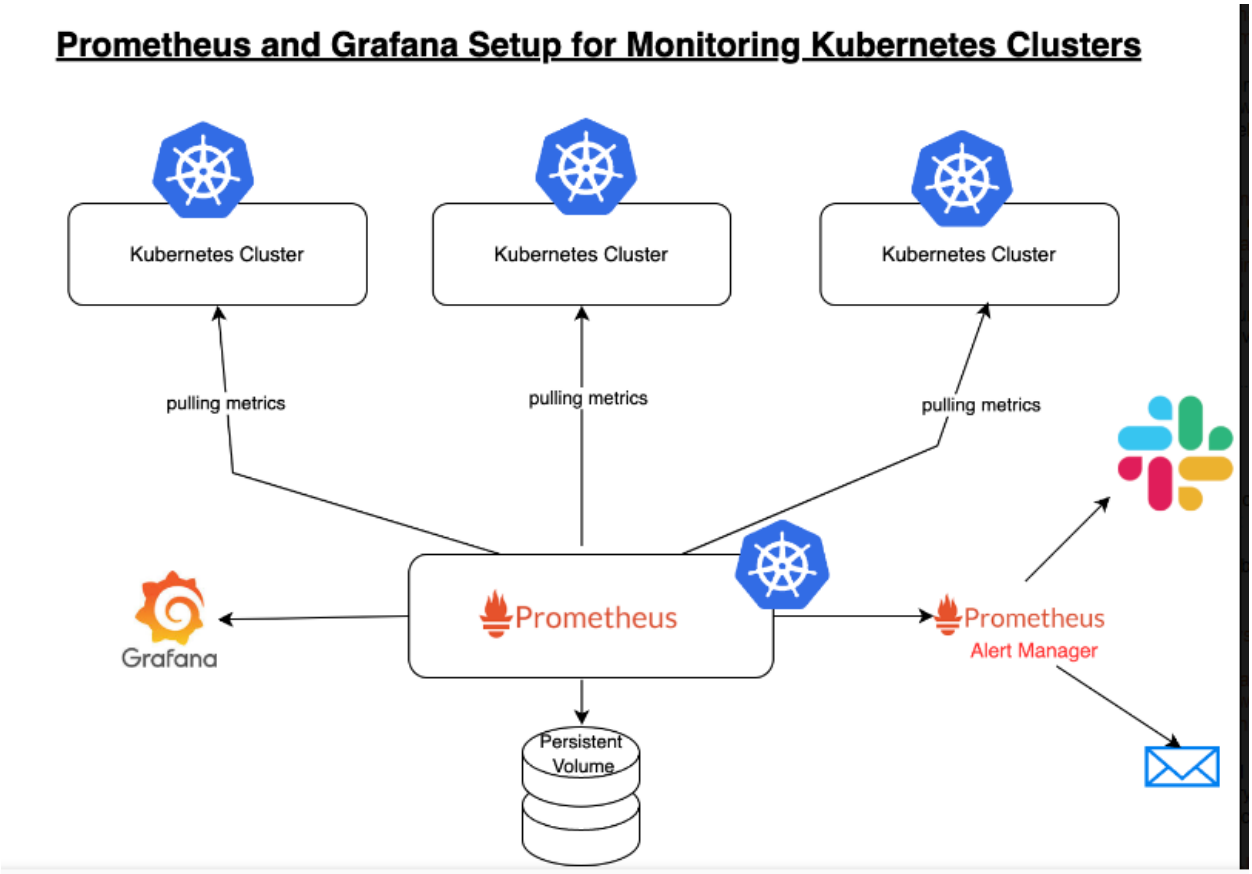
How to setup monitoring on Kubernetes Cluster using Prometheus and Grafana | Setup monitoring on EKS Cluster using Prometheus and Grafana

What is Prometheus?

- Prometheus is an open source monitoring tool and alerting tool.
- Provides out-of-the-box monitoring capabilities for the Kubernetes container orchestration platform. It can monitor servers and databases as well.
- Collects and stores metrics as time-series data, recording information with a timestamp
- It is based on pull and collects metrics from targets by scraping metrics HTTP endpoints.

What is Grafana?

- Grafana is an open source visualization and analytics software.
- It allows you to query, visualize, alert on, and explore your metrics no matter where they are stored.



