

컨테이너 인프라 환경 구축을 위한 쿠버네티스/도커

네이버클라우드 AlaaS 개발자 1기 김다예레

CONTENTS

01

깃허브 주소

02

헬름으로 그라파나
설치하기

03

프로메테우스를
데이터 소스로
구성하기

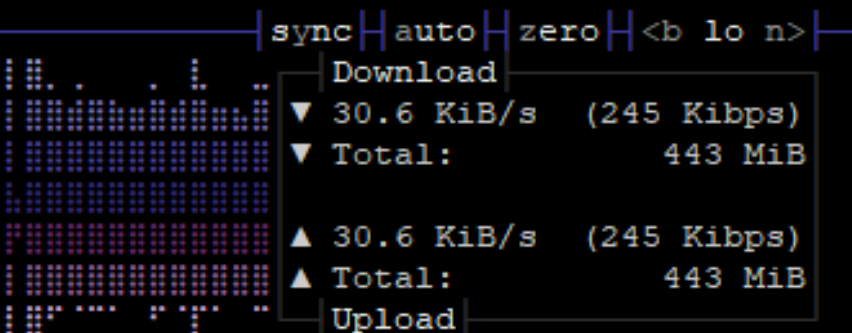
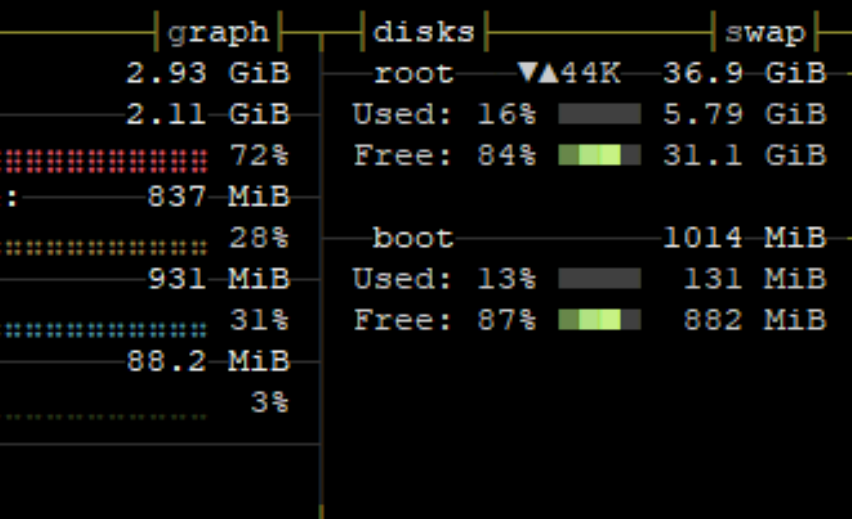
04

완성

01

모든 과정은 깃허브에 올려두었습니다.

<https://github.com/5000dosa/Docker-KubernetesGuide>



proc	filter	per-core	reverse	tree	<
Pid:	Program:	Tr:	User:	MemB	
13911	java	47	vagrant	2G	
3563	kube-apiserver	12	root	346M	
995	kubelet	21	root	75M	
2788	bpytop	3	root	21M	
4327	calico-node	14	root	28M	
999	dockerd	14	root	92M	
3358	etcd	12	root	73M	
1779	kube-controller-manage	9	root	77M	
1781	kube-scheduler	10	root	33M	
4782	coredns	11	root	24M	
5399	coredns	10	root	24M	
4246	containerd-shim	12	root	8M	
998	containerd	36	root	52M	
9	rcu_sched	1	root	0B	
4552	speaker	7	root	18M	
1	systemd	1	root	7M	
5674	containerd-shim	13	root	8M	
406	xfsaild/dm-0	1	root	0B	
3598	kworker/0:0	1	root	0B	
↑ select ↓ info □ terminate kill interrupt					



Namespace kube-system Pod All duration 2m node w2-k8s instance 192.168.1.102:9100

