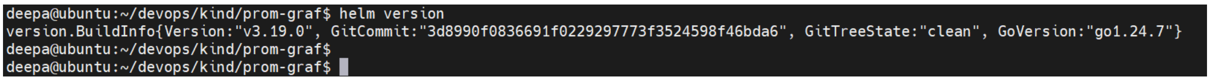
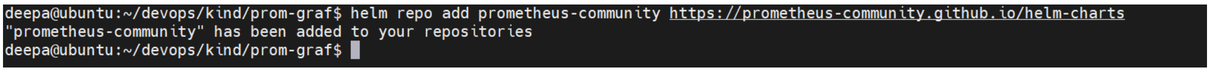
Helm: version



```
deepa@ubuntu:~/devops/kind/prom-graf$ helm version
version.BuildInfo{Version:"v3.19.0", GitCommit:"3d8990f0836691f0229297773f3524598f46bda6", GitTreeState:"clean", GoVersion:"go1.24.7"}
deepa@ubuntu:~/devops/kind/prom-graf$
deepa@ubuntu:~/devops/kind/prom-graf$
```

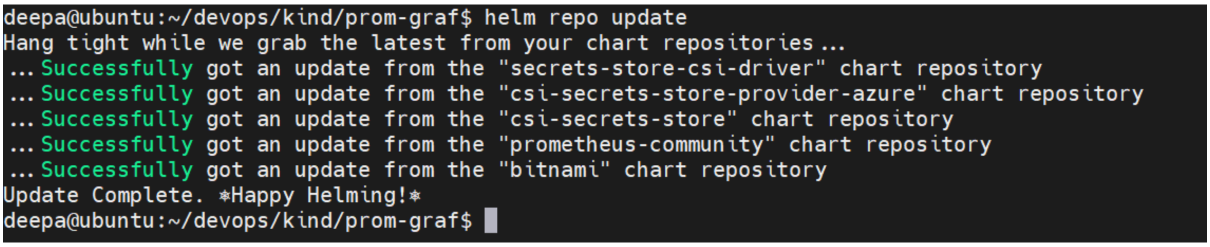
Now, add the **Prometheus community** Helm repo



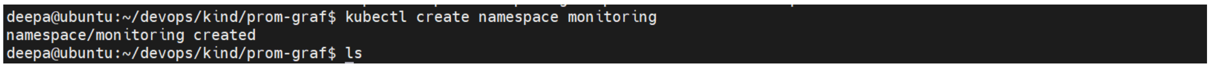
```
deepa@ubuntu:~/devops/kind/prom-graf$ helm repo add prometheus-community https://prometheus-community.github.io/helm-charts
"prometheus-community" has been added to your repositories
deepa@ubuntu:~/devops/kind/prom-graf$
```

update Helm repo list:

helm repo update



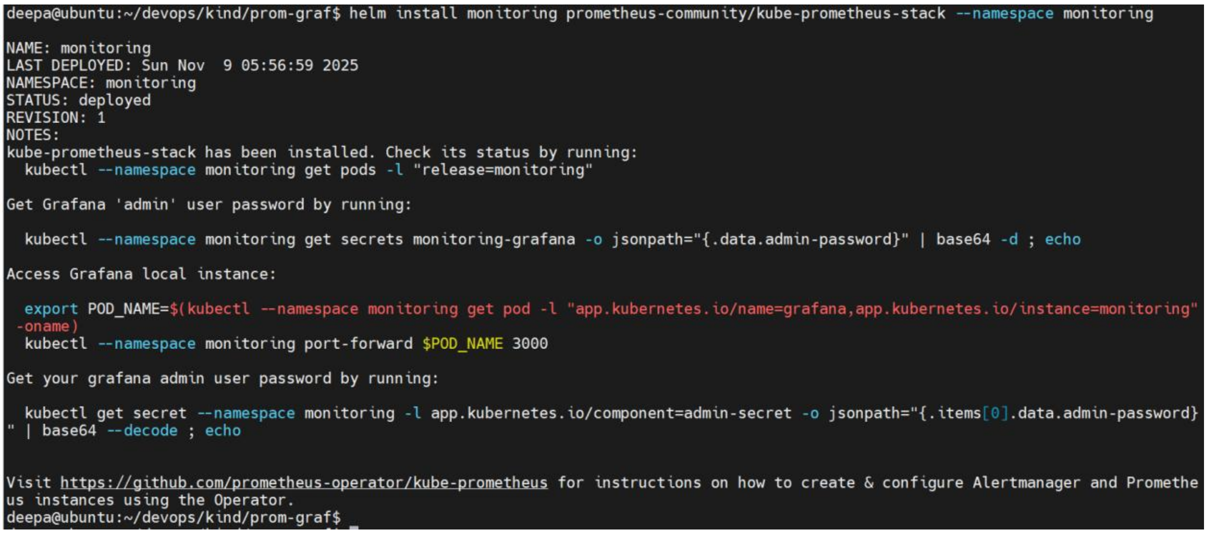
```
deepa@ubuntu:~/devops/kind/prom-graf$ helm repo update
Hang tight while we grab the latest from your chart repositories...
...Successfully got an update from the "secrets-store-csi-driver" chart repository
...Successfully got an update from the "csi-secrets-store-provider-azure" chart repository
...Successfully got an update from the "csi-secrets-store" chart repository
...Successfully got an update from the "prometheus-community" chart repository
...Successfully got an update from the "bitnami" chart repository
Update Complete. ⎈Happy Helming!⎈
deepa@ubuntu:~/devops/kind/prom-graf$
```



```
deepa@ubuntu:~/devops/kind/prom-graf$ kubectl create namespace monitoring
namespace/monitoring created
deepa@ubuntu:~/devops/kind/prom-graf$ ls
```

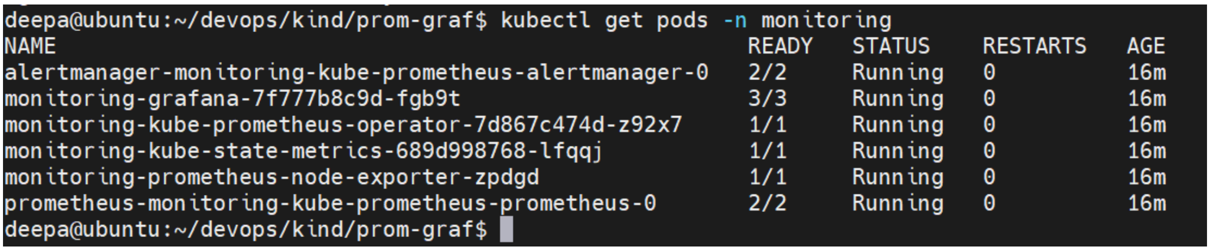
Install the kube-prometheus-stack (includes Prometheus + Grafana)
Single chart deploys **Prometheus**, **Grafana**, and **Alertmanager** together — all pre-integrated.
"helm install monitoring prometheus-community/kube-prometheus-stack --namespace monitoring"
This creates:

- prometheus-kube-prometheus-stack (Prometheus server)
- grafana (Grafana dashboard)
- alertmanager
- Node exporter, kube-state-metrics, etc.



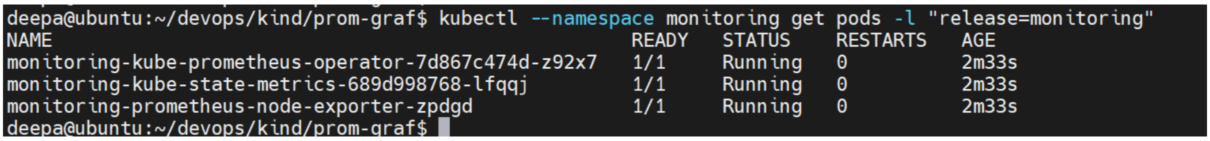
```
deepa@ubuntu:~/devops/kind/prom-graf$ helm install monitoring prometheus-community/kube-prometheus-stack --namespace monitoring

NAME: monitoring
LAST DEPLOYED: Sun Nov  9 05:56:59 2025
NAMESPACE: monitoring
STATUS: deployed
REVISION: 1
NOTES:
kube-prometheus-stack has been installed. Check its status by running:
  kubectl --namespace monitoring get pods -l "release=monitoring"

Get Grafana 'admin' user password by running:

  kubectl --namespace monitoring get secrets monitoring-grafana -o jsonpath="{.data.admin-password}" | base64 -d ; echo

Access Grafana local instance:

  export POD_NAME=$(kubectl --namespace monitoring get pod -l "app.kubernetes.io/name=grafana,app.kubernetes.io/instance=monitoring"
 -oname)
  kubectl --namespace monitoring port-forward $POD_NAME 3000

Get your grafana admin user password by running:

  kubectl get secret --namespace monitoring -l app.kubernetes.io/component=admin-secret -o jsonpath="{.items[0].data.admin-password}
" | base64 --decode ; echo


Visit https://github.com/prometheus-operator/kube-prometheus for instructions on how to create & configure Alertmanager and Promethe
us instances using the Operator.
deepa@ubuntu:~/devops/kind/prom-graf$
```