Key components:

1. Prometheus server - Processes and stores metrics data
2. Alert Manager - Sends alerts to any systems/channels
3. Grafana - Visualize scraped data in UI

Installation Method

There are many ways you can set up Prometheus and Grafana. You can install in following ways:

1. Create all configuration files of both Prometheus and Grafana and execute them in the right order.

2. Prometheus Operator - to simplify and automate the configuration and management of the Prometheus monitoring stack running on a Kubernetes cluster

3. Helm chart (Recommended) - Using helm to install Prometheus Operator including Grafana

Why use Helm?

Helm is a package manager for Kubernetes. Helm simplifies the installation of all components in one command. Install using Helm is recommended as you will not be missing any configuration steps and very efficient.

In helm 3 there is no tiller component. Helm client directly interacts with the Kubernetes API for the helm chart deployment.

Helm 3 can be installed many ways. We will install Helm 3 using the scripts option.

Download scripts

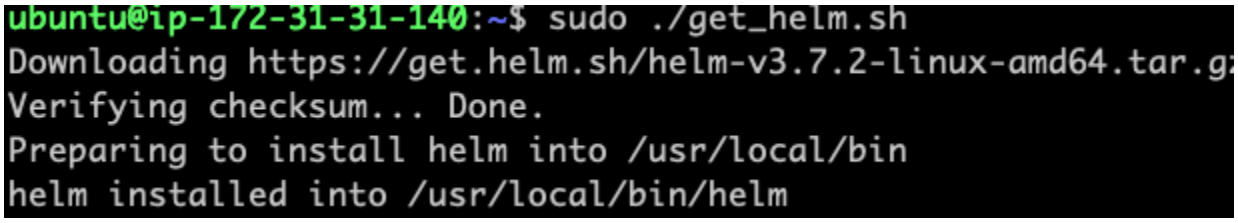
curl -fsSL -o get_helm.sh https://raw.githubusercontent.com/helm/helm/master/scripts/get-helm-3

provide permission

sudo chmod 700 get_helm.sh

Execute script to install

sudo ./get_helm.sh



Verify installation

helm version --client

Prerequisites

- kubernetes cluster is setup already
- Install Helm (already installed above)
- EC2 instance to access EKS cluster

Implementation steps

We need to add the Helm Stable Charts for your local client. Execute the below.

command:

helm repo add stable https://charts.helm.sh/stable

```
ubuntu@ip-172-31-6-100:~$ helm repo add stable https://charts.helm.sh/stable
"stable" has been added to your repositories
ubuntu@ip-172-31-6-100:~$
```

Add prometheus Helm repo

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

```
ubuntu@ip-172-31-6-100:~$ helm repo add prometheus-community https://prometheus-community.github.io/helm-charts
"prometheus-community" has been added to your repositories
```

helm search repo prometheus-community

Prometheus and grafana helm chart moved to kube prometheus stack

```
ubuntu@ip-172-31-8-184:~$ helm search repo prometheus-community
NAME                                            CHART VERSION   APP VERSION     DESCRIPTION

prometheus-community/alertmanager               0.18.0          v0.23.0         The Alertmanager handles alerts sent by cl
ient ...
prometheus-community/kube-prometheus-stack      35.2.0          0.56.2          kube-prometheus-stack collects Kubernetes
manif...
prometheus-community/kube-state-metrics         4.7.0           2.4.1           Install kube-state-metrics to generate and
 expo...
prometheus-community/prometheus                 15.8.7          2.34.0          Prometheus is a monitoring system and time
 seri...
prometheus-community/prometheus-adapter         3.2.2           v0.9.1          A Helm chart for k8s prometheus adapter

prometheus-community/prometheus-blackbox-exporter 5.8.0         0.20.0          Prometheus Blackbox Exporter
```

Create Prometheus namespace

kubectl create namespace prometheus

Below is the helm command to install kube-prometheus-stack. The helm repo kube-stack-prometheus (formerly prometheus-operator) comes with a grafana deployment embedded.

helm install stable prometheus-community/kube-prometheus-stack -n prometheus

```
ubuntu@ip-172-31-30-237:~$ helm install stable prometheus-community/kube-prometheus-stack -n prometheus



NAME: stable
LAST DEPLOYED: Fri May 20 23:14:54 2022
NAMESPACE: prometheus
STATUS: deployed
REVISION: 1
NOTES:
kube-prometheus-stack has been installed. Check its status by running:
  kubectl --namespace prometheus get pods -l "release=stable"
```

NOTE:

If you encounter an error, run: **`curl -L https://git.io/get_helm.sh | bash -s -- --version v3.8.2`**

Lets check if prometheus and grafana pods are running already

kubectl get pods -n prometheus

```
jenkins@ip-172-31-4-244:~$   kubectl get pods --namespace prometheus
NAME                                                     READY   STATUS    RESTARTS   AGE
alertmanager-stable-kube-prometheus-sta-alertmanager-0   2/2     Running   0          21m
prometheus-stable-kube-prometheus-sta-prometheus-0       2/2     Running   0          21m
stable-grafana-cbb58869b-8kb4g                           3/3     Running   0          21m
stable-kube-prometheus-sta-operator-6b565466fb-vr6f4     1/1     Running   0          21m
stable-kube-state-metrics-65bcb89bd9-j8hg4               1/1     Running   0          21m
stable-prometheus-node-exporter-grhwj                    1/1     Running   0          21m
stable-prometheus-node-exporter-qcjp4                    1/1     Running   0          21m
jenkins@ip-172-31-4-244:~$
```

kubectl get svc -n prometheus

```
ubuntu@ip-172-31-30-237:~$ kubectl get svc -n prometheus
NAME                                      TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)                      AGE
alertmanager-operated                     ClusterIP   None             <none>        9093/TCP,9094/TCP,9094/UDP   2m45s
prometheus-operated                       ClusterIP   None             <none>        9090/TCP                     2m45s
stable-grafana                            ClusterIP   10.100.4.90      <none>        80/TCP                       2m49s
stable-kube-prometheus-sta-alertmanager   ClusterIP   10.100.148.169   <none>        9093/TCP                     2m49s
stable-kube-prometheus-sta-operator       ClusterIP   10.100.64.78     <none>        443/TCP                      2m49s
stable-kube-prometheus-sta-prometheus     ClusterIP   10.100.121.15    <none>        9090/TCP                     2m49s
stable-kube-state-metrics                 ClusterIP   10.100.19.32     <none>        8080/TCP                     2m49s
stable-prometheus-node-exporter           ClusterIP   10.100.9.3       <none>        9100/TCP                     2m49s
```

This confirms that prometheus and grafana have been installed successfully using Helm.

In order to make prometheus and grafana available outside the cluster, use load balancer or NodePort.

Edit Prometheus Service

kubectl edit svc stable-kube-prometheus-sta-prometheus -n prometheus

```
  selector:
    app.kubernetes.io/name: prometheus
    prometheus: stable-kube-prometheus-sta-prometheus
  sessionAffinity: None
  type: LoadBalancer
status:
  loadBalancer: {}
```

Edit Grafana Service

kubectl edit svc stable-grafana -n prometheus

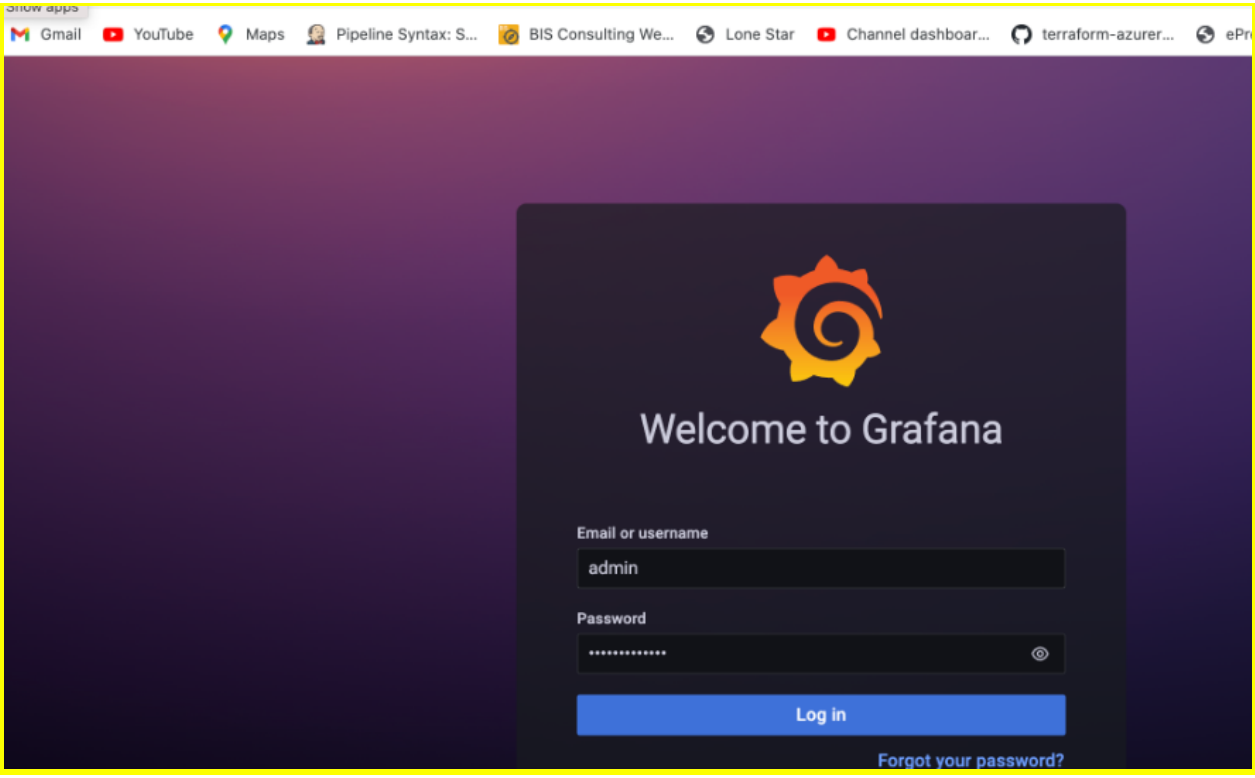
Verify if service is changed to LoadBalancer and also to get the Load Balancer URL.