클러스터 요약

마스터 정상 가동률

100%

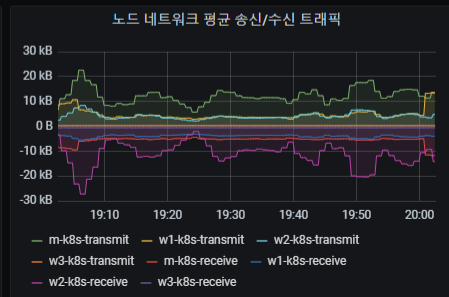
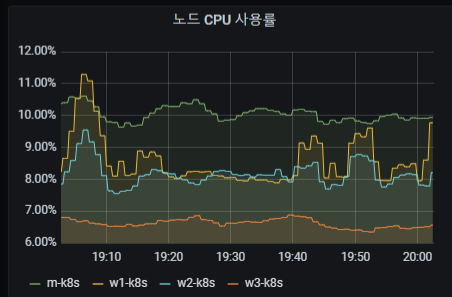
네임스페이스5

Pods29

PVC4

노드 현황						
Node name	Docker version	Kernel version	K8s version	OS version	Pod CIDR	role
w3-k8s	docker://18.9.9	3.10.0-1127.19.1.el7.x86_64	v1.18.4	CentOS Linux 7 (Core)	172.16.3.0/24	master
m-k8s	docker://18.9.9	3.10.0-1127.19.1.el7.x86_64	v1.18.4	CentOS Linux 7 (Core)	172.16.0.0/24	
w1-k8s	docker://18.9.9	3.10.0-1127.19.1.el7.x86_64	v1.18.4	CentOS Linux 7 (Core)	172.16.1.0/24	
w2-k8s	docker://18.9.9	3.10.0-1127.19.1.el7.x86_64	v1.18.4	CentOS Linux 7 (Core)	172.16.2.0/24	

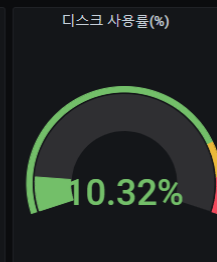
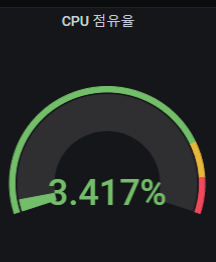
Cluster Metrics



노드 상태

m-k8s	Good
w1-k8s	Good
w2-k8s	Good
w3-k8s	Good

노드 m-k8s

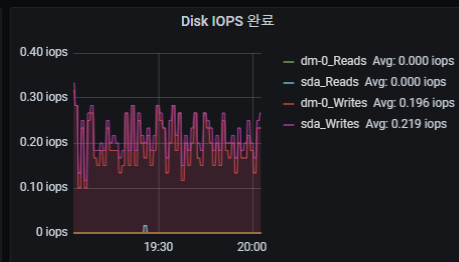
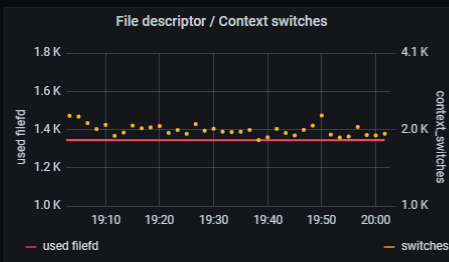


CPU 코어

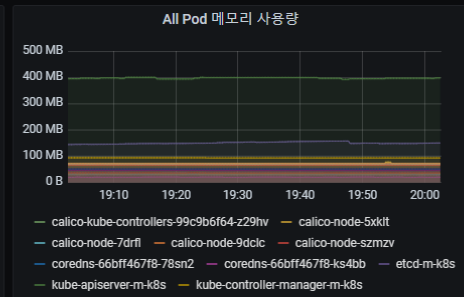
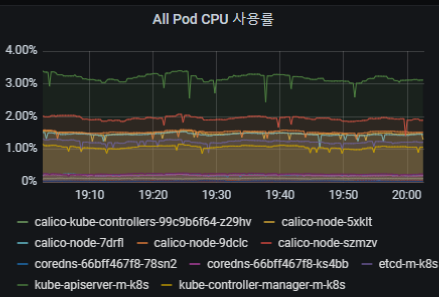
1

