Overview

This document will guide you how to setup the Prometheus (to collect the metrics from the AKS cluster) and Grafana (to visualize the metrics from the Prometheus on a Dashboard).

Prerequisites:

- K8S (AKS, EKS) cluster is setup and application is deployed on it.
- Clone this source code from GitHub
  https://github.com/sieunhantanbao/sd2411_azure_infrastructure
  - git clone https://github.com/sieunhantanbao/sd2411_azure_infrastructure.git
  - cd sd2411_azure_infrastructure/

1. Install Istio

Refer to this document to enable the Istio for the AKS cluster.

- Run this Azure CLI command to enable the Istio on the existing AKS cluster
  - `az aks mesh enable --resource-group ${RESOURCE_GROUP} --name ${CLUSTER}`
- Confirm the Istio is installed successfully
  - `kubectl get pods -n aks-istio-system`

```
@k8s-node-0:~$ kubectl get pods -n aks-istio-system
NAME                                    READY   STATUS    RESTARTS   AGE
istiod-asm-1-17-67d689bdf8-dm78k        1/1     Running   0          61m
istiod-asm-1-17-67d689bdf8-pf8vk        1/1     Running   0          61m
```

- Enable sidecar injection for a namespace
  - `kubectl label namespace qa istio.io/rev=asm-1-17`

Note: We can refer to this document to enable ingress gateway.

2. Install Prometheus and Grafana to AKS cluster

- Change directory to */tools/monitoring/*
- Run the below commands
  - Get AKS context: `az aks get-credentials --resource-group <your resource group> --name <your aks cluster name>`
  - Deploy Prometheus and Grafana: **`sh install-monitoring-tools.sh`**

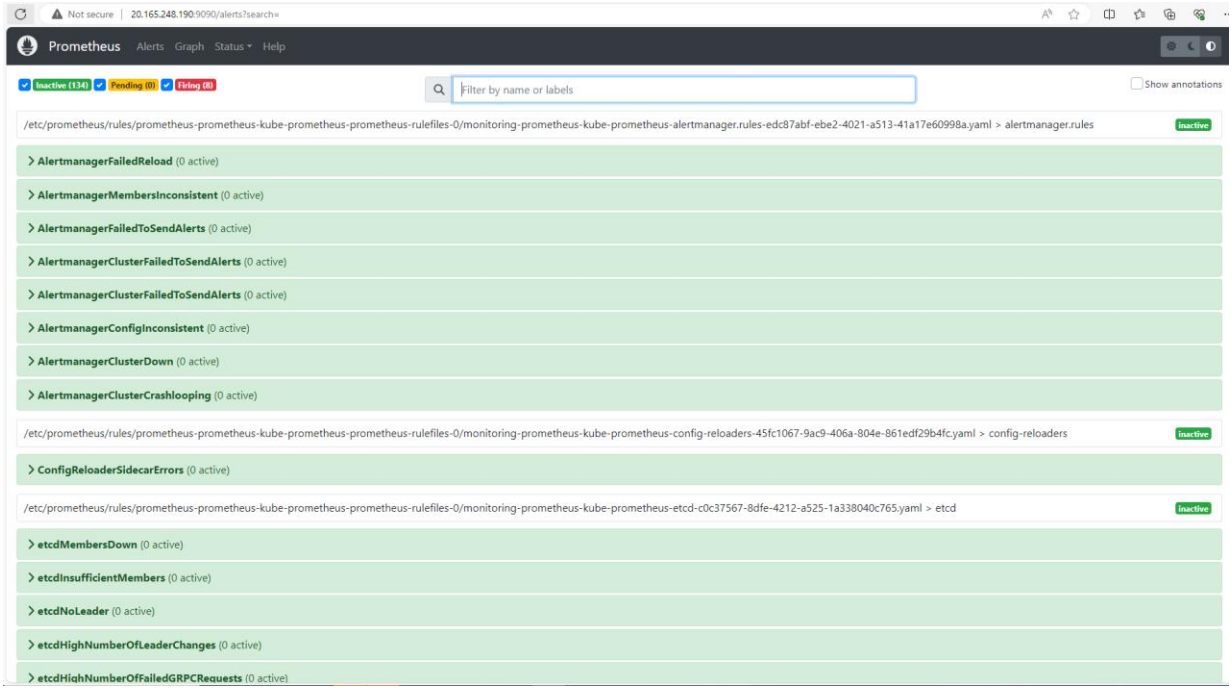
```
k8s-node-0:~/projects/sd2411_azure_infrastructure/tools/monitoring$ sh install-monitoring-tools.sh
"prometheus-community" already exists with the same configuration, skipping
Hang tight while we grab the latest from your chart repositories...
...Successfully got an update from the "mychart" chart repository
...Successfully got an update from the "sieunhantanbao" chart repository
...Successfully got an update from the "prometheus-community" chart repository
Update Complete. ⎈Happy Helming!⎈

NAME: prometheus
LAST DEPLOYED: Fri Oct 27 04:35:03 2023
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=prometheus"

Visit https://github.com/prometheus-operator/kube-prometheus for instructions on how to create & configure Alertmanager and Prometheus instances using the Operator.
NAME                                                READY   STATUS    RESTARTS   AGE
prometheus-kube-prometheus-operator-64f454f8bb-w4gqv   1/1     Running   0          17s
prometheus-kube-state-metrics-57b866d886-rdf2d      1/1     Running   0          17s
prometheus-prometheus-node-exporter-2pdr9           1/1     Running   0          17s
prometheus-prometheus-node-exporter-xwxfb           1/1     Running   0          17s
```