Verify all components:



```
deepa@ubuntu:~/devops/kind/prom-graf$ kubectl get pods -n monitoring
NAME                                                     READY   STATUS    RESTARTS   AGE
alertmanager-monitoring-kube-prometheus-alertmanager-0   2/2     Running   0          16m
monitoring-grafana-7f777b8c9d-fgb9t                      3/3     Running   0          16m
monitoring-kube-prometheus-operator-7d867c474d-z92x7     1/1     Running   0          16m
monitoring-kube-state-metrics-689d998768-lfqqj           1/1     Running   0          16m
monitoring-prometheus-node-exporter-zpdgd                1/1     Running   0          16m
prometheus-monitoring-kube-prometheus-prometheus-0       2/2     Running   0          16m
deepa@ubuntu:~/devops/kind/prom-graf$
```

kubectl --namespace monitoring get pods -l "release=monitoring"

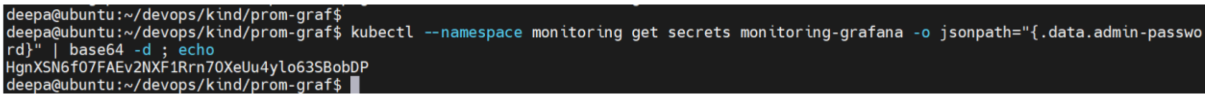


```
deepa@ubuntu:~/devops/kind/prom-graf$ kubectl --namespace monitoring get pods -l "release=monitoring"
NAME                                                   READY   STATUS    RESTARTS   AGE
monitoring-kube-prometheus-operator-7d867c474d-z92x7   1/1     Running   0          2m33s
monitoring-kube-state-metrics-689d998768-lfqqj         1/1     Running   0          2m33s
monitoring-prometheus-node-exporter-zpdgd              1/1     Running   0          2m33s
deepa@ubuntu:~/devops/kind/prom-graf$
```

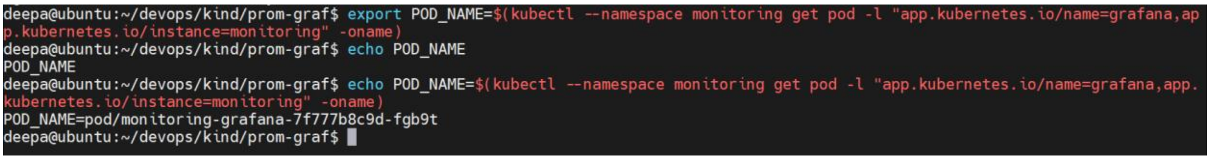
Access Grafana Dashboard:

Grafana runs as a ClusterIP service by default. To access it:

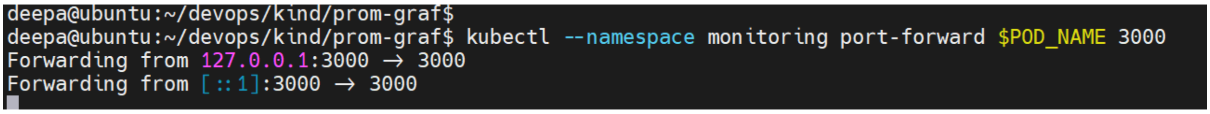
kubectl --namespace monitoring get secrets monitoring-grafana -o jsonpath="{.data.admin-password}" | base64 -d ; echo



```
deepa@ubuntu:~/devops/kind/prom-graf$
deepa@ubuntu:~/devops/kind/prom-graf$ kubectl --namespace monitoring get secrets monitoring-grafana -o jsonpath="{.data.admin-passwo
rd}" | base64 -d ; echo
HgnXSN6f07FAEv2NXF1Rrn70XeUu4ylo63SBobDP
deepa@ubuntu:~/devops/kind/prom-graf$
```

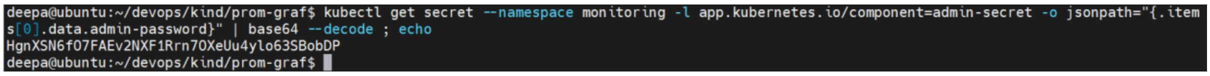


```
deepa@ubuntu:~/devops/kind/prom-graf$ export POD_NAME=$(kubectl --namespace monitoring get pod -l "app.kubernetes.io/name=grafana,ap
p.kubernetes.io/instance=monitoring" -oname)
deepa@ubuntu:~/devops/kind/prom-graf$ echo POD_NAME
POD_NAME
deepa@ubuntu:~/devops/kind/prom-graf$ echo POD_NAME=$(kubectl --namespace monitoring get pod -l "app.kubernetes.io/name=grafana,app.
kubernetes.io/instance=monitoring" -oname)
POD_NAME=pod/monitoring-grafana-7f777b8c9d-fgb9t
deepa@ubuntu:~/devops/kind/prom-graf$
```

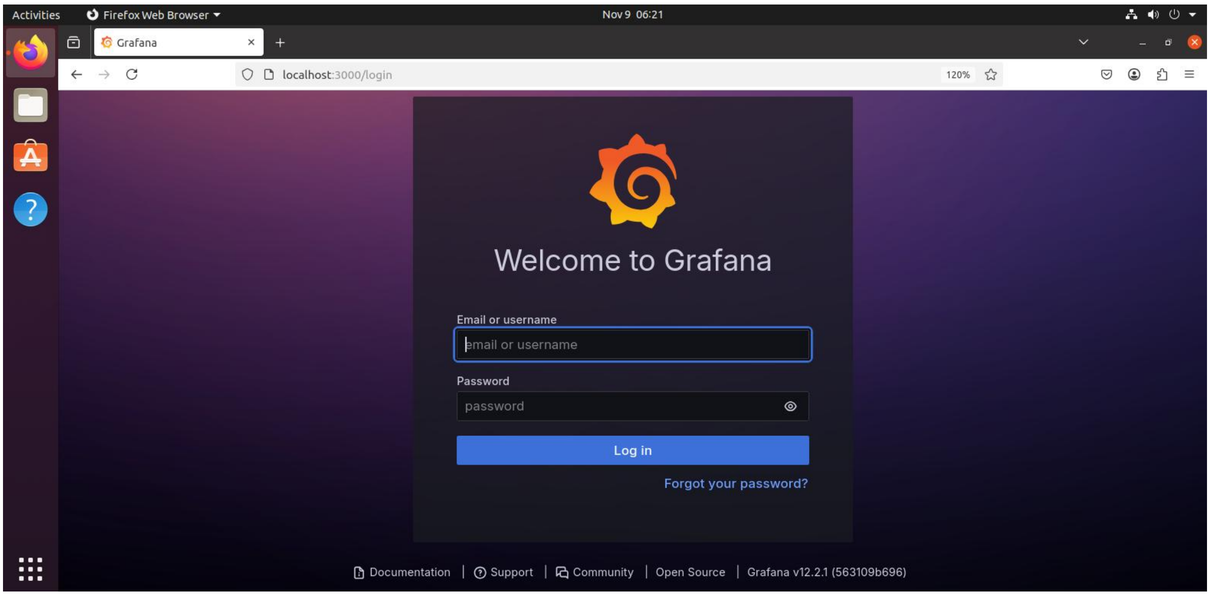


```
deepa@ubuntu:~/devops/kind/prom-graf$
deepa@ubuntu:~/devops/kind/prom-graf$ kubectl --namespace monitoring port-forward $POD_NAME 3000
Forwarding from 127.0.0.1:3000 → 3000
Forwarding from [::1]:3000 → 3000
```

Grafana’s admin credentials are stored in a secret:



```
deepa@ubuntu:~/devops/kind/prom-graf$ kubectl get secret --namespace monitoring -l app.kubernetes.io/component=admin-secret -o jsonpath="{.item
s[0].data.admin-password}" | base64 --decode ; echo
HgnXSN6f07FAEv2NXF1Rrn70XeUu4ylo63SBobDP
deepa@ubuntu:~/devops/kind/prom-graf$
```