메모리

2.625 GB



kube-system Namespace Metrics



API 서버 응답 시간(5분/SLA 99%)

1.00 min

Pod 상태	
Failed	0
Pending	0
Running	15
Succeeded	0
Unknown	0

02

헬름으로 그라파나
설치하기

```
[root@m-k8s ~]# ~/_Book_k8sInfra/ch6/6.4.1/grafana-preconfig.sh
[Step 1/4] Task [Check helm status]
[Step 1/4] ok
[Step 2/4] Task [Check MetalLB status]
[Step 2/4] ok
[Step 3/4] Task [Create NFS directory for grafana]
/nfs_shared/grafana created
[Step 3/4] Successfully completed
[Step 4/4] Task [Create PV,PVC for grafana]
persistentvolume/grafana created
persistentvolumeclaim/grafana created
[Step 4/4] Successfully completed
```

```
[root@m-k8s ~]# ~/_Book_k8sInfra/ch6/6.4.1/grafana-install.sh
NAME: grafana
LAST DEPLOYED: Tue Mar 28 19:09:34 2023
NAMESPACE: default
STATUS: deployed
REVISION: 1
NOTES:
1. Get your 'admin' user password by running:

    kubectl get secret --namespace default grafana -o jsonpath="{.data.admin-password}" | base64 --decode ; echo

2. The Grafana server can be accessed via port 80 on the following DNS name from within your cluster:

    grafana.default.svc.cluster.local

    Get the Grafana URL to visit by running these commands in the same shell:
NOTE: It may take a few minutes for the LoadBalancer IP to be available.
    You can watch the status of by running 'kubectl get svc --namespace default -w grafana'
    export SERVICE_IP=$(kubectl get svc --namespace default grafana -o jsonpath='{.status.loadBalancer.ingress[0].ip
}')
    http://$SERVICE_IP:80

3. Login with the password from step 1 and the username: admin
```



```
$ grafana-install.sh ✕
```

```
_Book_k8sInfra(코드) > ch6 > 6.4.1 > $ grafana-install.sh
```