3. Setup Grafana Dashboard to use the metrics from Prometheus

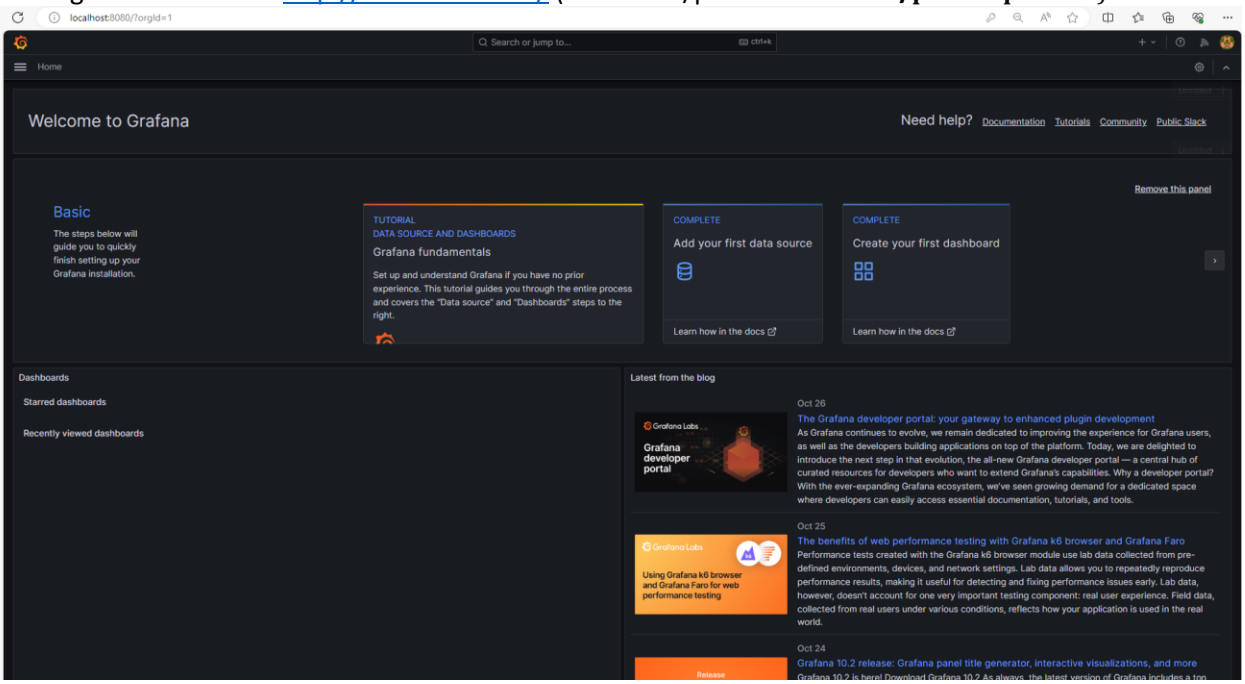
- Change the Service type of the Prometheus service to LoadBalancer to allow Grafana to access
  - `kubectl edit svc/prometheus-kube-prometheus-prometheus -n monitoring`