localhost:3000/login

Welcome to Grafana

Email or username

email or username

Password

password

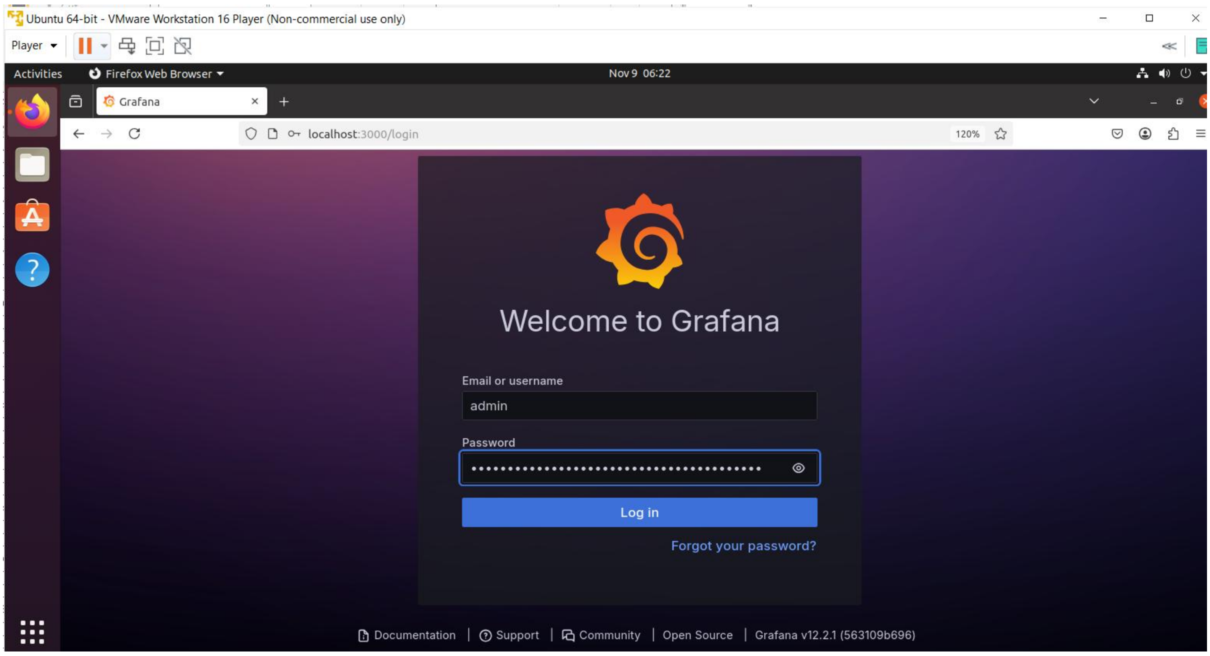
Log in

Forgot your password?

Documentation | Support | Community | Open Source | Grafana v12.2.1 (563109b696)

Default username: admin

Password: HgnXSN6fO7FAEv2NXF1Rrn7OXeUu4ylo63SBobDP



Ubuntu 64-bit - VMware Workstation 16 Player (Non-commercial use only)

localhost:3000/login

Welcome to Grafana

Email or username

admin

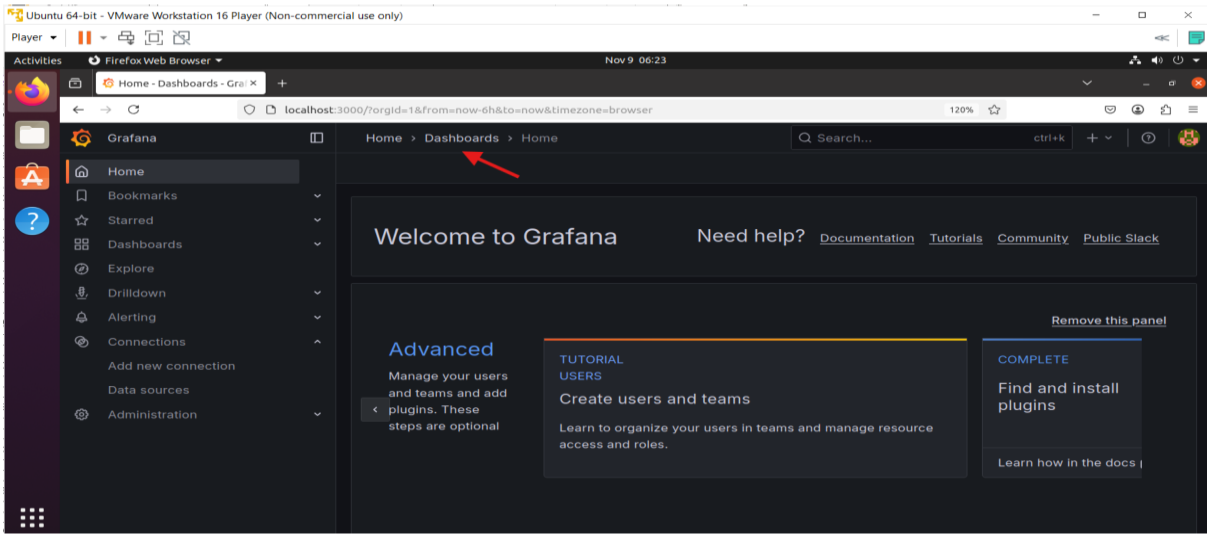
Password

••

Log in

Forgot your password?

Documentation | Support | Community | Open Source | Grafana v12.2.1 (563109b696)



Welcome to Grafana

Advanced

Manage your users and teams and add plugins. These steps are optional

TUTORIAL
USERS

Create users and teams

Learn to organize your users in teams and manage resource access and roles.

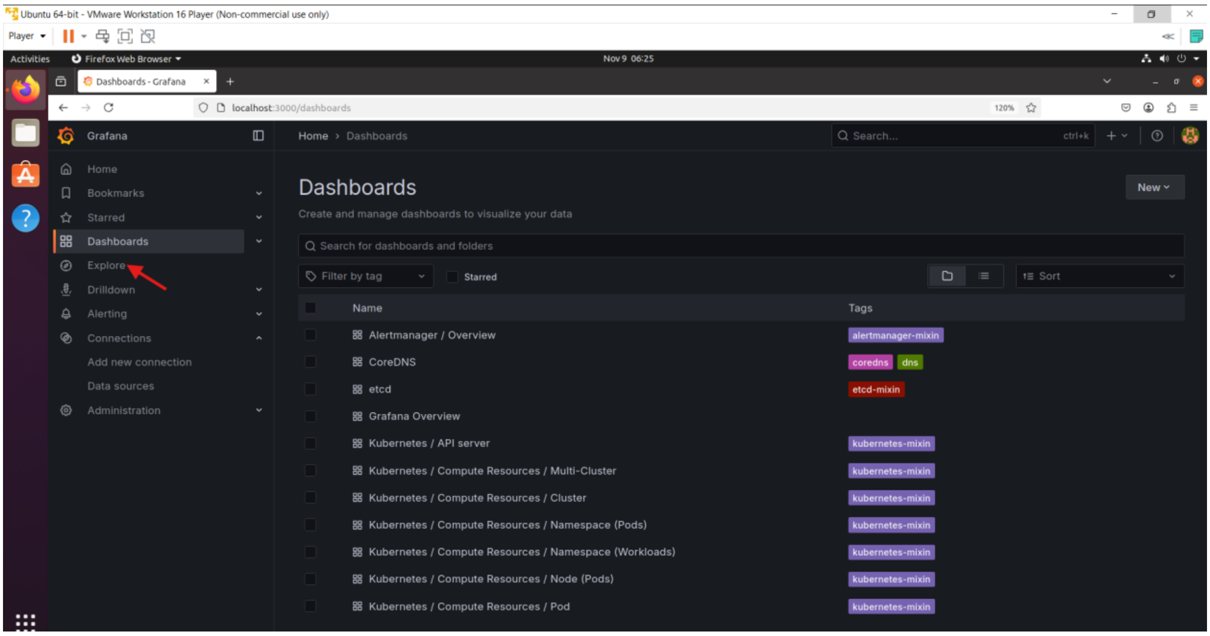
COMPLETE

Find and install plugins

Learn how in the docs

To View metrics & dashboards:

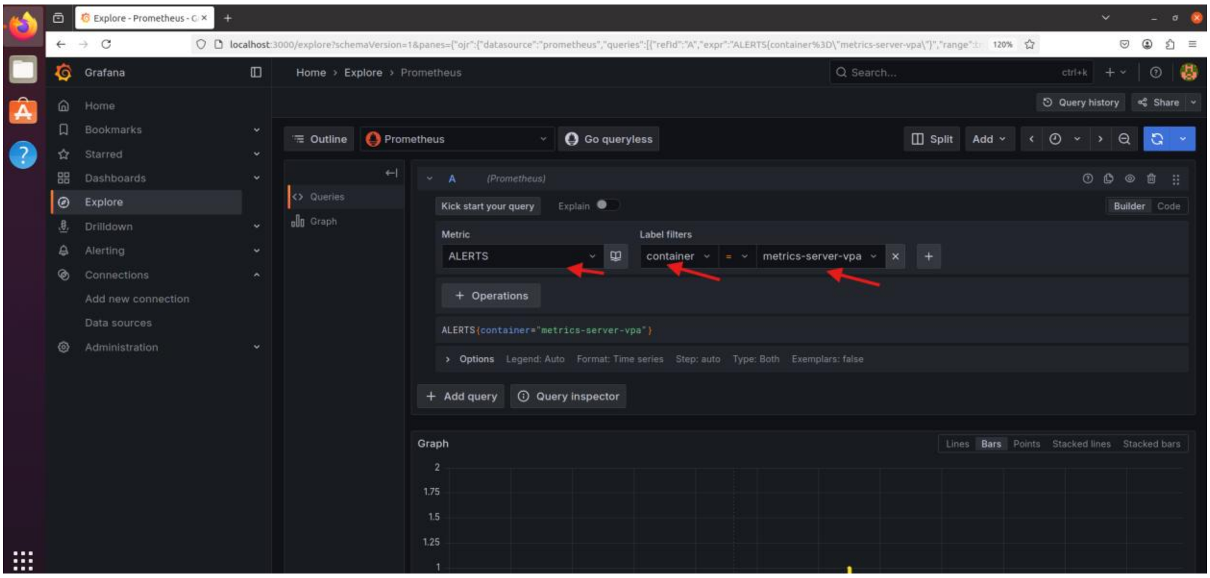
Go to **Dashboards → Manage**



Dashboards

Create and manage dashboards to visualize your data

Name	Tags
Alertmanager / Overview	alertmanager-mixin
CoreDNS	coredns dns
etcd	etcd-mixin
Grafana Overview	
Kubernetes / API server	kubernetes-mixin
Kubernetes / Compute Resources / Multi-Cluster	kubernetes-mixin
Kubernetes / Compute Resources / Cluster	kubernetes-mixin
Kubernetes / Compute Resources / Namespace (Pods)	kubernetes-mixin
Kubernetes / Compute Resources / Namespace (Workloads)	kubernetes-mixin
Kubernetes / Compute Resources / Node (Pods)	kubernetes-mixin
Kubernetes / Compute Resources / Pod	kubernetes-mixin