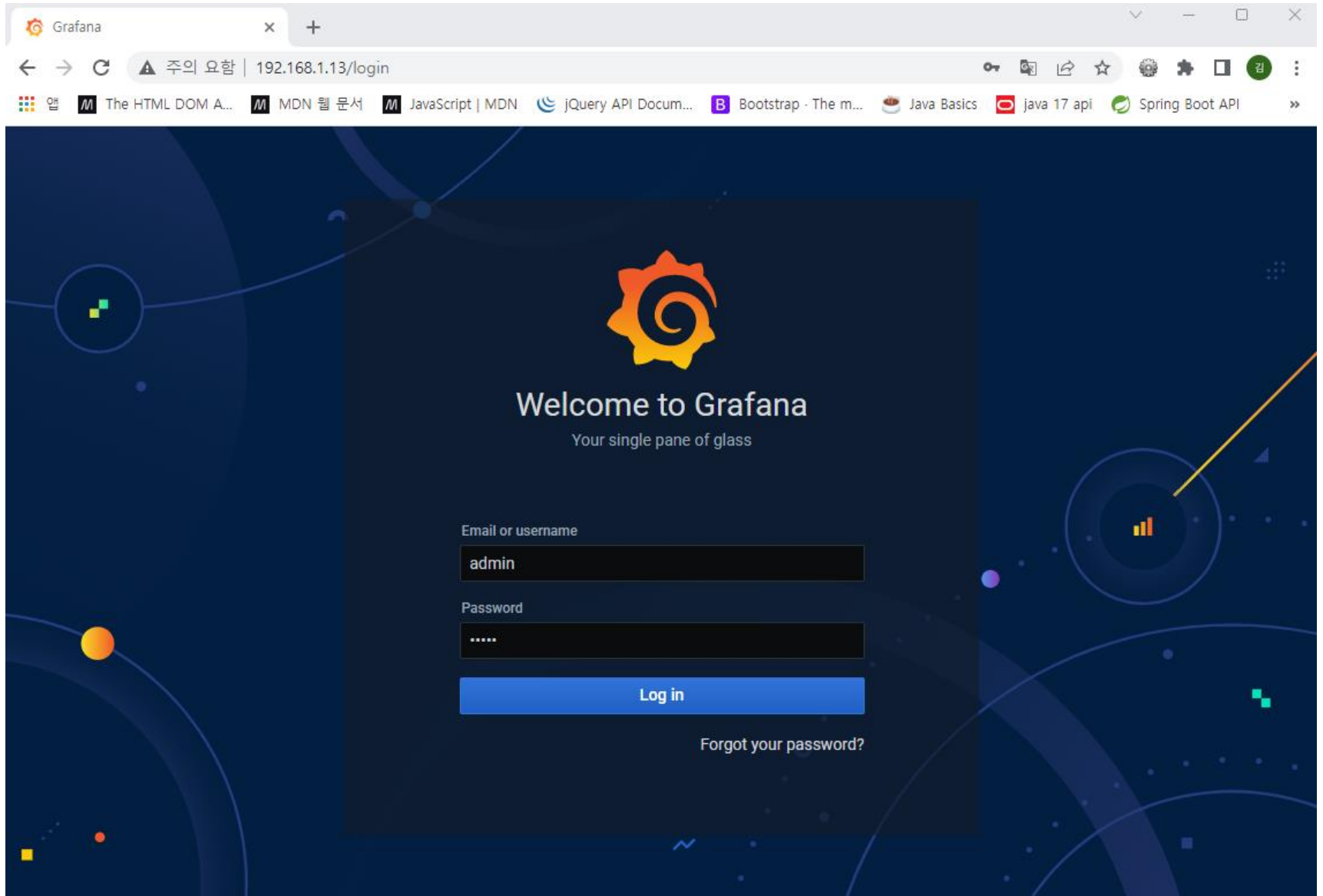
```
1  #!/usr/bin/env bash
2  helm install grafana edu/grafana \
3  --set persistence.enabled=true \
4  --set persistence.existingClaim=grafana \
5  --set service.type=LoadBalancer \
6  --set securityContext.runAsUser=1000 \
7  --set securityContext.runAsGroup=1000 \
8  --set adminPassword="admin"
```


```
[root@m-k8s ~]# kubectl get deployment grafana
```








NAME	READY	UP-TO-DATE	AVAILABLE	AGE
grafana	1/1	1	1	3m13s

```
[root@m-k8s ~]# kubectl get service grafana
```

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
grafana	LoadBalancer	10.109.237.199	192.168.1.13	80:30291/TCP	4m15s



 Home



Welcome to Grafana

Need help? [Documentation](#) [Tutorials](#) [Community](#) [Public Slack](#)

■ Basic


The steps below will guide you to quickly finish setting up your Grafana installation.

TUTORIAL

DATA SOURCE AND DASHBOARDS


Grafana fundamentals


Set up and understand Grafana if you have no prior experience. This tutorial guides you through the entire process and covers the "Data source" and "Dashboards" steps to the right.



DATA SOURCES


Add your first data source




Learn how in the docs 

DASHBOARDS

Create your first dashboard



Learn how in the docs 

[Remove this panel](#)