- Access to Prometheus through the external IP

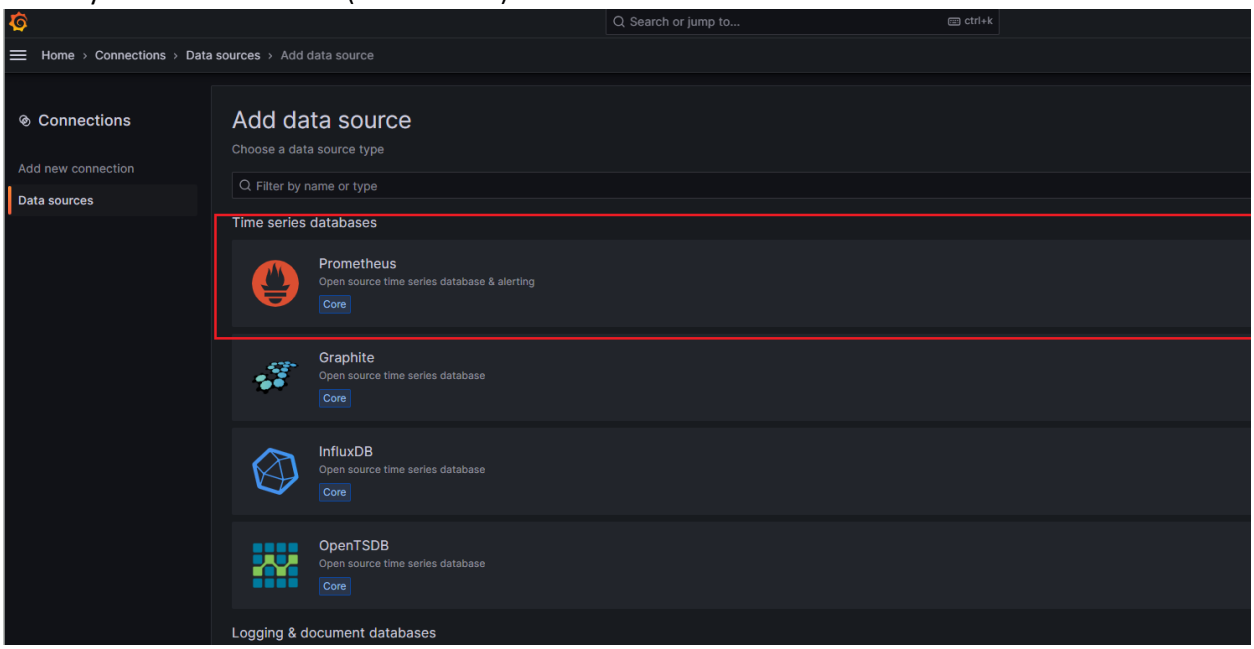
```
NAME                                      TYPE           CLUSTER-IP     EXTERNAL-IP      P
)                                  AGE
prometheus-kube-prometheus-prometheus     LoadBalancer   10.0.135.68    20.165.248.190   9
2586/TCP,8080:31955/TCP    175m
```