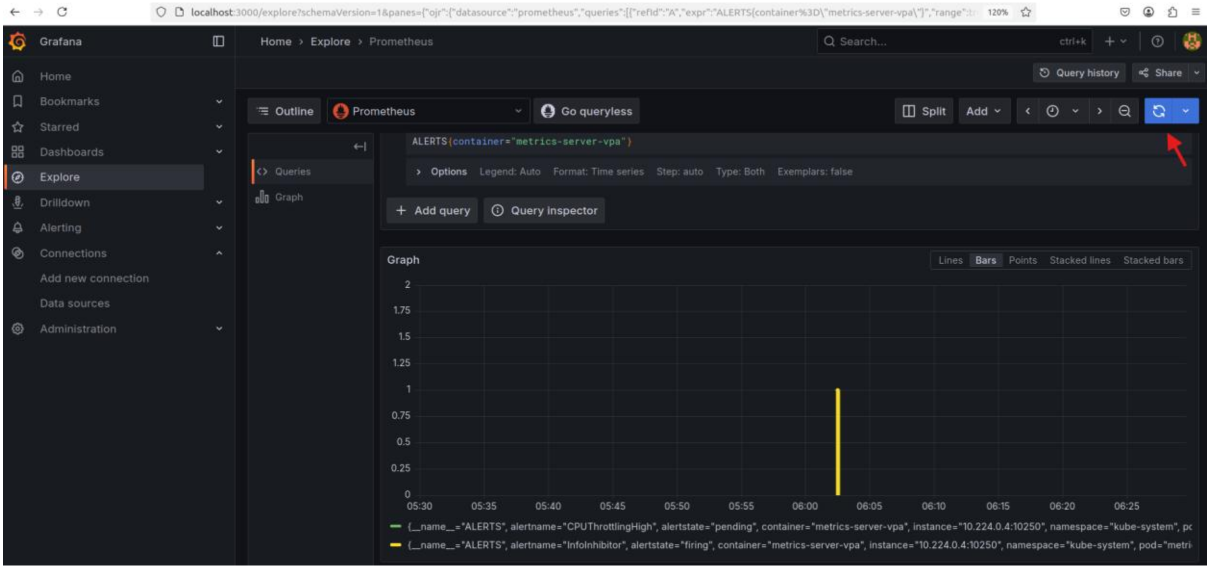


Metric: ALERTS

Label filters: container = metrics-server-vpa

```
ALERTS{container="metrics-server-vpa"}
```

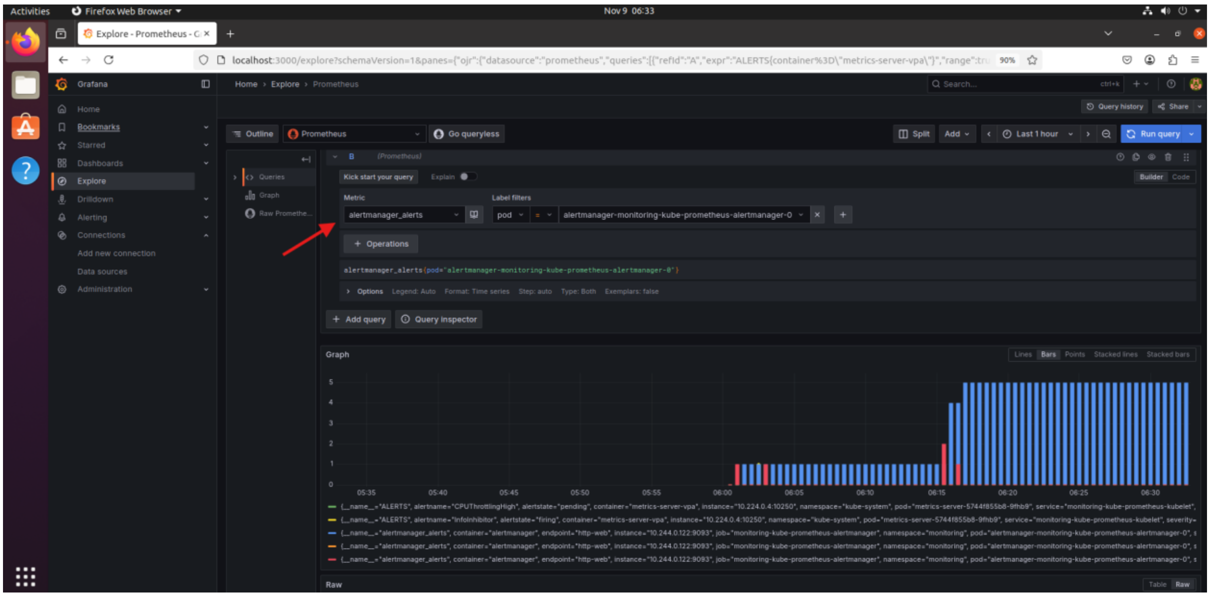
Options Legend: Auto Format: Time series Step: auto Type: Both Exemplars: false



```
ALERTS{container="metrics-server-vpa"}
```

Options Legend: Auto Format: Time series Step: auto Type: Both Exemplars: false

```
{__name__="ALERTS", alertname="CPUThrottlingHigh", alertstate="pending", container="metrics-server-vpa", instance="10.224.0.4:10250", namespace="kube-system", pc
{__name__="ALERTS", alertname="InfoInhibitor", alertstate="firing", container="metrics-server-vpa", instance="10.224.0.4:10250", namespace="kube-system", pod="metri
```



Metric: alertmanager_alerts

Label filters: pod = alertmanager-monitoring-kube-prometheus-alertmanager-0

```
alertmanager_alerts{pod="alertmanager-monitoring-kube-prometheus-alertmanager-0"}
```

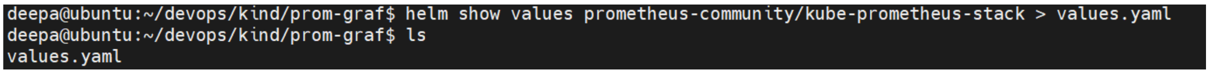
Options Legend: Auto Format: Time series Step: auto Type: Both Exemplars: false

```
{__name__="ALERTS", alertname="CPUThrottlingHigh", alertstate="pending", container="metrics-server-vpa", instance="10.224.0.4:10250", namespace="kube-system", pod="metrics-server-5744f855b8-9fhb9", service="monitoring-kube-prometheus-kubelet",
{__name__="ALERTS", alertname="InfoInhibitor", alertstate="firing", container="metrics-server-vpa", instance="10.224.0.4:10250", namespace="kube-system", pod="metrics-server-5744f855b8-9fhb9", service="monitoring-kube-prometheus-kubelet", severity=
{__name__="alertmanager_alerts", container="alertmanager", endpoint="http-web", instance="10.244.0.122:9093", job="monitoring-kube-prometheus-alertmanager", namespace="monitoring", pod="alertmanager-monitoring-kube-prometheus-alertmanager-0", s
{__name__="alertmanager_alerts", container="alertmanager", endpoint="http-web", instance="10.244.0.122:9093", job="monitoring-kube-prometheus-alertmanager", namespace="monitoring", pod="alertmanager-monitoring-kube-prometheus-alertmanager-0", s
{__name__="alertmanager_alerts", container="alertmanager", endpoint="http-web", instance="10.244.0.122:9093", job="monitoring-kube-prometheus-alertmanager", namespace="monitoring", pod="alertmanager-monitoring-kube-prometheus-alertmanager-0", s
```

alertmanager_alerts{container="alertmanager",endpoint="http-web",instance="10.244.0.122:9093",job="monitoring-kube-prometheus-alertmanager",namespace="monitoring",pod="alertmanager-monitoring-kube-prometheus-alertmanager-0",service="monitoring-kube-prometheus-alertmanager",state="active"}