Dashboards

Starred dashboards

Recently viewed dashboards

Latest from the blog

A year in Mimir: Massive scale, new metrics formats, increased adoption3월 27

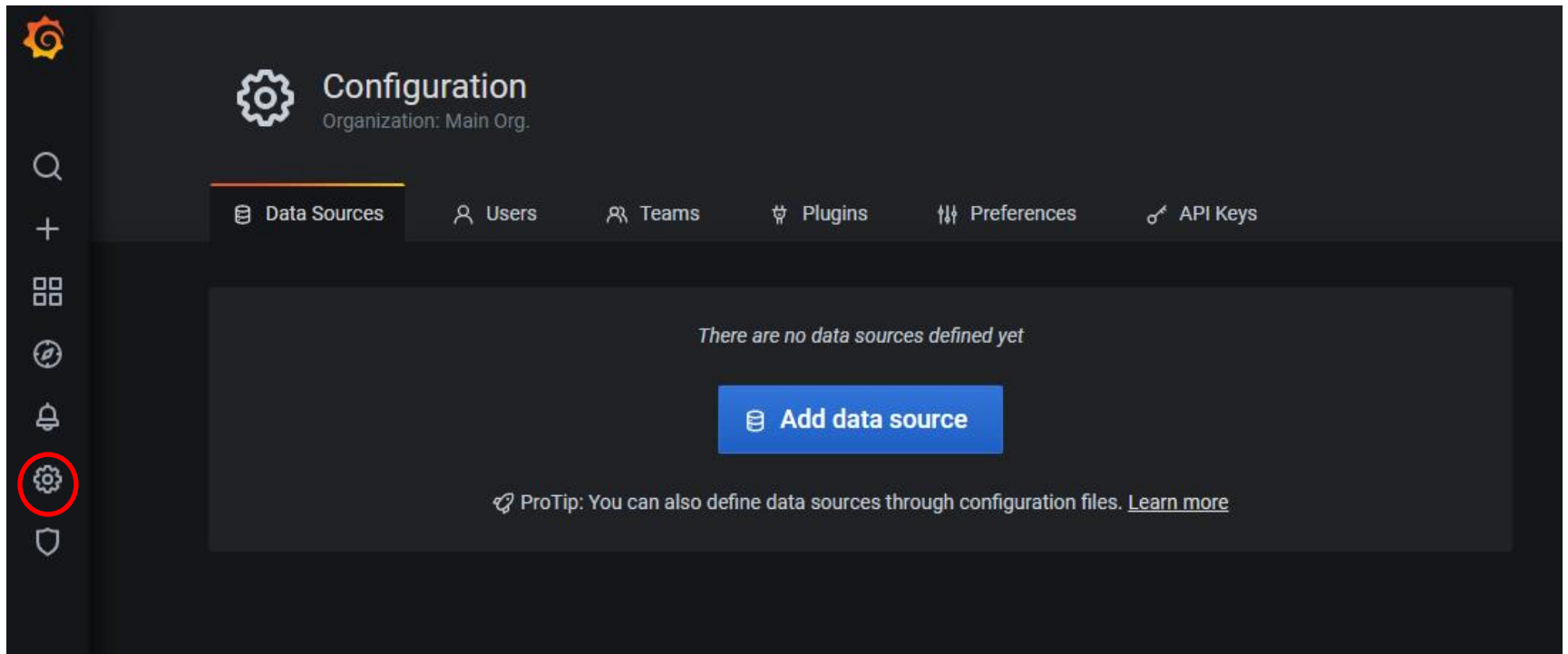
When we introduced Grafana Mimir into the open source ecosystem, we weren't shy about our ambitions. Once we got past answering some of the easier questions (For the record, the name Mimir comes from Norse mythology, and it's pronounced /mɪˈmɪr/), we quickly got to work making good on our promise to deliver the most scalable, most performant open source time series database (TSDB) in the world. Since launching Mimir a year ago, we have focused our efforts on enhancing Mimir's core capabilities to provide significant scale — 1 billion active series and beyond — with easy deployment, multi-tenancy, durable storage, high availability, and super fast query performance.