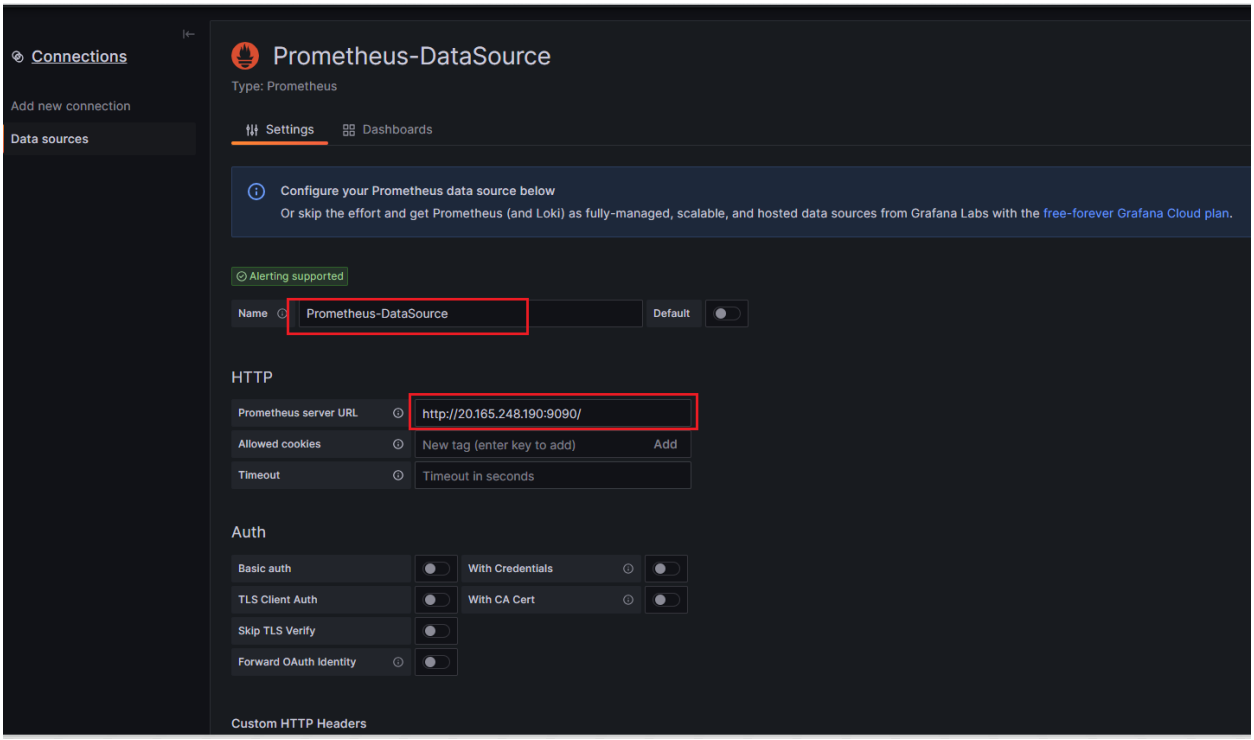


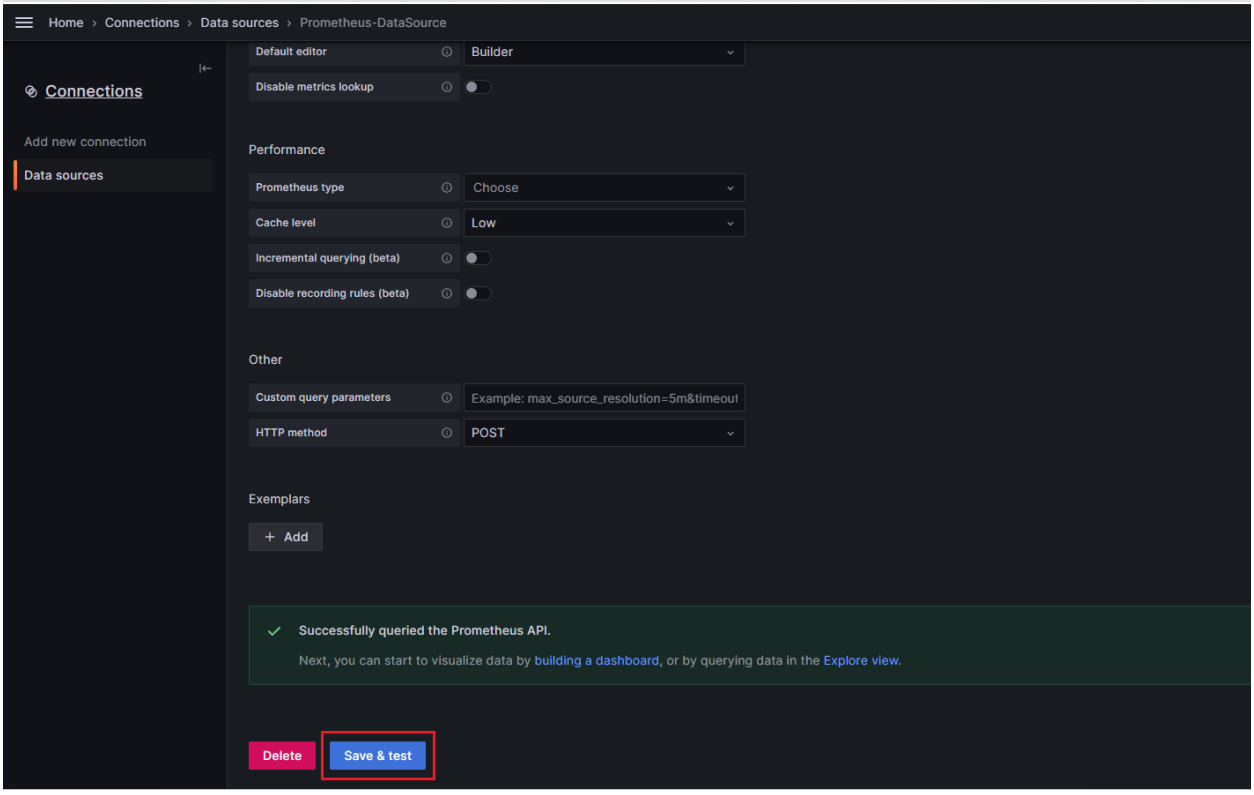
- Configure Grafana to get data (metrics) from Prometheus
    - Use kubectl port forward to access the Grafana on local: **`kubectl port-forward --namespace monitoring svc/prometheus-grafana 8080:80`**
- Login to Grafana at: http://localhost:8080/ (username/password: **admin/prom-operator**)