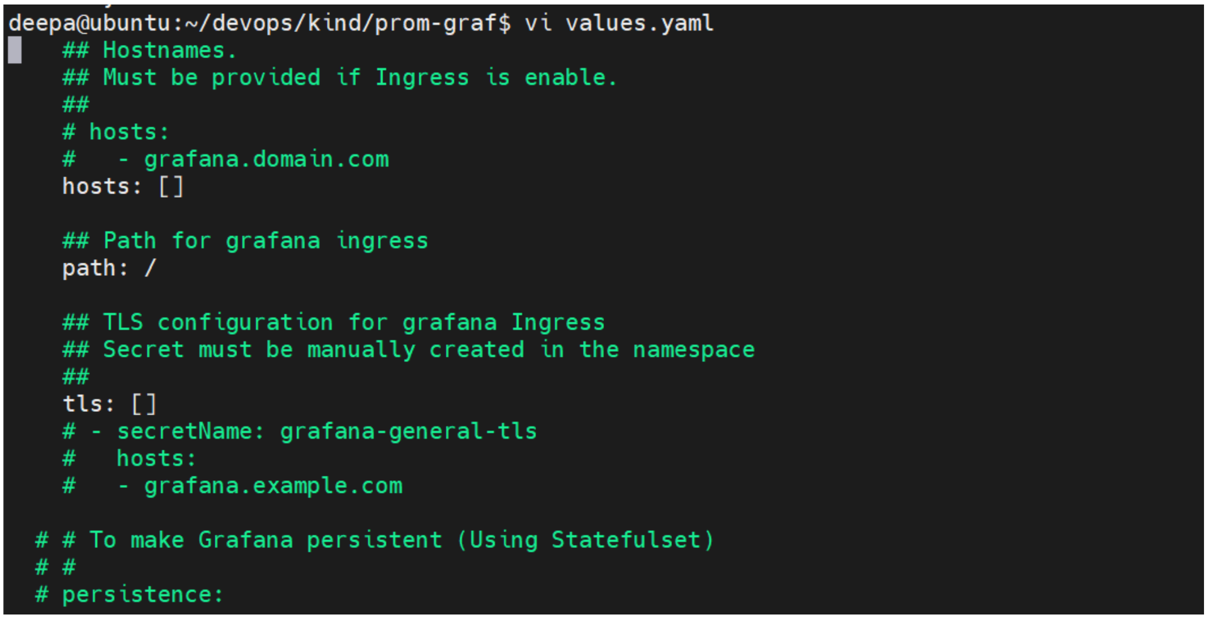
ADD query: we can add new queries.

To customize: Edit values.yaml

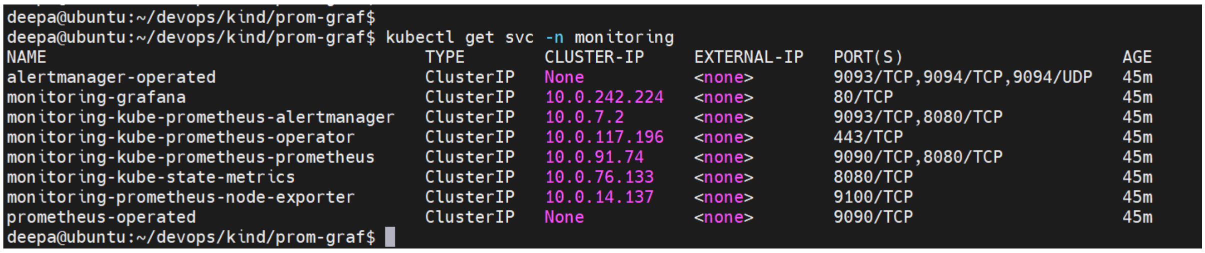


```
deepa@ubuntu:~/devops/kind/prom-graf$ helm show values prometheus-community/kube-prometheus-stack > values.yaml
deepa@ubuntu:~/devops/kind/prom-graf$ ls
values.yaml
```

Edit this file, then install:

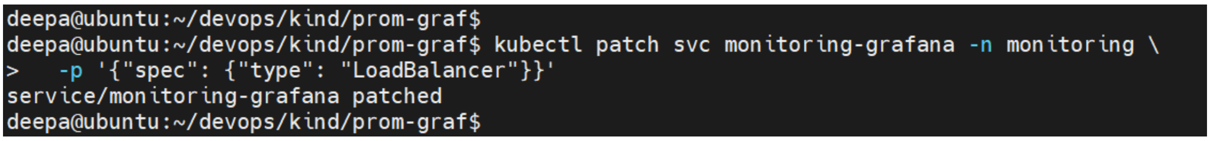


```
deepa@ubuntu:~/devops/kind/prom-graf$ vi values.yaml
    ## Hostnames.
    ## Must be provided if Ingress is enable.
    ##
    # hosts:
    #   - grafana.domain.com
    hosts: []

    ## Path for grafana ingress
    path: /

    ## TLS configuration for grafana Ingress
    ## Secret must be manually created in the namespace
    ##
    tls: []
    # - secretName: grafana-general-tls
    #   hosts:
    #   - grafana.example.com

  # # To make Grafana persistent (Using Statefulset)
  # #
  # persistence:
```

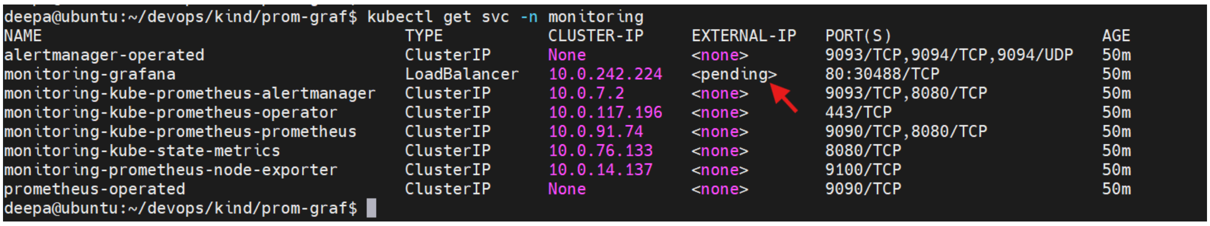


```
deepa@ubuntu:~/devops/kind/prom-graf$
deepa@ubuntu:~/devops/kind/prom-graf$ kubectl get svc -n monitoring
NAME                                      TYPE        CLUSTER-IP     EXTERNAL-IP   PORT(S)                      AGE
alertmanager-operated                     ClusterIP   None           <none>        9093/TCP,9094/TCP,9094/UDP   45m
monitoring-grafana                        ClusterIP   10.0.242.224   <none>        80/TCP                       45m
monitoring-kube-prometheus-alertmanager   ClusterIP   10.0.7.2       <none>        9093/TCP,8080/TCP            45m
monitoring-kube-prometheus-operator       ClusterIP   10.0.117.196   <none>        443/TCP                      45m
monitoring-kube-prometheus-prometheus     ClusterIP   10.0.91.74     <none>        9090/TCP,8080/TCP            45m
monitoring-kube-state-metrics             ClusterIP   10.0.76.133    <none>        8080/TCP                     45m
monitoring-prometheus-node-exporter       ClusterIP   10.0.14.137    <none>        9100/TCP                     45m
prometheus-operated                       ClusterIP   None           <none>        9090/TCP                     45m
deepa@ubuntu:~/devops/kind/prom-graf$
```

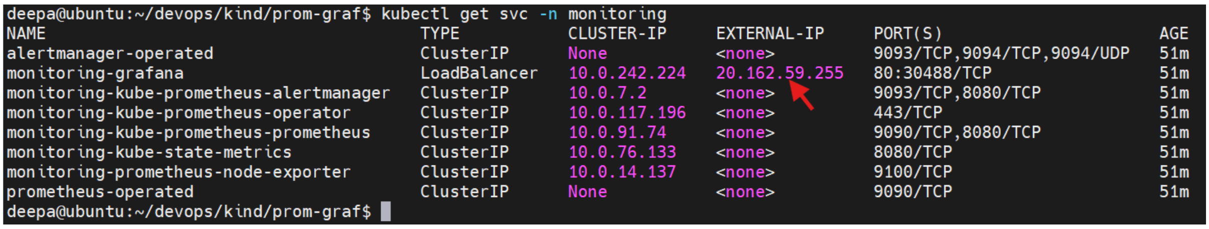
Change ClusterIP to LoadBalancer



```
deepa@ubuntu:~/devops/kind/prom-graf$
deepa@ubuntu:~/devops/kind/prom-graf$ kubectl patch svc monitoring-grafana -n monitoring \
>   -p '{"spec": {"type": "LoadBalancer"}}'
service/monitoring-grafana patched
deepa@ubuntu:~/devops/kind/prom-graf$
```



```
deepa@ubuntu:~/devops/kind/prom-graf$ kubectl get svc -n monitoring
NAME                                      TYPE           CLUSTER-IP     EXTERNAL-IP   PORT(S)                      AGE
alertmanager-operated                     ClusterIP      None           <none>        9093/TCP,9094/TCP,9094/UDP   50m
monitoring-grafana                        LoadBalancer   10.0.242.224   <pending>     80:30488/TCP                 50m
monitoring-kube-prometheus-alertmanager   ClusterIP      10.0.7.2       <none>        9093/TCP,8080/TCP            50m
monitoring-kube-prometheus-operator       ClusterIP      10.0.117.196   <none>        443/TCP                      50m
monitoring-kube-prometheus-prometheus     ClusterIP      10.0.91.74     <none>        9090/TCP,8080/TCP            50m
monitoring-kube-state-metrics             ClusterIP      10.0.76.133    <none>        8080/TCP                     50m
monitoring-prometheus-node-exporter       ClusterIP      10.0.14.137    <none>        9100/TCP                     50m
prometheus-operated                       ClusterIP      None           <none>        9090/TCP                     50m
deepa@ubuntu:~/devops/kind/prom-graf$
```



```
deepa@ubuntu:~/devops/kind/prom-graf$ kubectl get svc -n monitoring
NAME                                      TYPE           CLUSTER-IP     EXTERNAL-IP     PORT(S)                      AGE
alertmanager-operated                     ClusterIP      None           <none>          9093/TCP,9094/TCP,9094/UDP   51m
monitoring-grafana                        LoadBalancer   10.0.242.224   20.162.59.255   80:30488/TCP                 51m
monitoring-kube-prometheus-alertmanager   ClusterIP      10.0.7.2       <none>          9093/TCP,8080/TCP            51m
monitoring-kube-prometheus-operator       ClusterIP      10.0.117.196   <none>          443/TCP                      51m
monitoring-kube-prometheus-prometheus     ClusterIP      10.0.91.74     <none>          9090/TCP,8080/TCP            51m
monitoring-kube-state-metrics             ClusterIP      10.0.76.133    <none>          8080/TCP                     51m
monitoring-prometheus-node-exporter       ClusterIP      10.0.14.137    <none>          9100/TCP                     51m
prometheus-operated                       ClusterIP      None           <none>          9090/TCP                     51m
deepa@ubuntu:~/devops/kind/prom-graf$
```