kubectl get svc -n prometheus

```
jenkins@ip-172-31-4-244:~$   kubectl get svc --namespace prometheus
NAME                                      TYPE           CLUSTER-IP       EXTERNAL-IP                                                                PORT(S)                      AGE
alertmanager-operated                     ClusterIP      None             <none>                                                                     9093/TCP,9094/TCP,9094/UDP   22m
prometheus-operated                       ClusterIP      None             <none>                                                                     9090/TCP                     22m
stable-grafana                            LoadBalancer   10.100.145.244   a18e78f997d1b43a280ab79b43102bc1-1857160455.us-east-2.elb.amazonaws.com    80:31794/TCP                 22m
stable-kube-prometheus-sta-alertmanager   ClusterIP      10.100.202.80    <none>                                                                     9093/TCP                     22m
stable-kube-prometheus-sta-operator       ClusterIP      10.100.27.36     <none>                                                                     443/TCP                      22m
stable-kube-prometheus-sta-prometheus     LoadBalancer   10.100.99.75     a66a3304a3ec04a7c9fb19836338e430-1534821484.us-east-2.elb.amazonaws.com    9090:31486/TCP               22m
stable-kube-state-metrics                 ClusterIP      10.100.73.111    <none>                                                                     8080/TCP                     22m
stable-prometheus-node-exporter           ClusterIP      10.100.237.128   <none>                                                                     9100/TCP                     22m
jenkins@ip-172-31-4-244:~$
```

Access Grafana UI in the browser

Get the URL from the above screenshot and put in the browser



UserName: admin

Password: prom-operator

Create Dashboard in Grafana

In Grafana, we can create various kinds of dashboards as per our needs.

How to Create a Kubernetes Monitoring Dashboard?

For creating a dashboard to monitor the cluster:

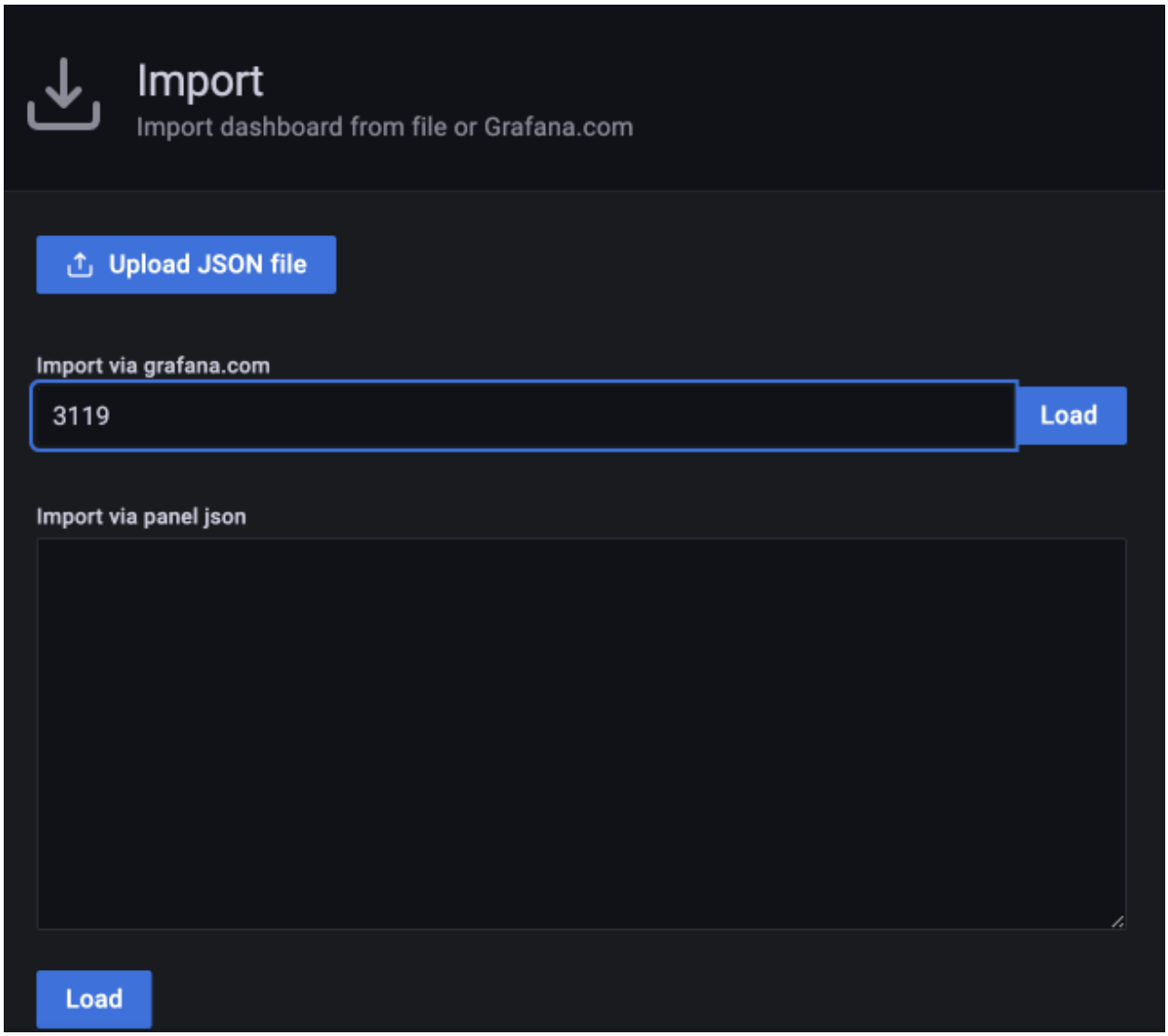
Click the '+' button on the left panel and select ‘Import’.
Enter `12740` dashboard id under Grafana.com Dashboard.
Click ‘Load’.
Select ‘Prometheus’ as the endpoint under prometheus data sources drop down.
Click ‘Import’.