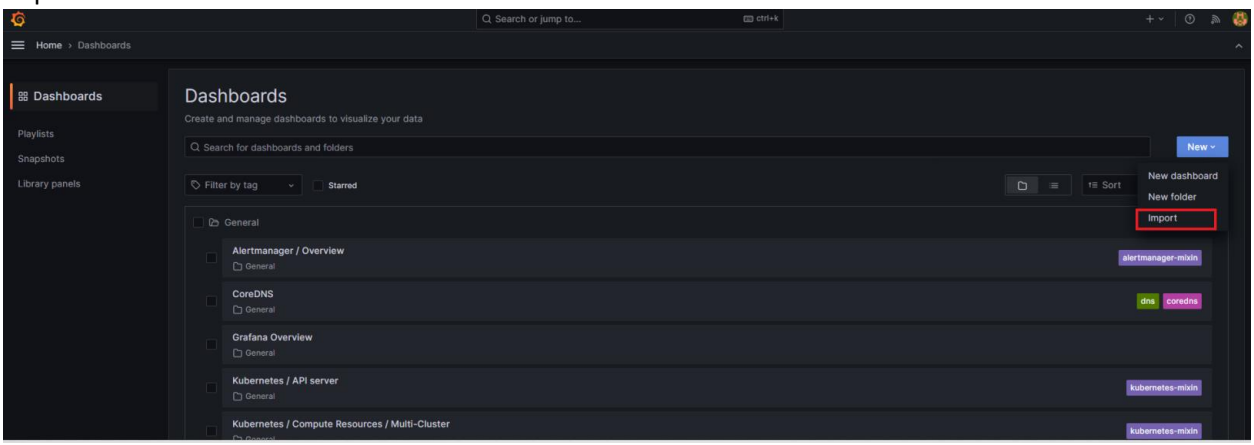
- Create your first data source (Prometheus)