Meet the minds behind Grafana Pyroscope: Christian, Cyril, Dmitry, and Ryan3월 23

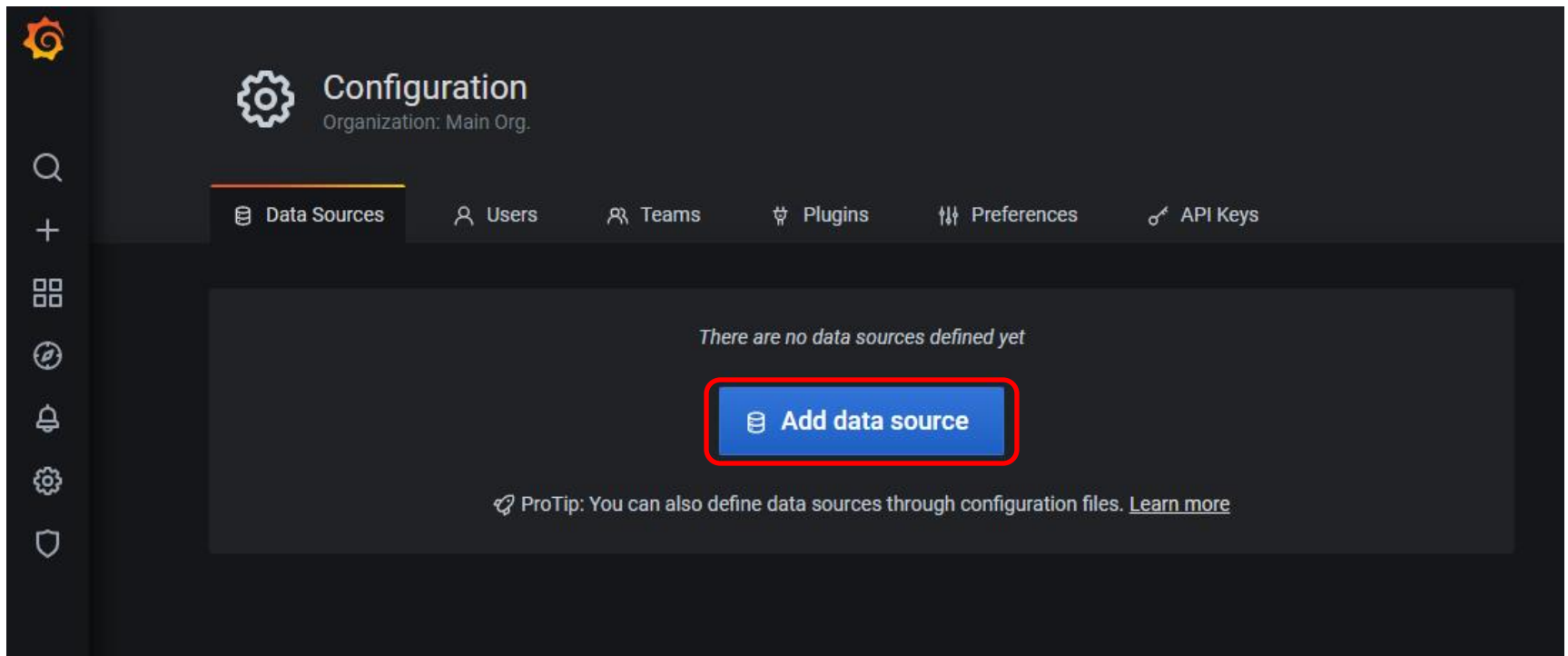
What do you get when you combine the wit, wisdom, and weird humor of four talented tech minds? As it turns out, a surprisingly lively Q&A! As Grafana Pyroscope emerges from the union of Grafana Phlare and Pyroscope, it's time to really get to know the people behind these continuous profiling projects. That's why we brought together the Pyroscope founders, Dmitry Filimonov and Ryan Perry, and Phlare technical leads, Cyril Toven and Christian Simon, for this light-hearted conversation.

Grafana security release: New versions with security fixes for CVE-2023-14103월 22