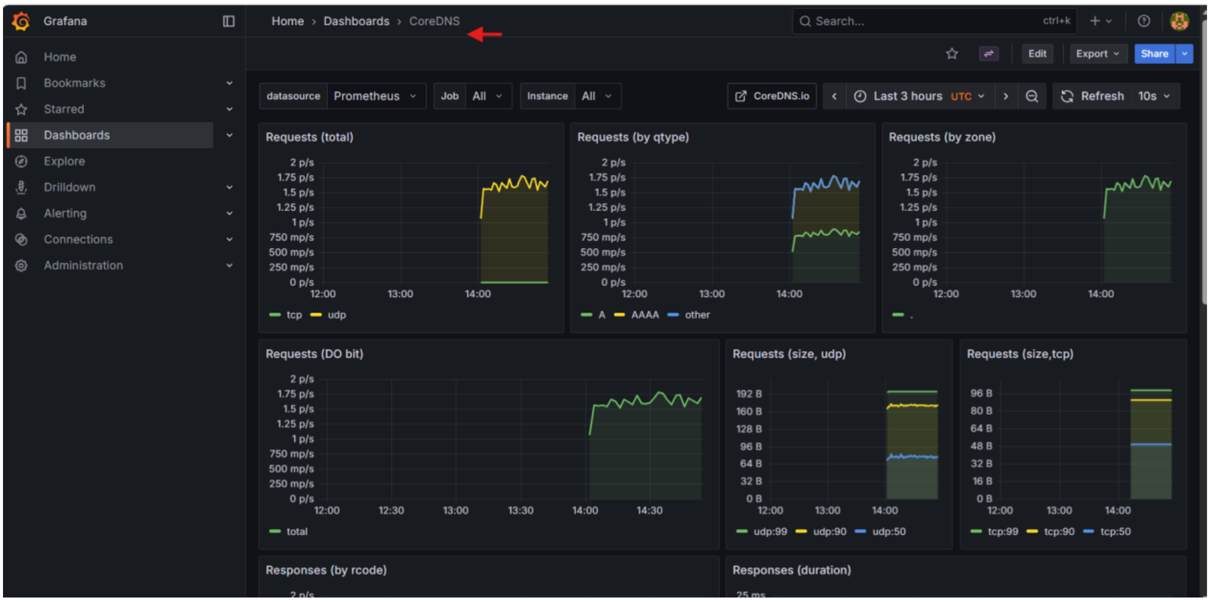
http://<grafana-external-ip>

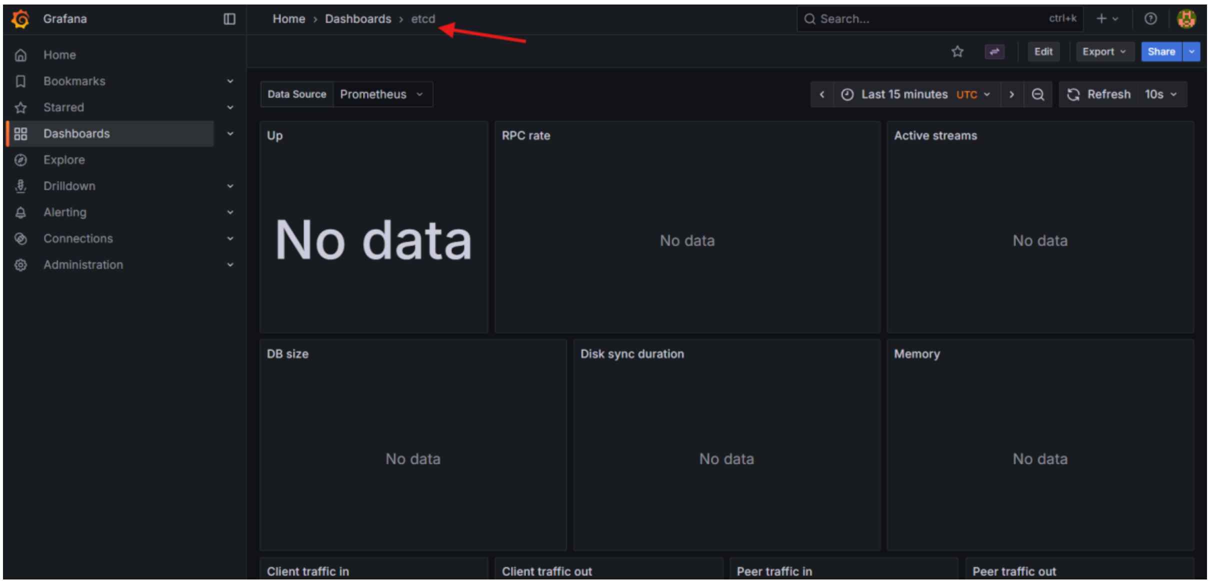
http://20.162.59.255

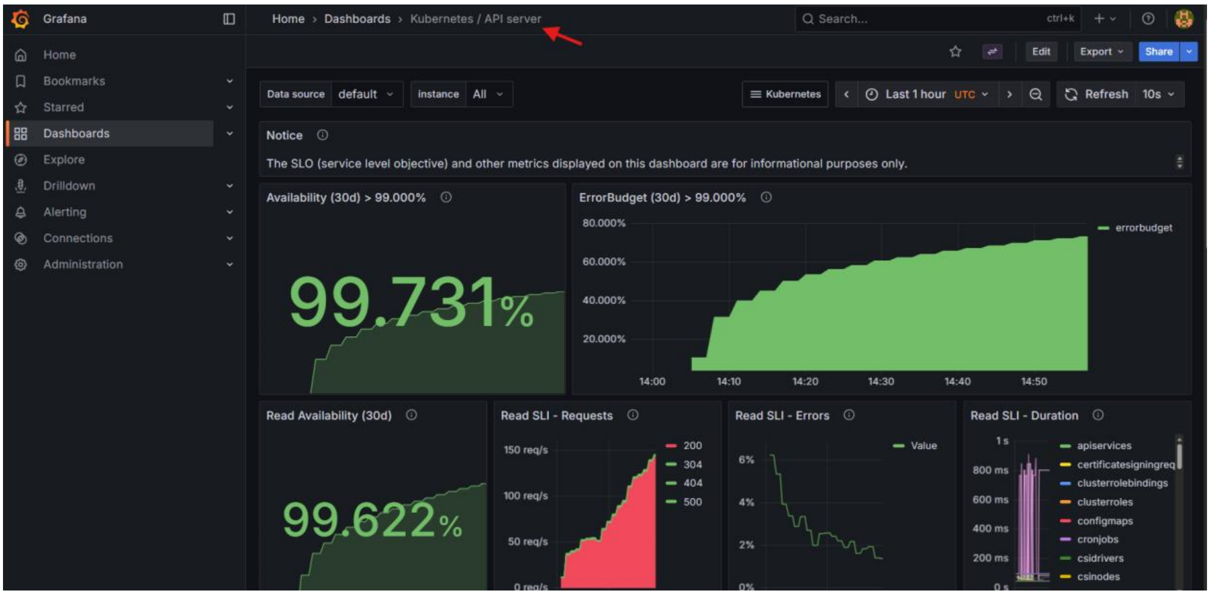
Username: admin

Password: earlier obtained

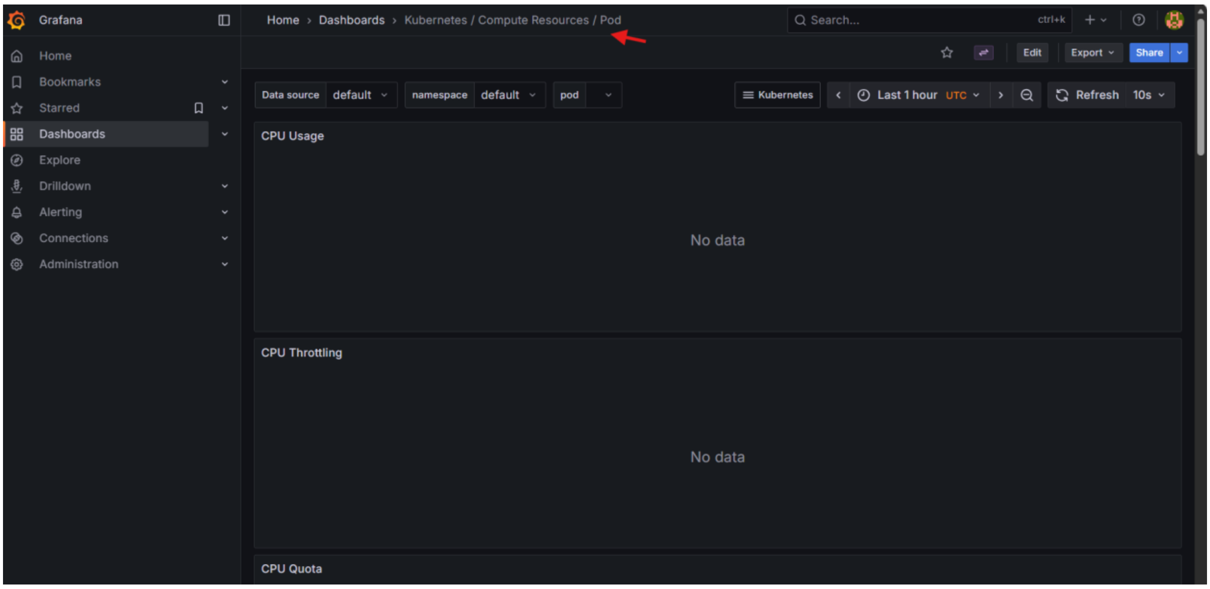
Go to Dashboard

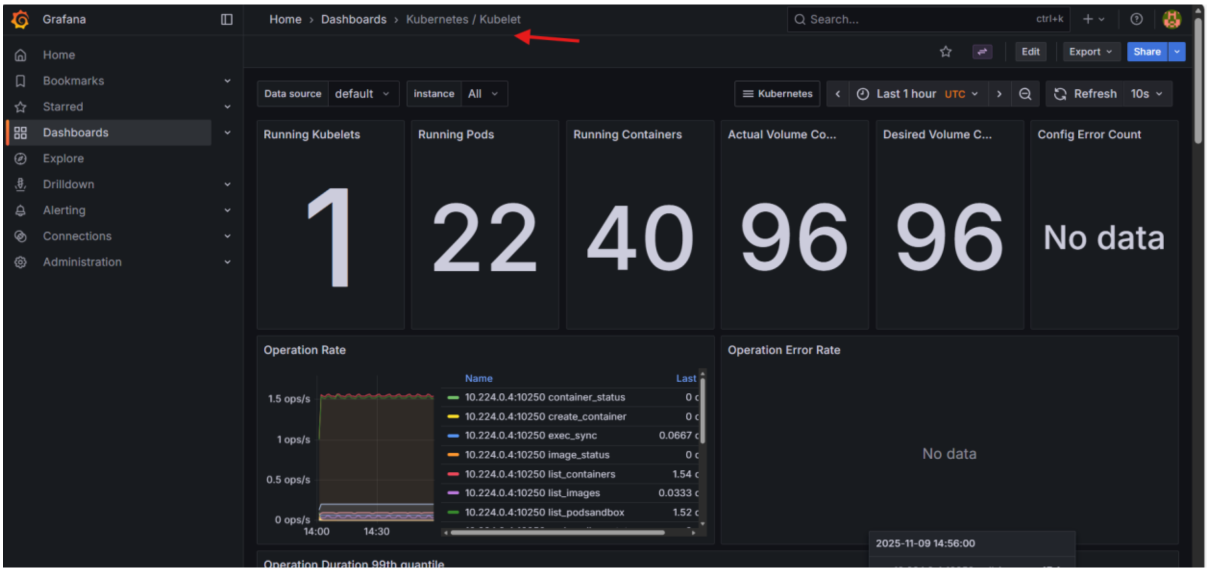






The SLO (service level objective) and other metrics displayed on this dashboard are for informational purposes only.





Data source: default | instance: All | Last 1 hour UTC

Running Kubelets	Running Pods	Running Containers	Actual Volume Co...	Desired Volume C...	Config Error Count
1	22	40	96	96	No data

Operation Rate

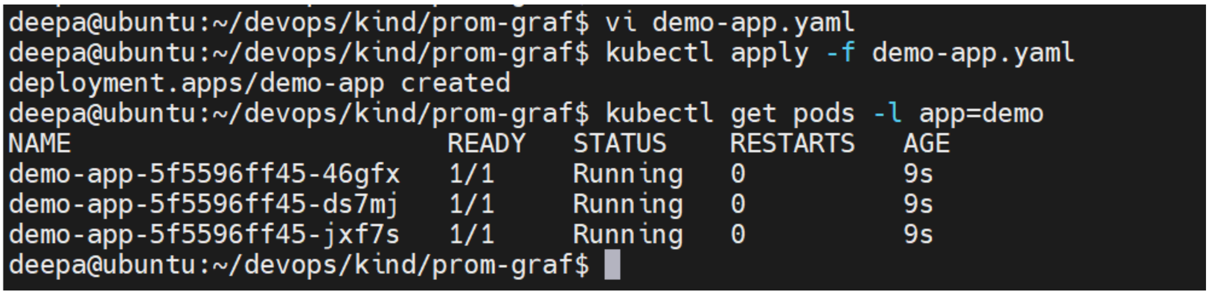
Name	Last
10.224.0.4:10250 container_status	0 c
10.224.0.4:10250 create_container	0 c
10.224.0.4:10250 exec_sync	0.0667 c
10.224.0.4:10250 image_status	0 c
10.224.0.4:10250 list_containers	1.54 c
10.224.0.4:10250 list_images	0.0333 c