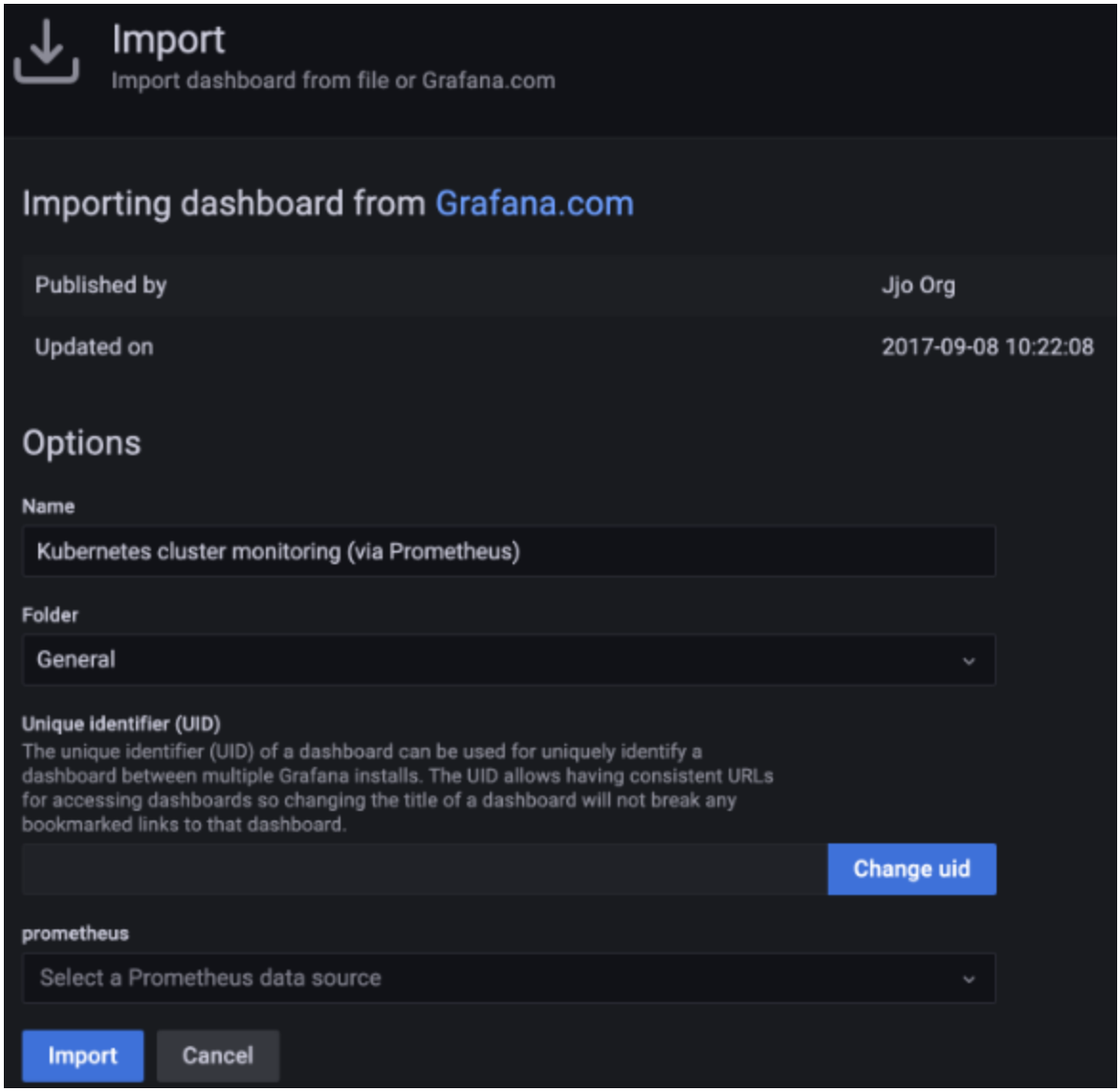
This will show monitoring dashboard for all cluster nodes

How to Create Kubernetes Cluster Monitoring Dashboard?

For creating a dashboard to monitor the cluster:

Click the '+' button on the left panel and select ‘Import’.
Enter 3119 dashboard id under Grafana.com Dashboard.
Click ‘Load’.
Select ‘Prometheus’ as the endpoint under prometheus data sources drop down.
Click ‘Import’.
This will show monitoring dashboard for all cluster nodes