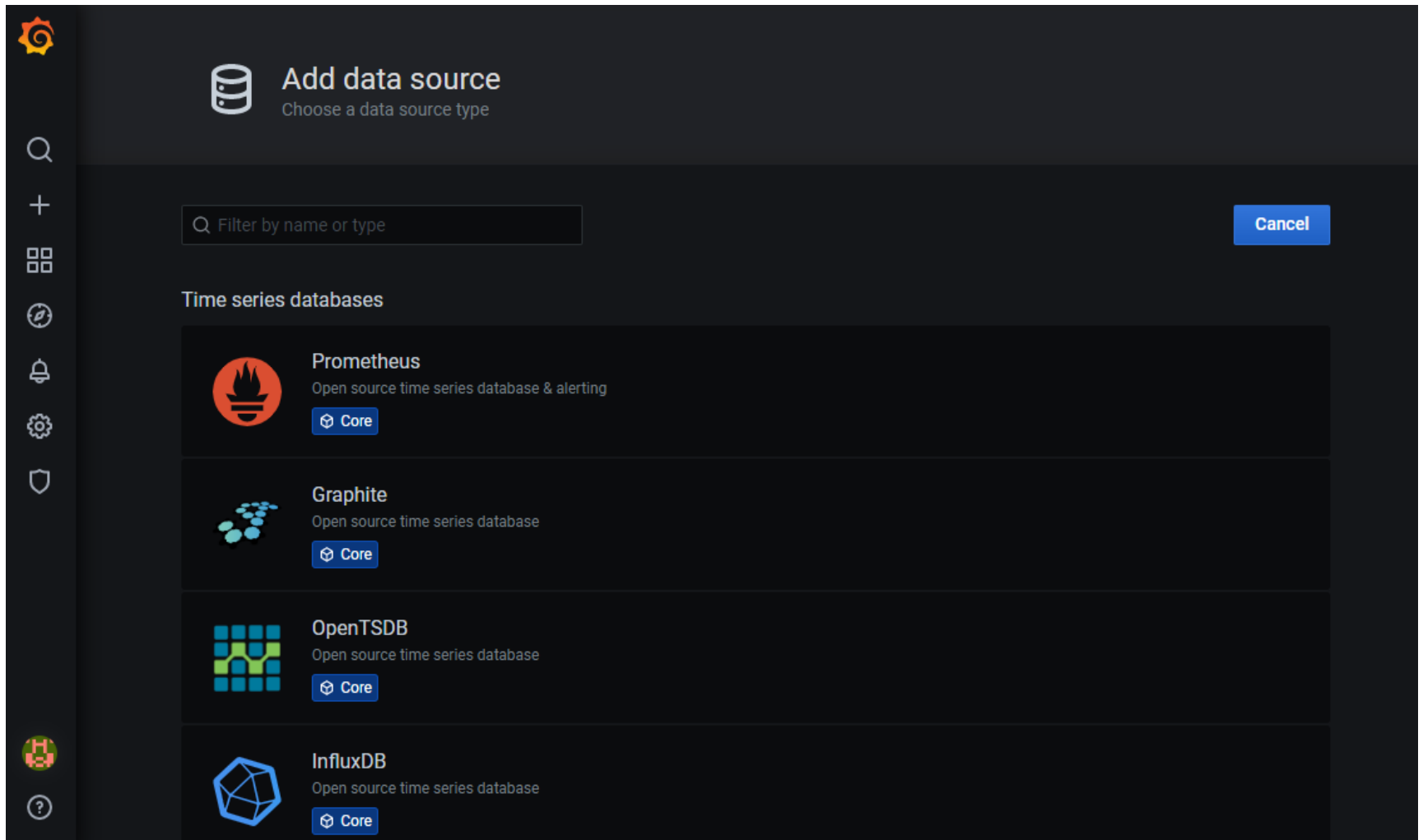
Today we are releasing Grafana 9.4.7, which includes updates such as enhanced navigation and custom visualization panels. In addition



