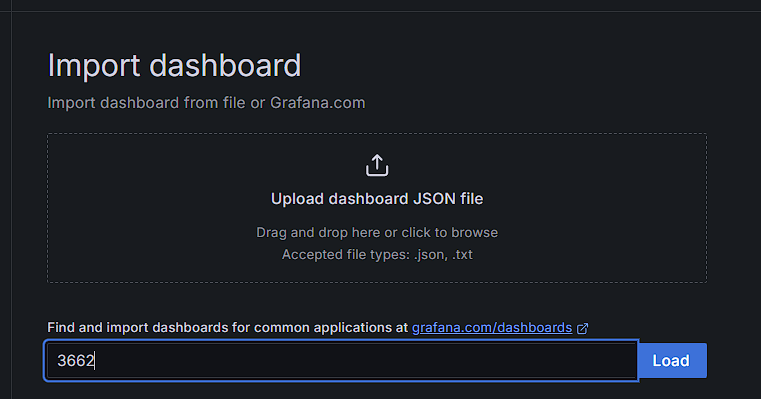
Now click on building a dashboard, you will be redirected to:



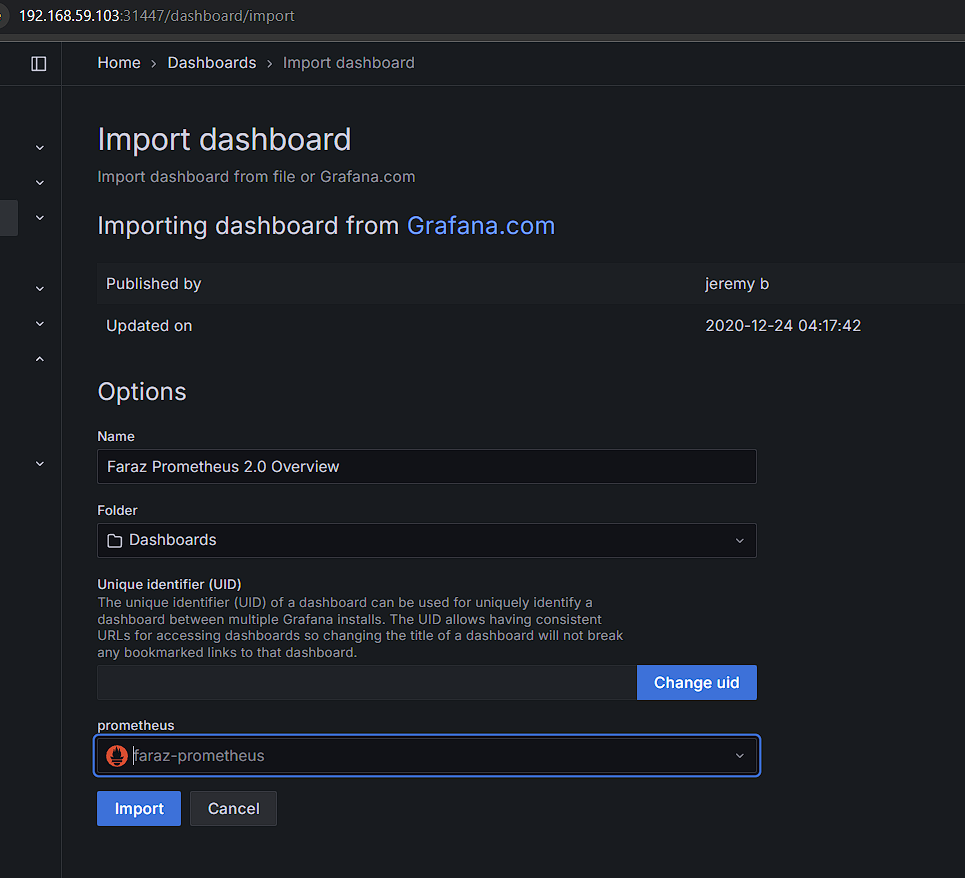
Now click on import dashboard:



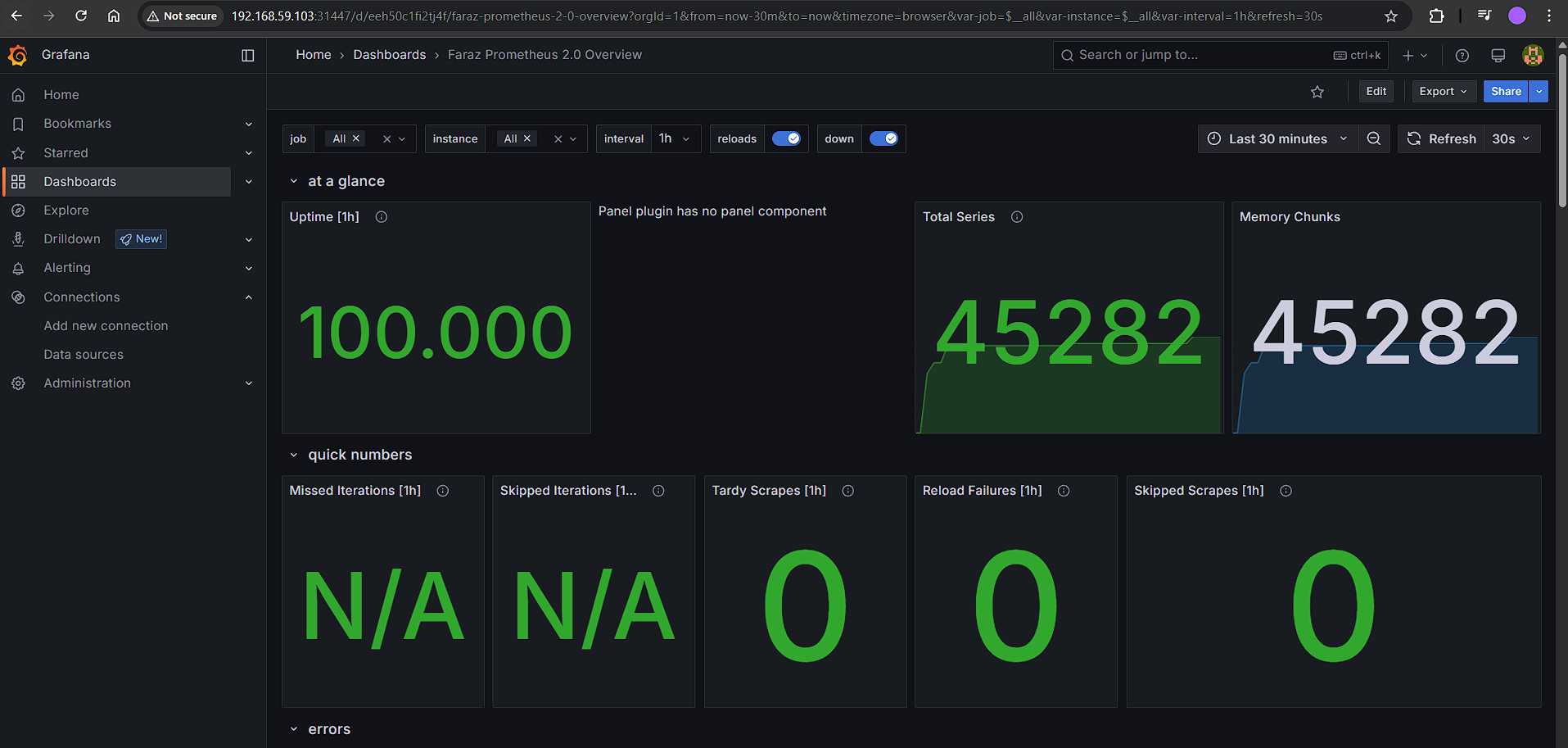
Use 3662 dashboard:

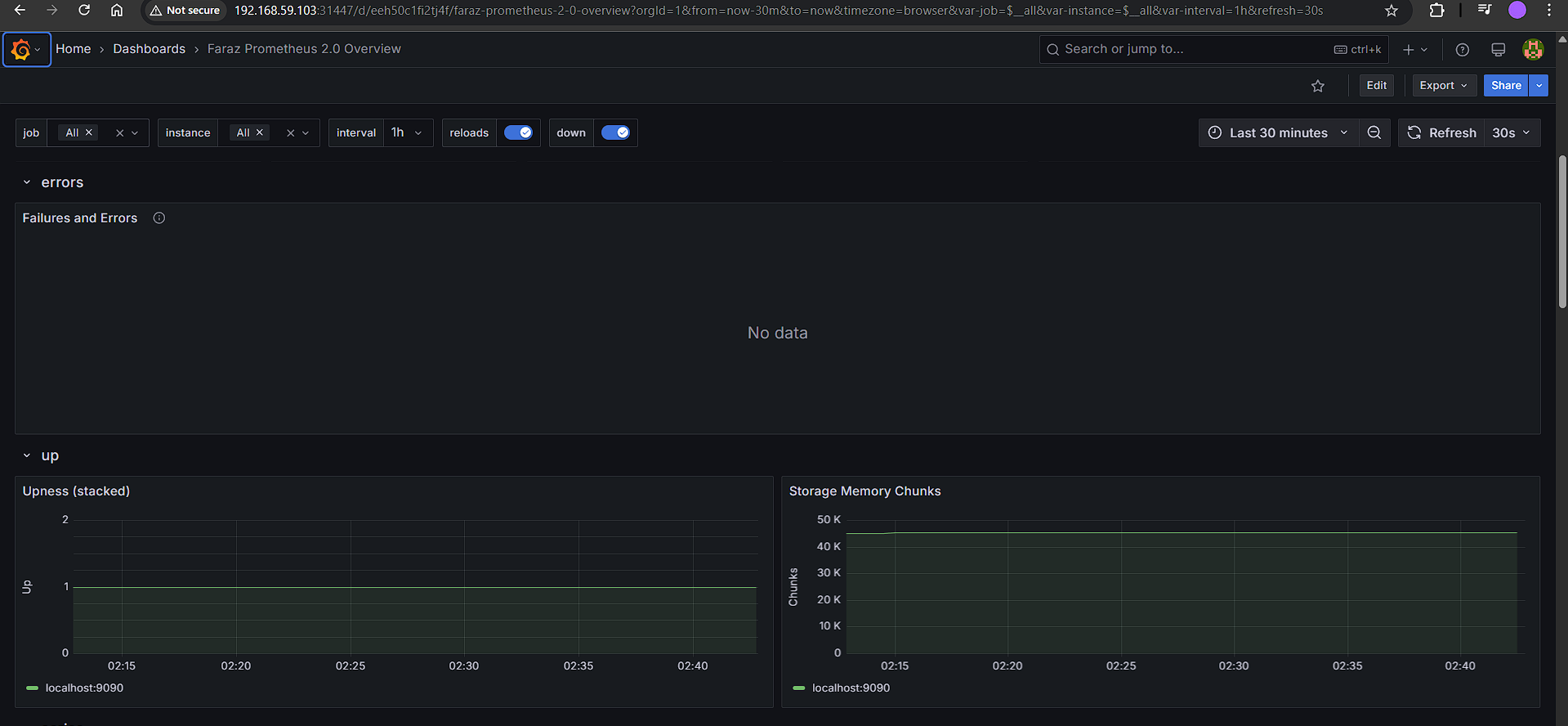


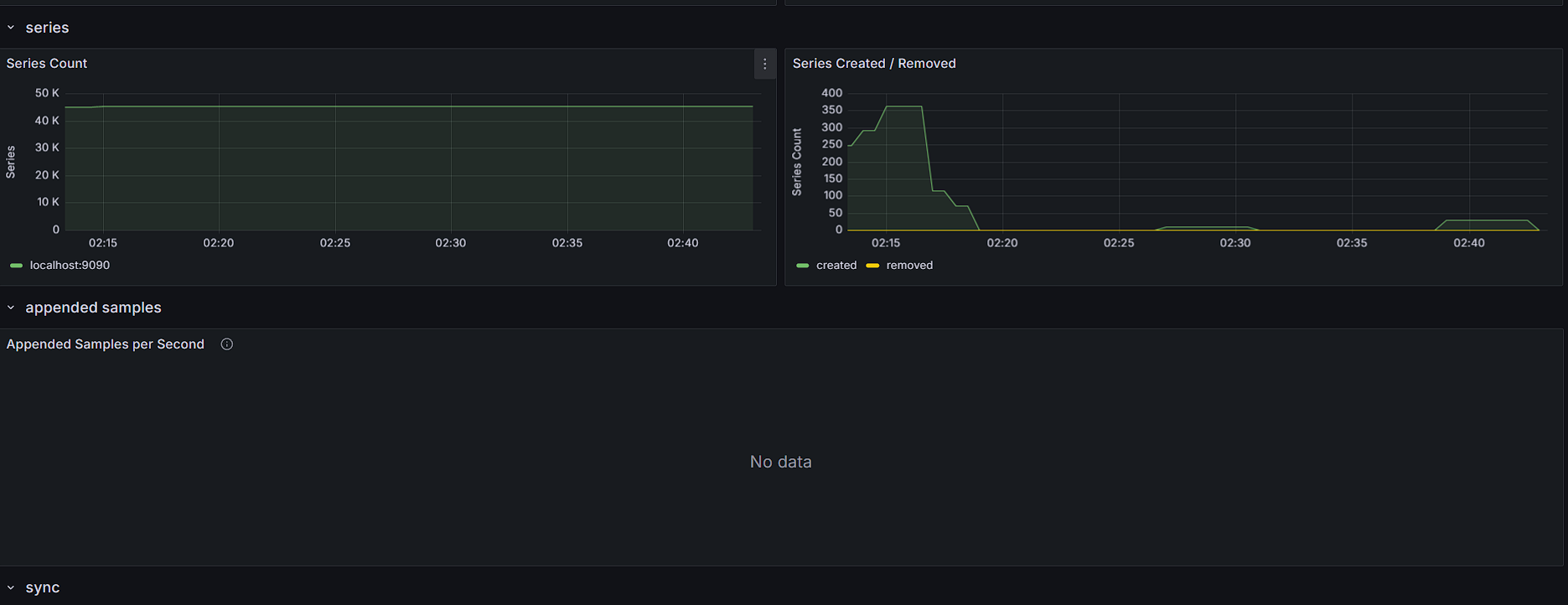
Now click on load & then import:



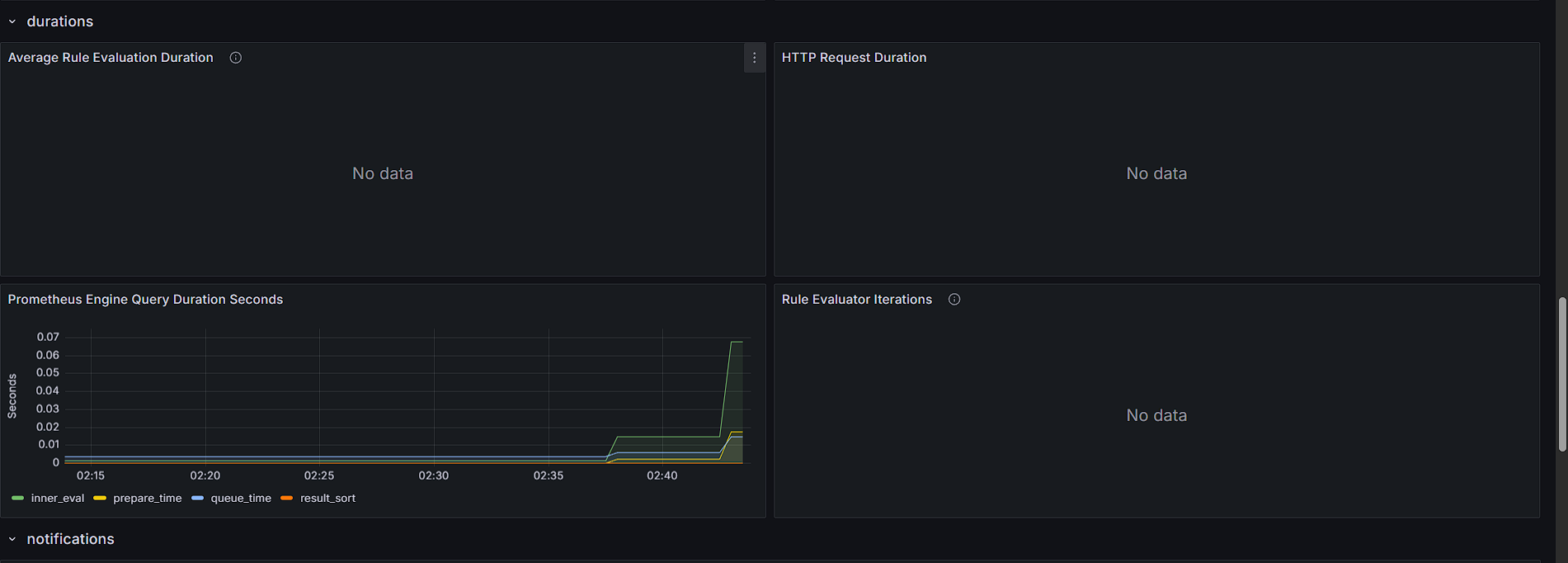
You may be able to view the dashboard:



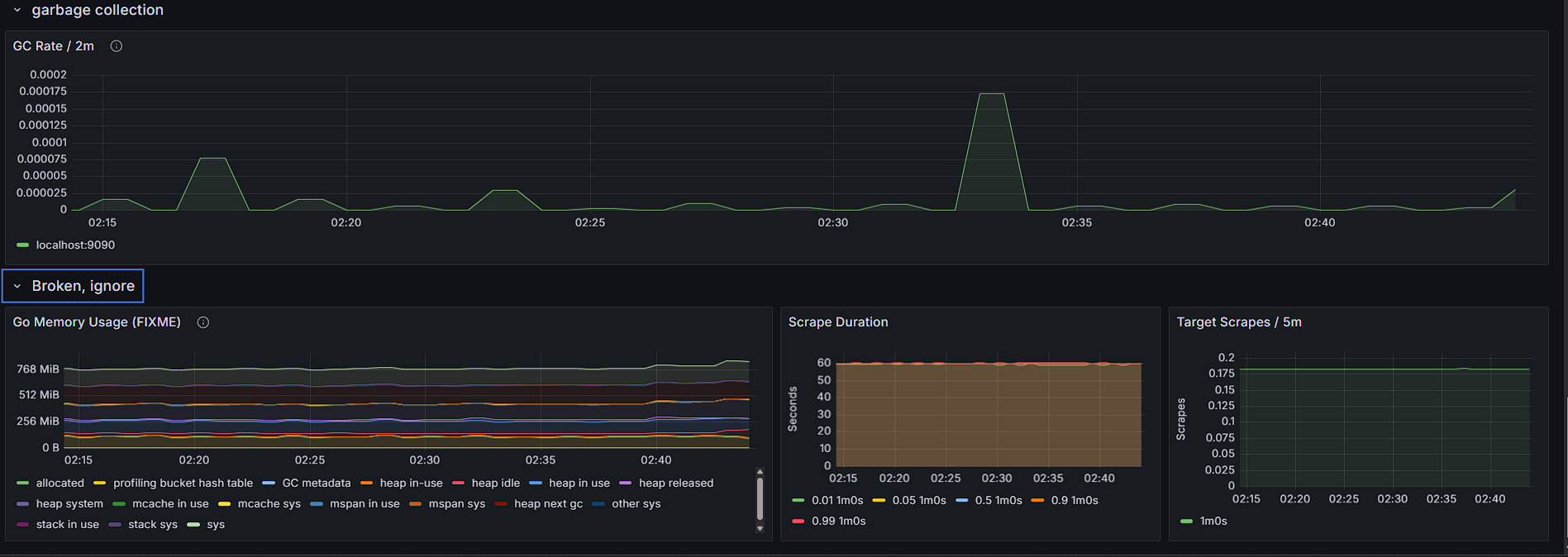






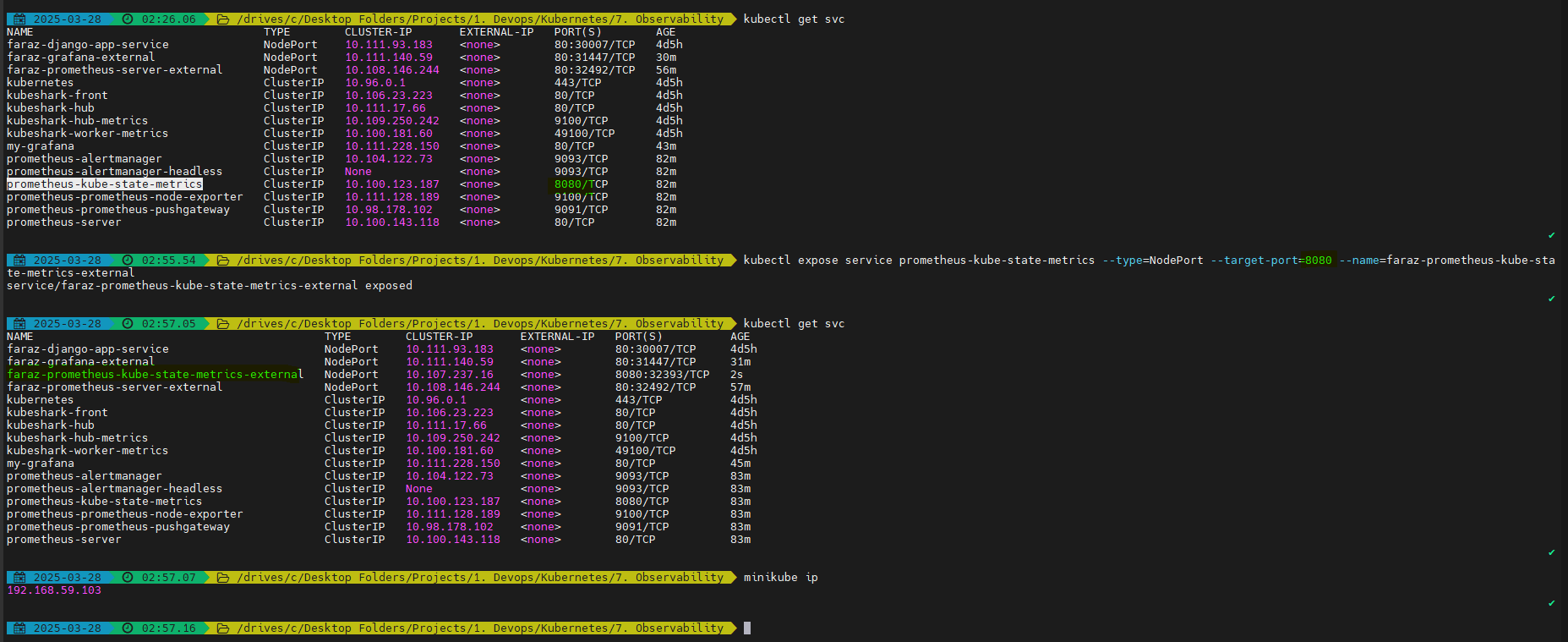




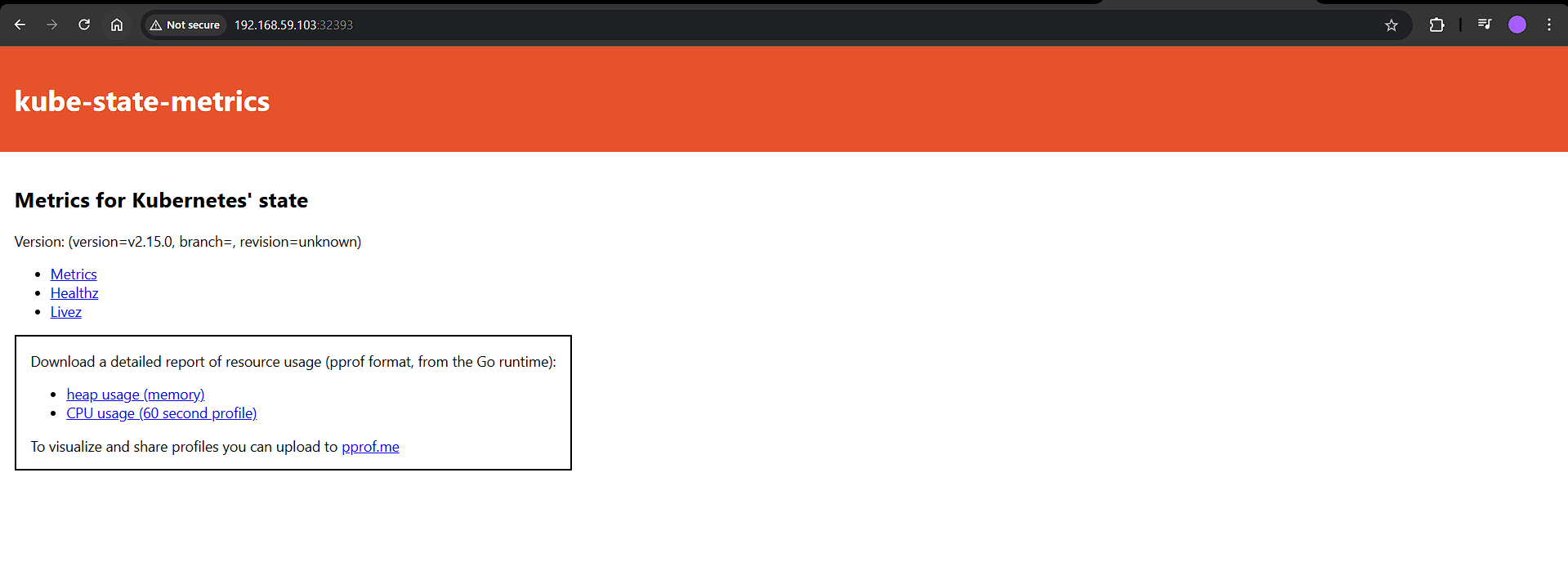


We can also expose kube-state-matrix:

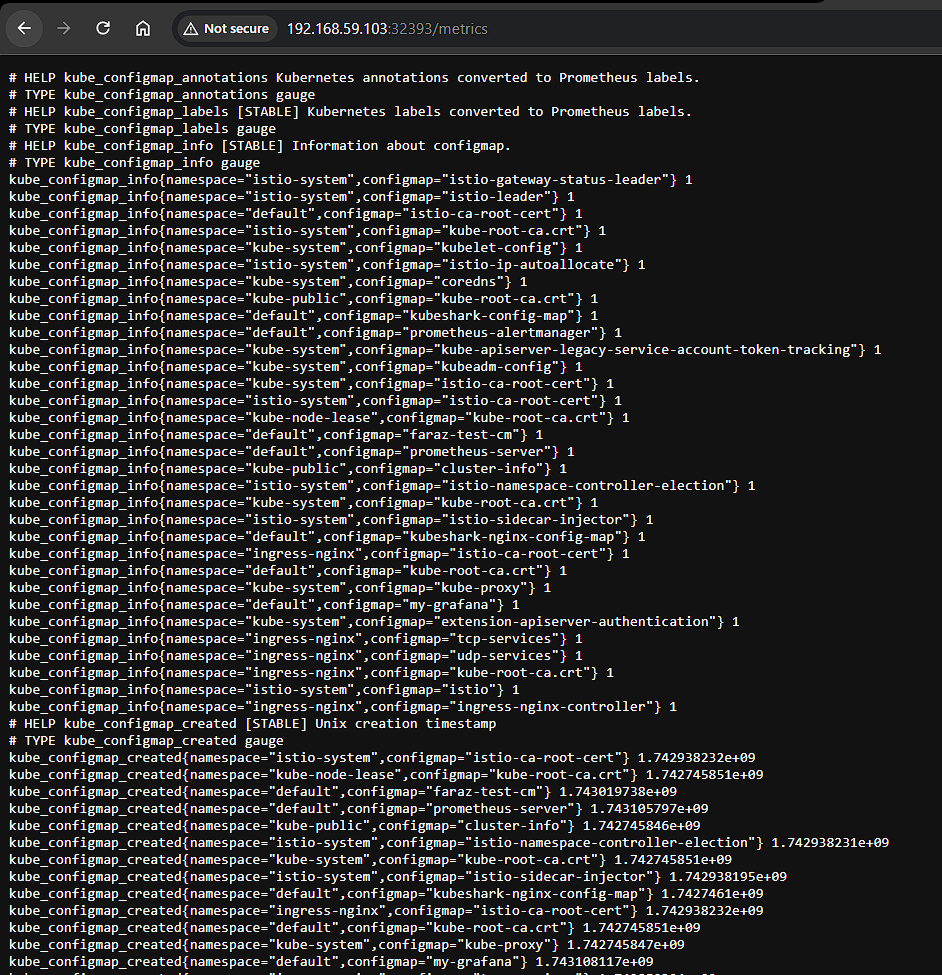
* kubectl expose service prometheus-kube-state-metrics --type=NodePort --target-port=8080 --name=faraz-prometheus-kube-state-metrics-external



You will be able to view:

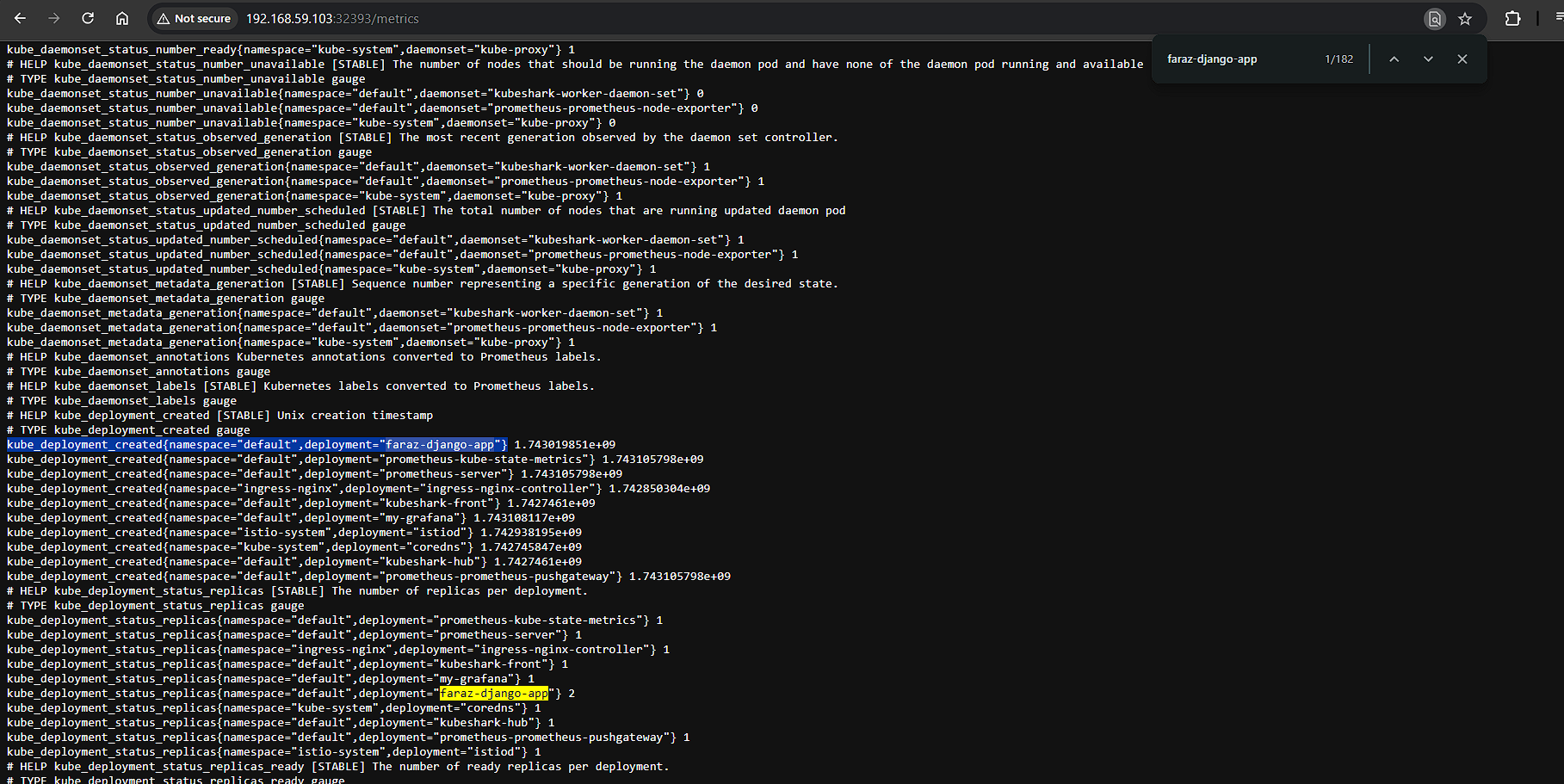


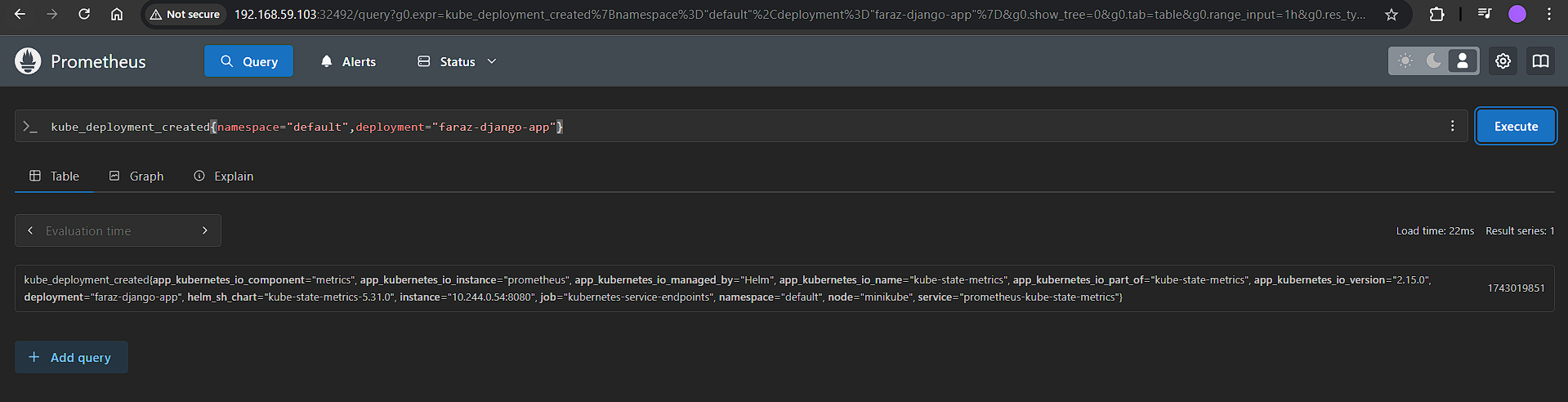
You get information of your metrics:

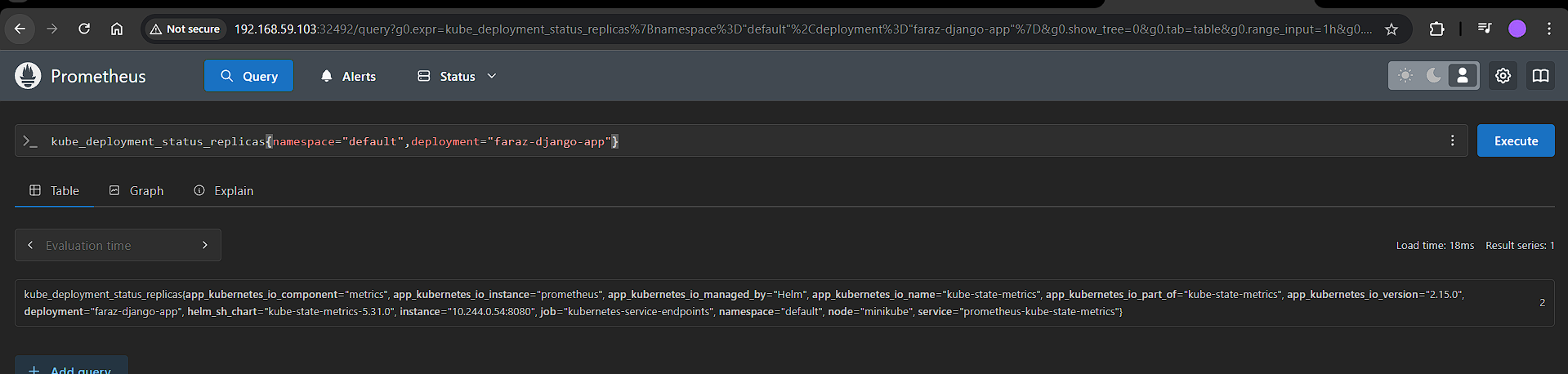


You can copy paste the promQL queries from here and use it in Prometheus

For example:

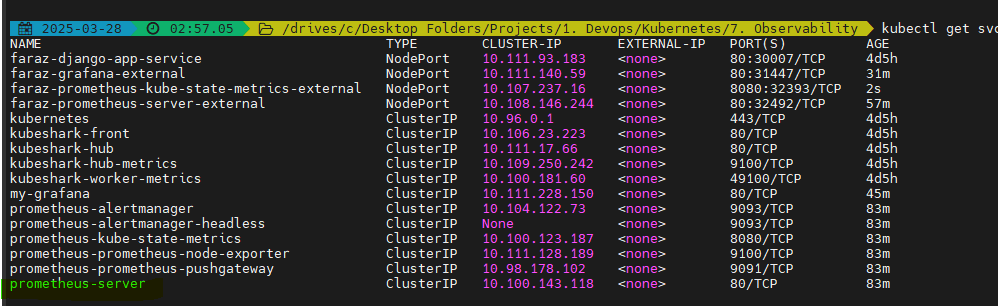




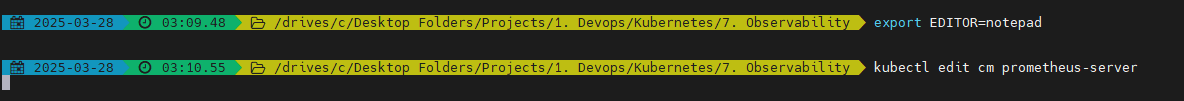


If we want more information in Grafana we can expose the Kube state matrix endpoint on <http://192.168.59.103:32393/metrics>

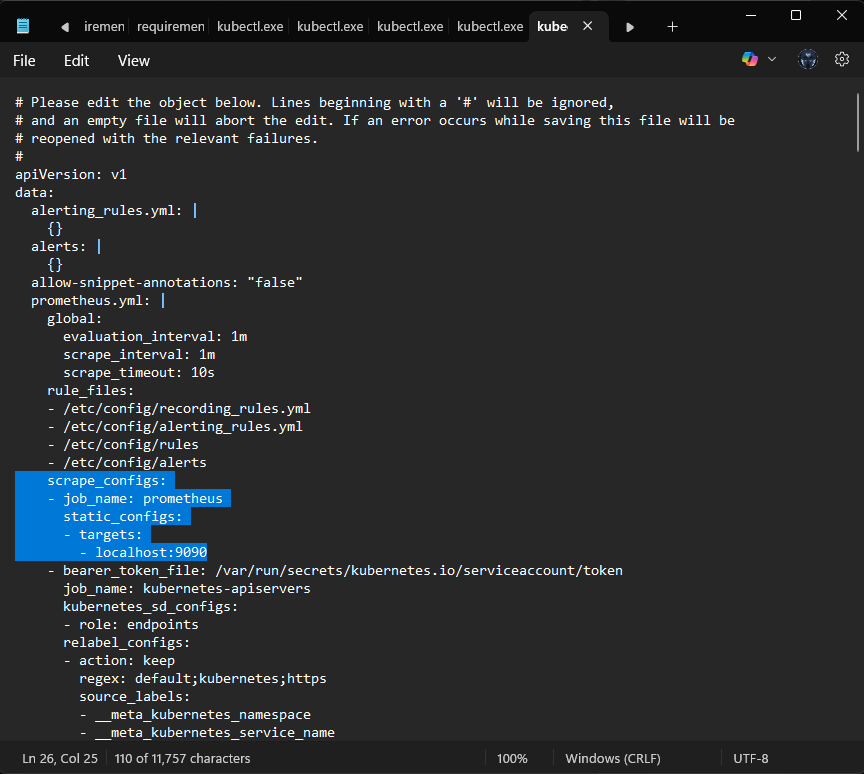
You can edit prometheus-server:



* export EDITOR=notepad
* kubectl edit cm prometheus-server



You can create a new scraping data:



- job\_name: state\_metrics

static\_configs:

- targets:

- <http://192.168.59.103:32393/metrics>

We can also ask developers to write Metrics server using Prometheus library