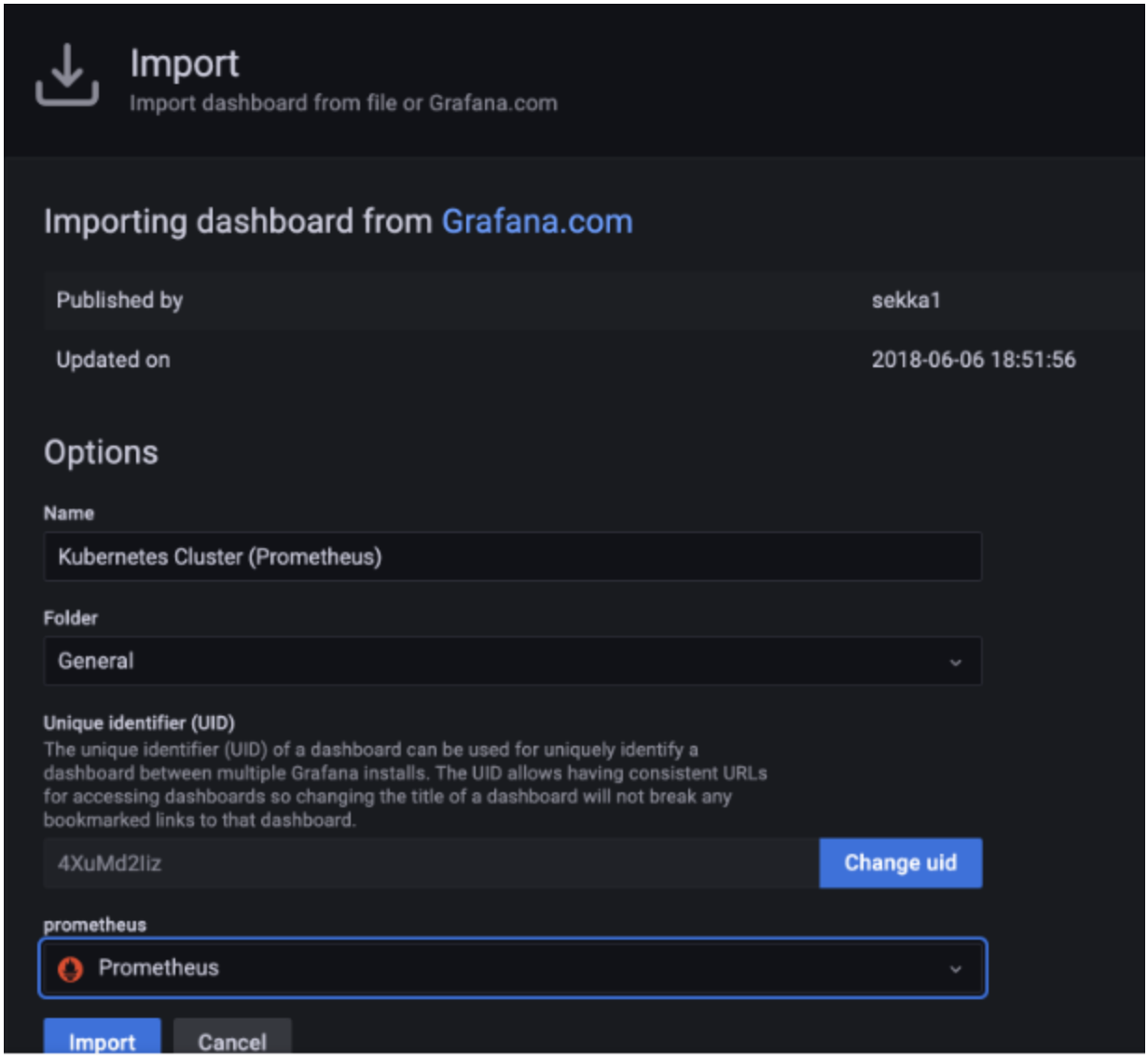



Import
Import dashboard from file or Grafana.com
Importing dashboard from Grafana.com
Published by
Jjo Org
Updated on
2017-09-08 10:22:08
Options
Name
Kubernetes cluster monitoring (via Prometheus)
Folder
General
Unique identifier (UID)
The unique identifier (UID) of a dashboard can be used for uniquely 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.
Change uid
prometheus
Select a Prometheus data source
Import
Cancel

Create POD Monitoring Dashboard

For creating a dashboard to monitor the cluster:

Click the '+' button on the left panel and select ‘Import’.
Enter `6417` dashboard id under Grafana.com Dashboard.
Click ‘Load’.
Select ‘Prometheus’ as the endpoint under prometheus data sources drop down.
Click ‘Import’.



This will show a monitoring dashboard for all cluster nodes.


General / Kubernetes Cluster (Prometheus)
Last 30 minutes
node
.*
namespace
.*
Cluster Health
Cluster Pod Usage
N/A
Cluster CPU Usage
N/A
Cluster Memory Usage
N/A
Cluster Disk Usage
N/A
Cluster Pod Capacity
pods
20
15
10
5
18:30
18:40
18:50
requested
Cluster CPU Capacity
cores
1
0.500
0
-0.50
-1
No data
18:30
18:40
18:50
Cluster Mem Capacity
1 B
0.500 B
0 B
-0.50 B
-1 B
No data
18:30
18:40
18:50
Cluster Disk Capacity
1 B
0.500 B
0 B
-0.50 B
-1 B
No data
18:30
18:40
Deployments
Deployment Replicas - Up To Date
Time
Metric
Value
2022-05-20 18:53:04
stable-kube-state-metrics
1
2022-05-20
stable-kube-prometheus-sta-
1
Deployment Replicas
5
Deployment Replicas - Updated
5
Deployment Replicas - Unavailal
0