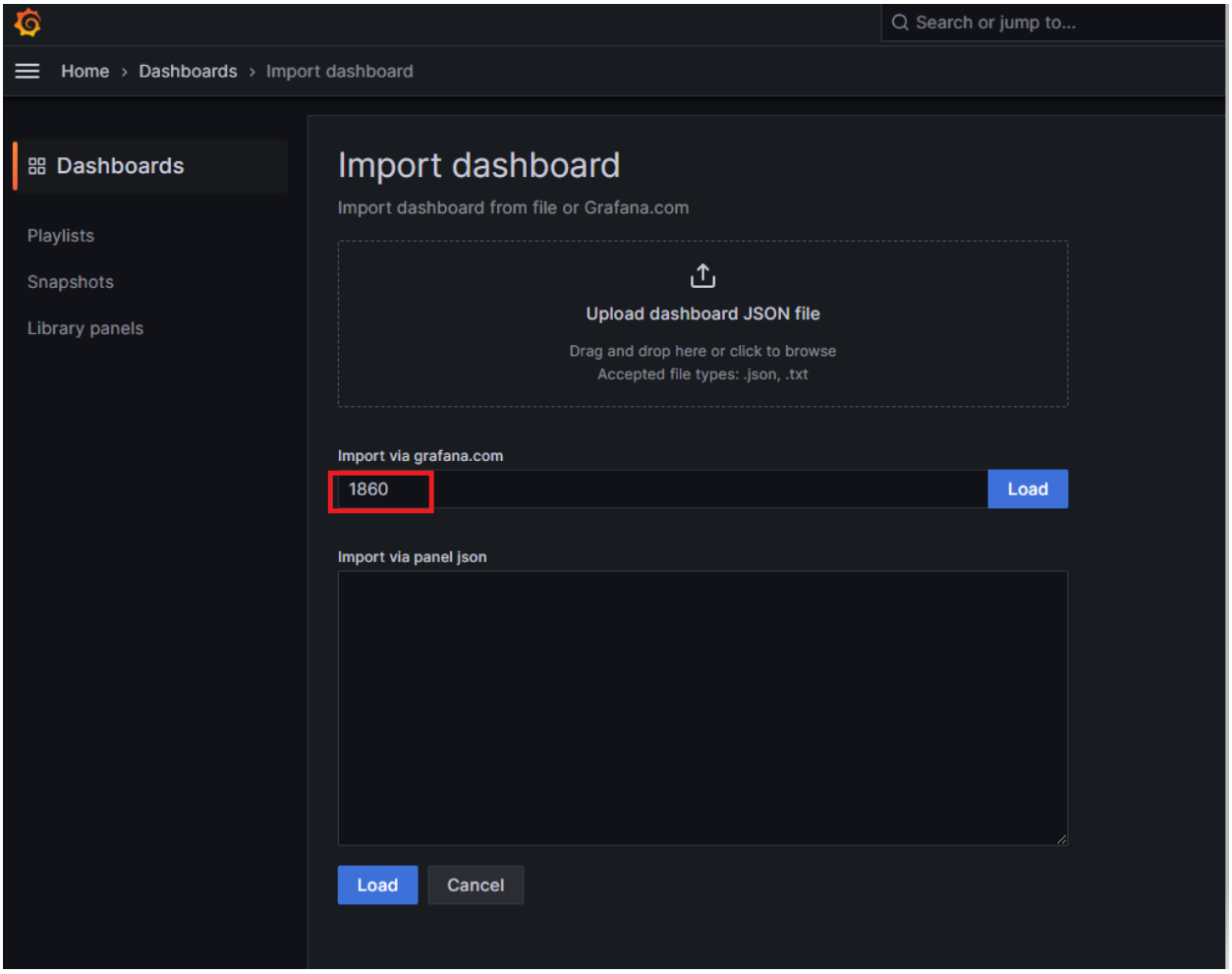

Connections
Add new connection
Data sources
Prometheus-DataSource
Type: Prometheus
Settings
Dashboards
Configure your Prometheus data source below
Or skip the effort and get Prometheus (and Loki) as fully-managed, scalable, and hosted data sources from Grafana Labs with the free-forever Grafana Cloud plan.
Alerting supported
Name
Prometheus-DataSource
Default
HTTP
Prometheus server URL
http://20.165.248.190:9090/
Allowed cookies
New tag (enter key to add)
Add
Timeout
Timeout in seconds
Auth
Basic auth
With Credentials
TLS Client Auth
With CA Cert
Skip TLS Verify
Forward OAuth Identity
Custom HTTP Headers

