

MONITORING IN K8S

Prometheus and Grafana are two popular open-source tools commonly used in conjunction with Kubernetes (K8s) for monitoring and visualization of applications and infrastructure.

Prometheus:

Prometheus in Kubernetes is a monitoring tool that helps keep an eye on the health and performance of applications and the overall system running in a Kubernetes cluster.

Grafana:

Grafana in Kubernetes (K8s) is a tool used for creating visual dashboards to monitor and analyze the performance of applications and infrastructure within a Kubernetes cluster.

STEPS TO SETUP PROMETHEUS & GRAFANA IN KOPS:

INSTALL HELM:

- `curl -fsSL -o get_helm.sh https://raw.githubusercontent.com/helm/helm/main/scripts/get-helm-3`
- `chmod 700 get_helm.sh`
- `./get_helm.sh`
- `helm version`

INSTALL K8S METRICS SERVER:

- `kubectl apply -f https://github.com/kubernetes-sigs/metrics-server/releases/latest/download/components.yaml`

Verify that the metrics-server deployment is running the desired number of pods

- `kubectl get pods -n kube-system`
- `kubectl get deployment metrics-server -n kube-system`

INSTALL PROMETHEUS:

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

UPDATE HELM CHART REPOS:

- helm repo update
- helm repo list

CREATE PROMETHEUS NAMESPACE:

- kubectl create namespace prometheus
- kubectl get ns

INSTALL PROMETHEUS:

- helm install prometheus prometheus-community/prometheus --namespace prometheus --set alertmanager.persistentVolume.storageClass="gp2" --set server.persistentVolume.storageClass="gp2"
- kubectl get pods -n prometheus
- kubectl get all -n prometheus

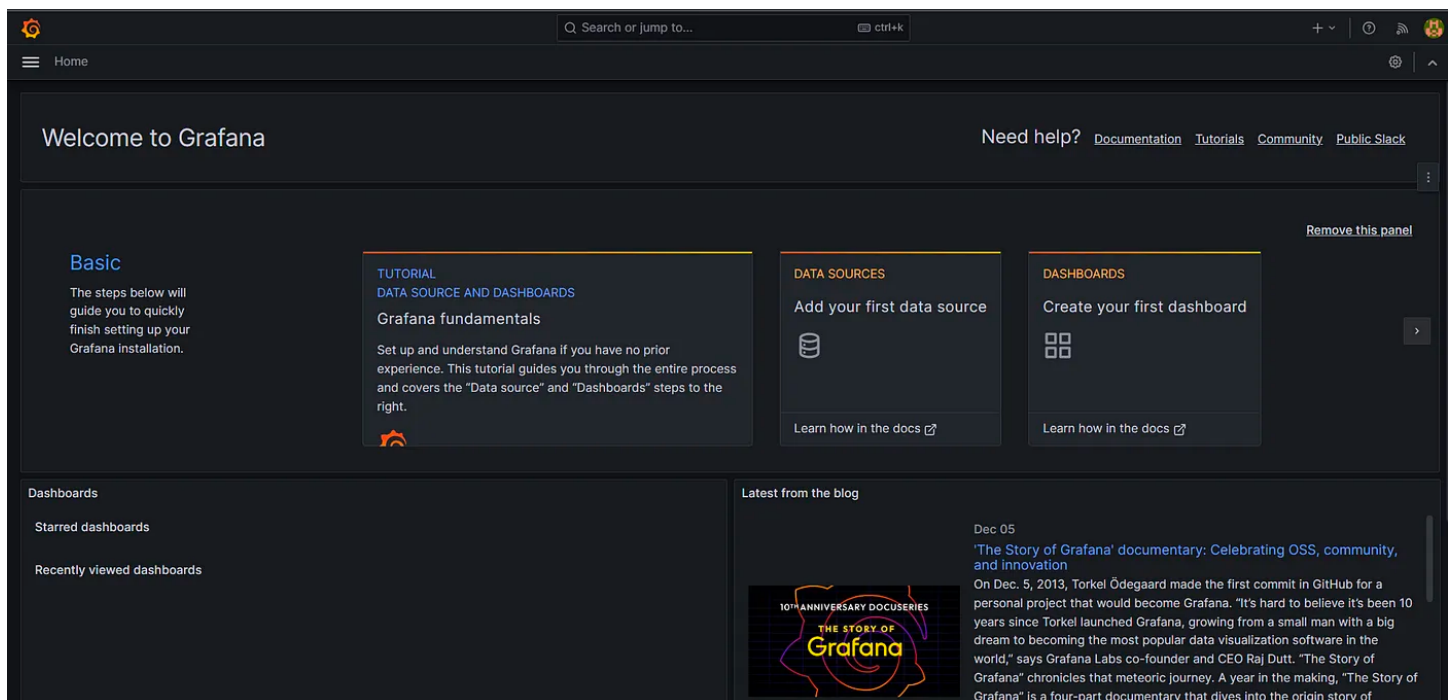
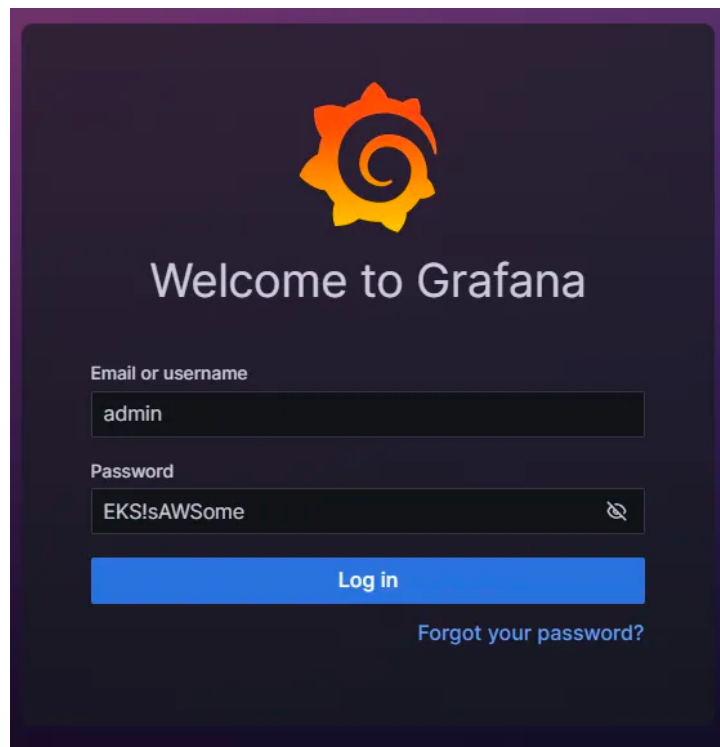
CREATE GRAFANA NAMESPACE:

- kubectl create namespace grafana

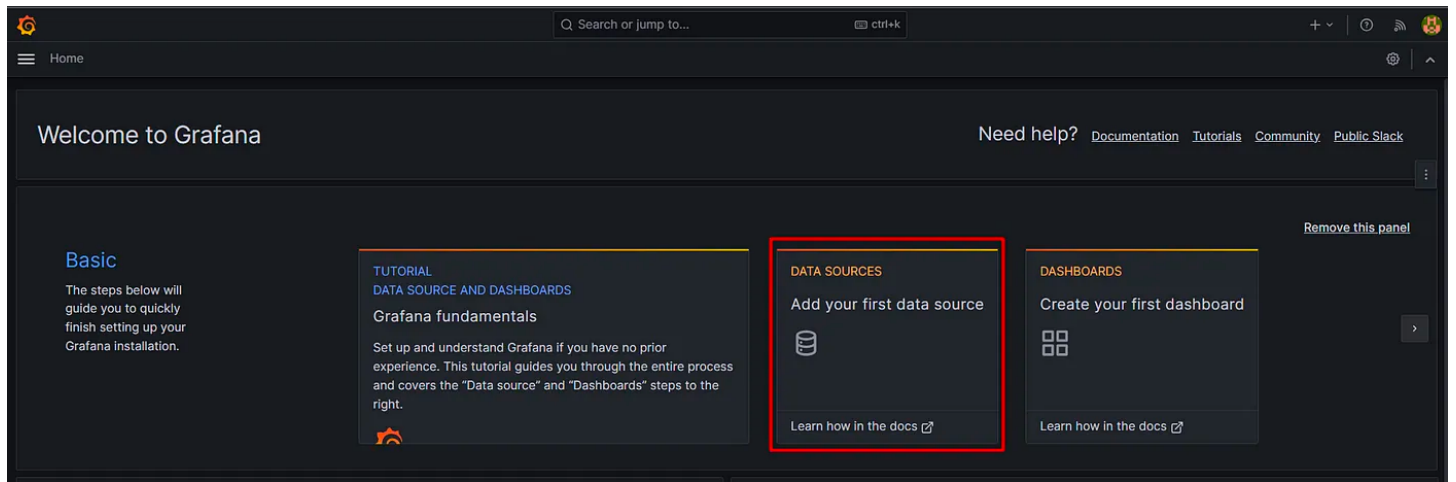
INSTALL GRAFANA:

- helm repo add grafana <https://grafana.github.io/helm-charts>
- helm install grafana grafana/grafana --namespace grafana --set persistence.storageClassName="gp2" --set persistence.enabled=true --set adminPassword='EKS!sAWSome' --set service.type=LoadBalancer
- kubectl get pods -n grafana
- kubectl get service -n grafana