10.224.0.4:10250 list_podsandbox	1.52 c

Operation Error Rate

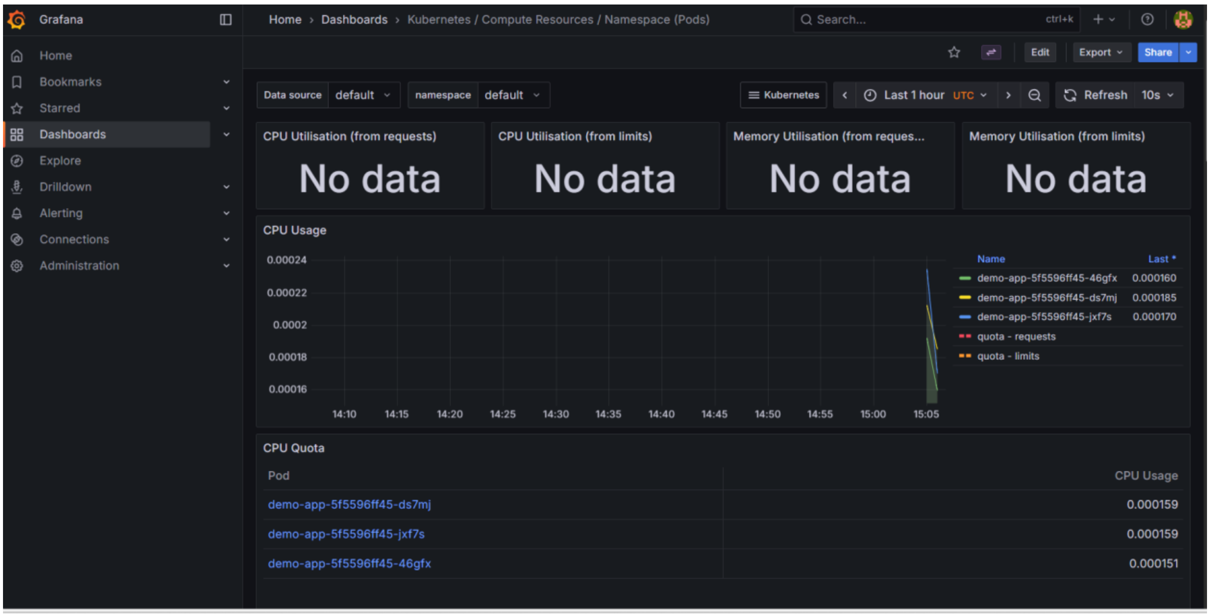
No data

2025-11-09 14:56:00

Operation Duration 99th quantile



```
deepa@ubuntu:~/devops/kind/prom-graf$ vi demo-app.yaml
deepa@ubuntu:~/devops/kind/prom-graf$ kubectl apply -f demo-app.yaml
deployment.apps/demo-app created
deepa@ubuntu:~/devops/kind/prom-graf$ kubectl get pods -l app=demo
NAME                        READY   STATUS    RESTARTS   AGE
demo-app-5f5596ff45-46gfx   1/1     Running   0          9s
demo-app-5f5596ff45-ds7mj   1/1     Running   0          9s
demo-app-5f5596ff45-jxf7s   1/1     Running   0          9s
deepa@ubuntu:~/devops/kind/prom-graf$ 
```



Data source: default | namespace: default | Last 1 hour UTC

CPU Utilisation (from requests)	CPU Utilisation (from limits)	Memory Utilisation (from reques...	Memory Utilisation (from limits)
No data	No data	No data	No data