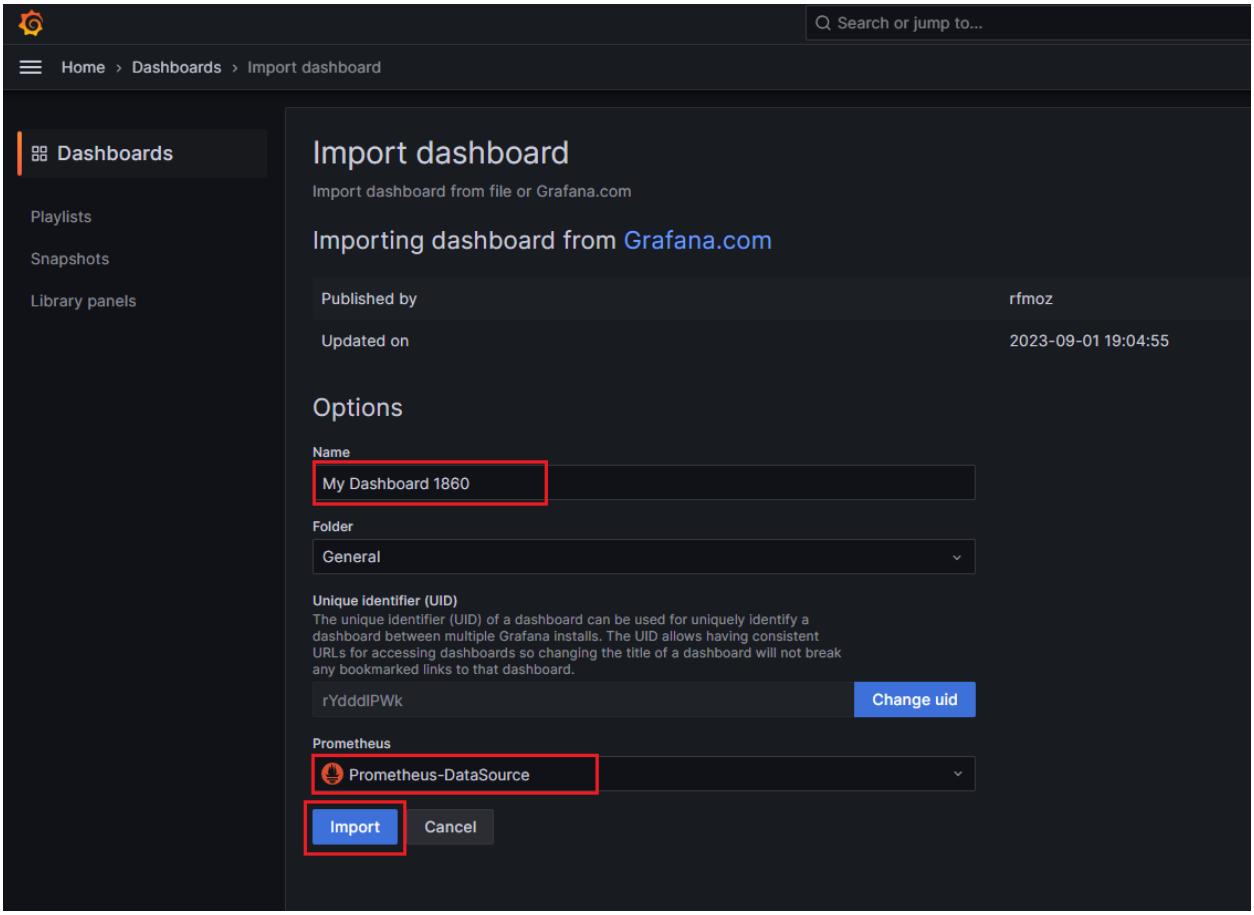

Home › Connections › Data sources › Prometheus-DataSource
Connections
Add new connection
Data sources
Default editor
Builder
Disable metrics lookup
Performance
Prometheus type
Choose
Cache level
Low
Incremental querying (beta)
Disable recording rules (beta)
Other
Custom query parameters
Example: max_source_resolution=5m&timeout
HTTP method
POST
Exemplars
+ Add
Successfully queried the Prometheus API.
Next, you can start to visualize data by building a dashboard, or by querying data in the Explore view.
Delete
Save & test

- Import new Dashboard



- Input **1860** to the “Import via grafana.com” (more detail can be found here https://grafana.com/grafana/dashboards/) and click “Load”





- View the Imported Dashboard