Copy the EXTERNAL-IP and paste in browser

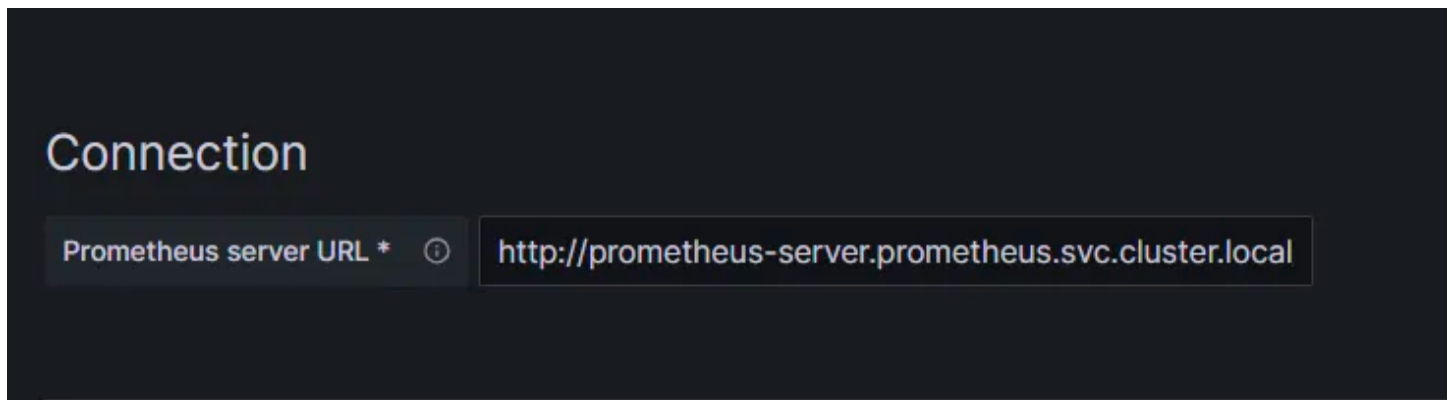


Go to Grafana Dashboard → Add the Datasource → Select the Prometheus



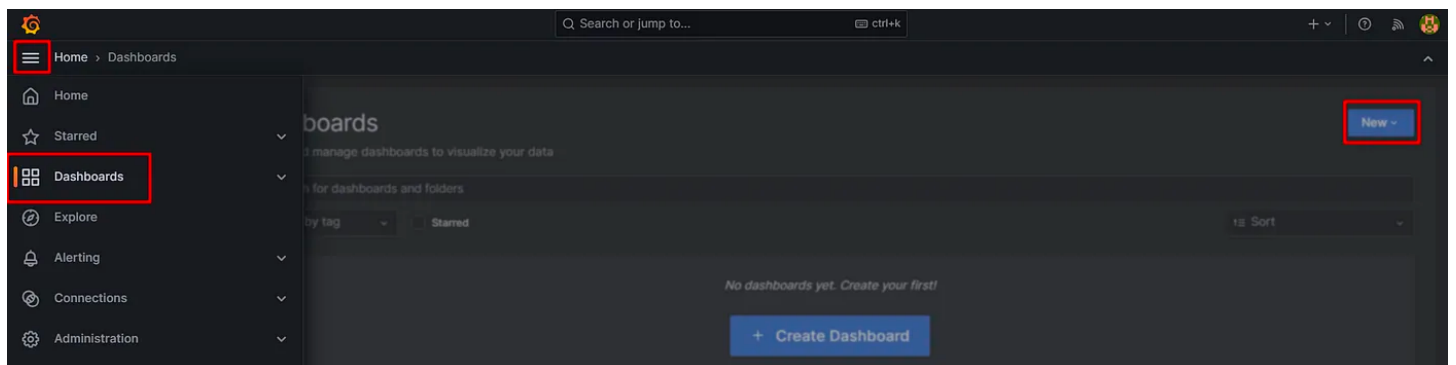
add the below *url* in **Connection** and **save and test**

<http://prometheus-server.prometheus.svc.cluster.local/>



Import Grafana dashboard from Grafana Labs

grafana dashboard → new → Import → 6417 → load → select prometheus → import



Import dashboard

Import dashboard from file or Grafana.com



Upload dashboard JSON file

Drag and drop here or click to browse

Accepted file types: .json, .txt

Find and import dashboards for common applications at grafana.com/dashboards

6417

Load

Import via dashboard JSON model

```
{
  "title": "Example - Repeating Dictionary variables",
  "uid": "_OHnEoN4z",
  "panels": [...]
  ...
}
```

Load

Cancel

Import dashboard

Import dashboard from file or Grafana.com

Importing dashboard from Grafana.com

Published by

sekka1

Updated on

2018-06-07 05:21:56

Options

Name

Kubernetes Cluster (Prometheus)

Folder

Dashboards


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.

4XuMd2liz

Change uid

prometheus

 Prometheus

Import

Cancel

NOW DEPLOY ANY APPLICATION AND SEE THE RESULT IN DASHBOARD.

ADD 315 PORT TO MONITOR THE FOLLOWING TERMS:

- Network I/O pressure.
- Cluster CPU usage.
- Cluster Memory usage.
- Cluster filesystem usage.
- Pods CPU usage.

ADD 1860 PORT TO MONITOR NODES INDIVIDUALLY