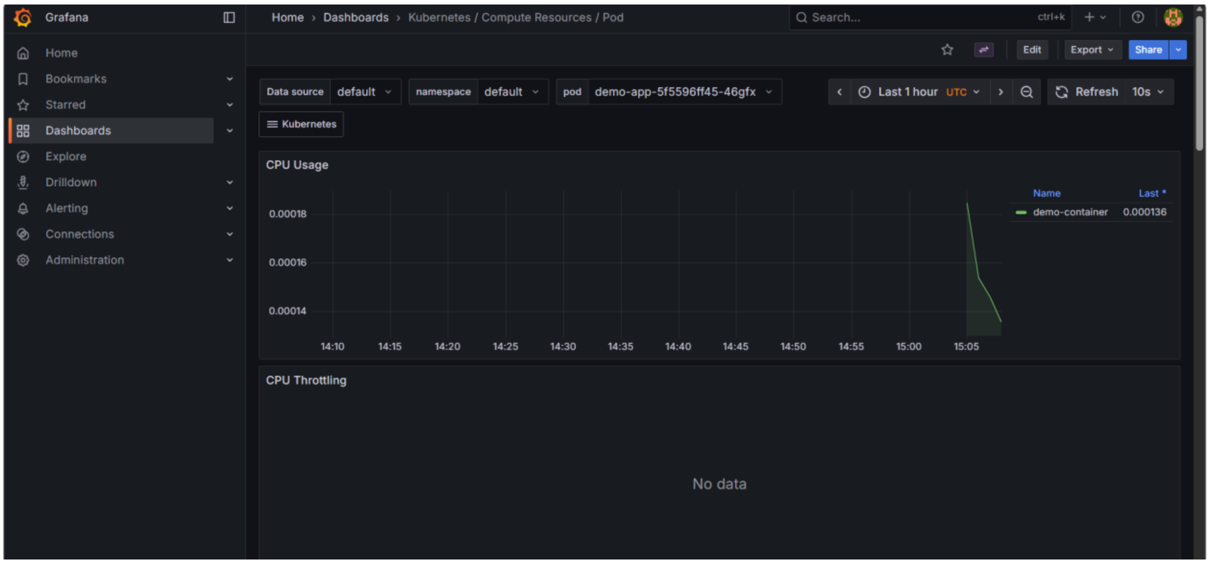
CPU Usage

Name	Last *
demo-app-5f5596ff45-46gfx	0.000160
demo-app-5f5596ff45-ds7mj	0.000185
demo-app-5f5596ff45-jxf7s	0.000170
quota - requests	
quota - limits	

CPU Quota

Pod	CPU Usage
demo-app-5f5596ff45-ds7mj	0.000159
demo-app-5f5596ff45-jxf7s	0.000159
demo-app-5f5596ff45-46gfx	0.000151

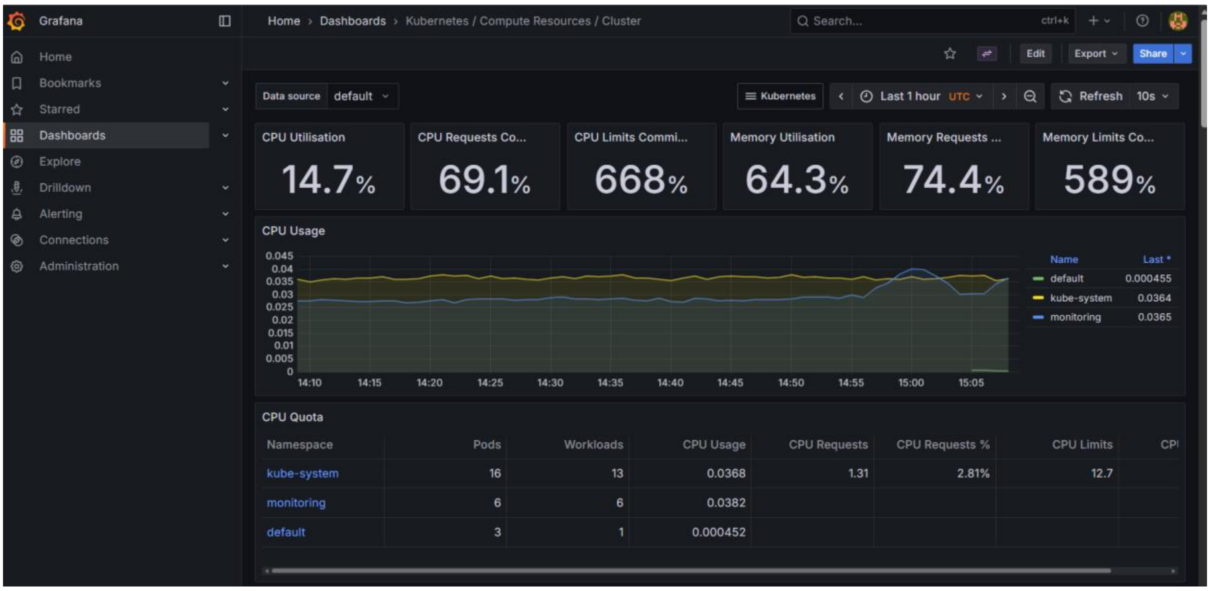
Earlier it was empty:



Home › Dashboards › Kubernetes / Compute Resources / Pod

Data source: default | namespace: default | pod: demo-app-5f5596ff45-46gfx | Last 1 hour UTC | Refresh 10s

Kubernetes / Compute Resources / Cluster



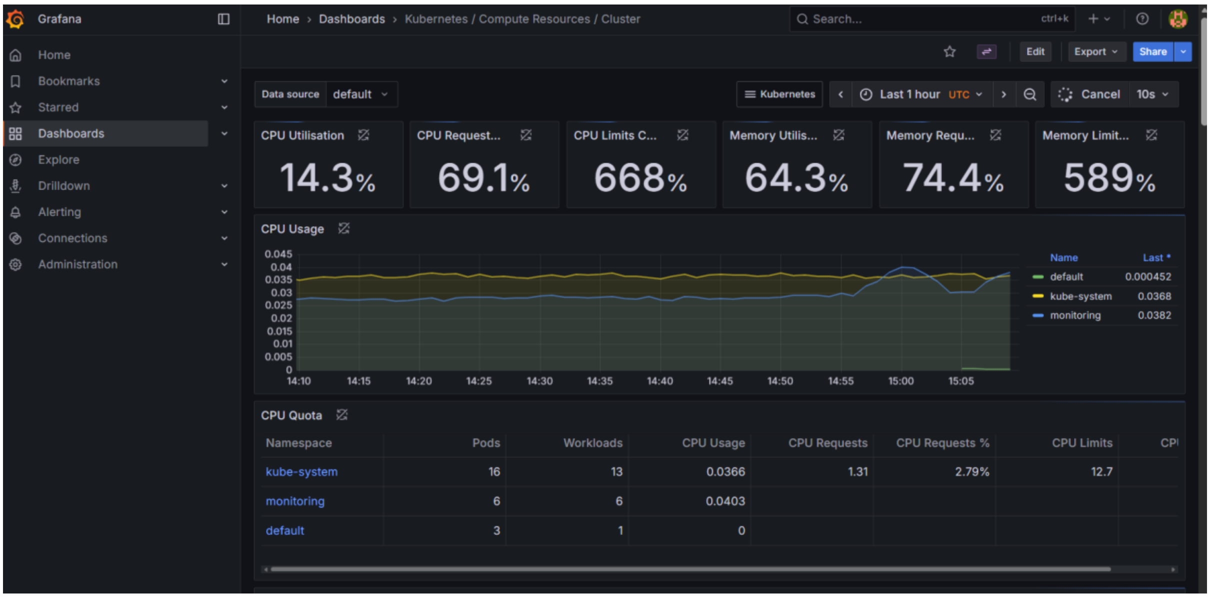
Home › Dashboards › Kubernetes / Compute Resources / Cluster

Data source: default | Last 1 hour UTC | Refresh 10s

CPU Utilisation	CPU Requests Co...	CPU Limits Commi...	Memory Utilisation	Memory Requests ...	Memory Limits Co...
14.7%	69.1%	668%	64.3%	74.4%	589%

CPU Quota

Namespace	Pods	Workloads	CPU Usage	CPU Requests	CPU Requests %	CPU Limits	CPI
kube-system	16	13	0.0368	1.31	2.81%	12.7	
monitoring	6	6	0.0382				
default	3	1	0.000452				



Home › Dashboards › Kubernetes / Compute Resources / Cluster

Data source: default | Last 1 hour UTC | Cancel 10s

CPU Utilisation	CPU Request...	CPU Limits C...	Memory Utilis...	Memory Requ...	Memory Limit...
14.3%	69.1%	668%	64.3%	74.4%	589%

CPU Quota

Namespace	Pods	Workloads	CPU Usage	CPU Requests	CPU Requests %	CPU Limits	CPI
kube-system	16	13	0.0366	1.31	2.79%	12.7	
monitoring	6	6	0.0403				
default	3	1	0				

```
az aks delete -n aks-deepa-cluster -g rg-deepa --yes
```