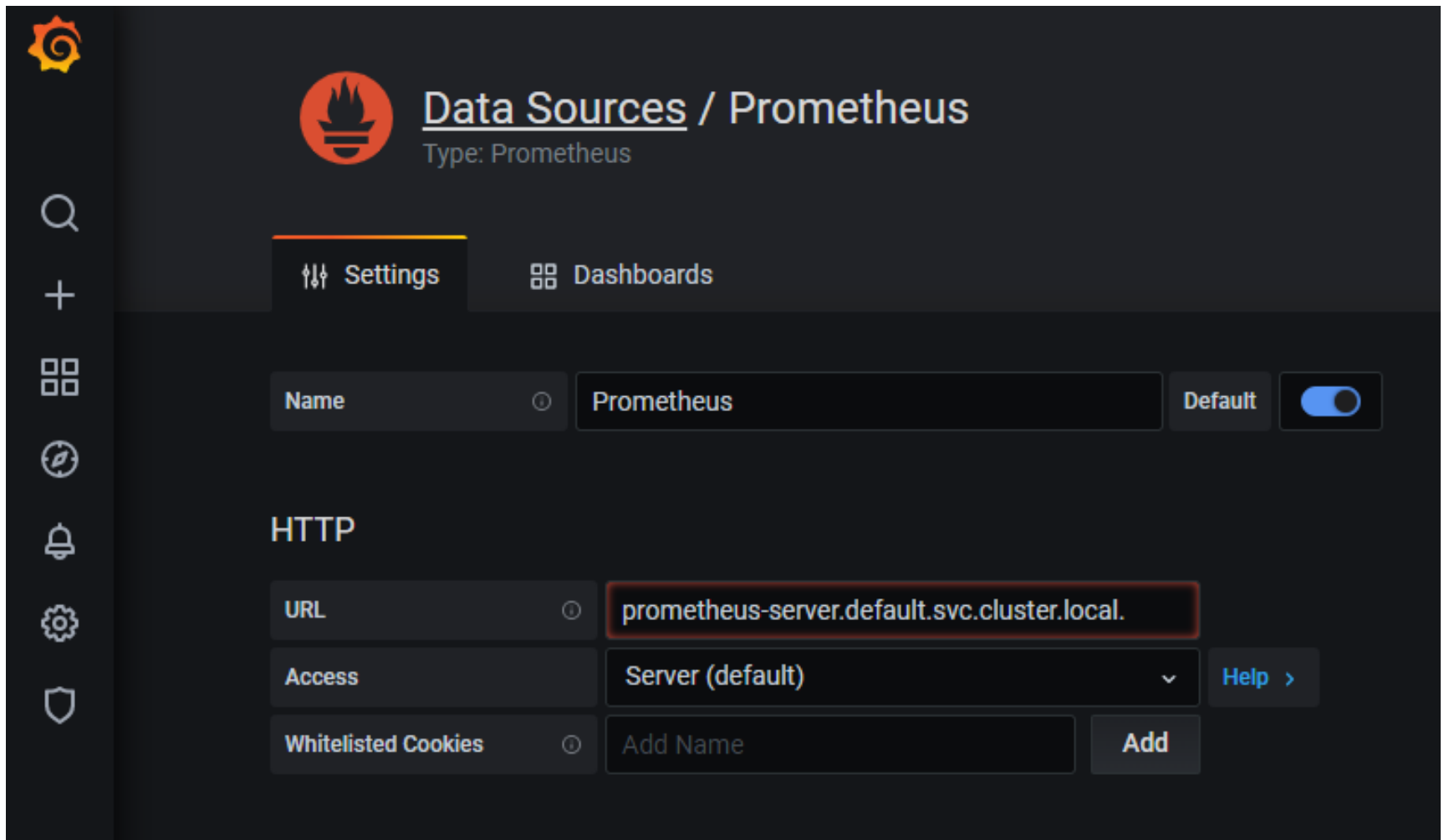
왼쪽 메뉴에서 톱니바퀴 모양의 아이콘을 누른 후 Data Sources를 선택합니다.





다음 화면이 보이면 Add data source 버튼을 누릅니다.



화면에 표시되는 다양한 소스 중에서 프로메테우스 아이템 위에 마우스를 올리고 Select 버튼을 눌러 프로메테우스를 데이터 소스로 선택합니다.



  **Data Sources / Prometheus**
Type: Prometheus

Settings **Dashboards**

Name ⓘ Prometheus **Default** ☒

HTTP

URL ⓘ prometheus-server.default.svc.cluster.local.

Access Server (default) **Help >**

Whitelisted Cookies ⓘ Add Name **Add**

```
[root@m-k8s ~]# kubectl run net --image=sysnet4admin/net-tools --restart=Never --rm -it -- nslookup 192.168.1.12
12.1.168.192.in-addr.arpa      name = prometheus-server.default.svc.cluster.local.
pod "net" deleted
```

슈퍼푸티창에서 URL 복사붙여넣기 합니다.

Misc

Disable metrics lookup



Custom query parameters



Example: max_source_resolution=5m&timeout=10

✓ Data source is working

Save & Test

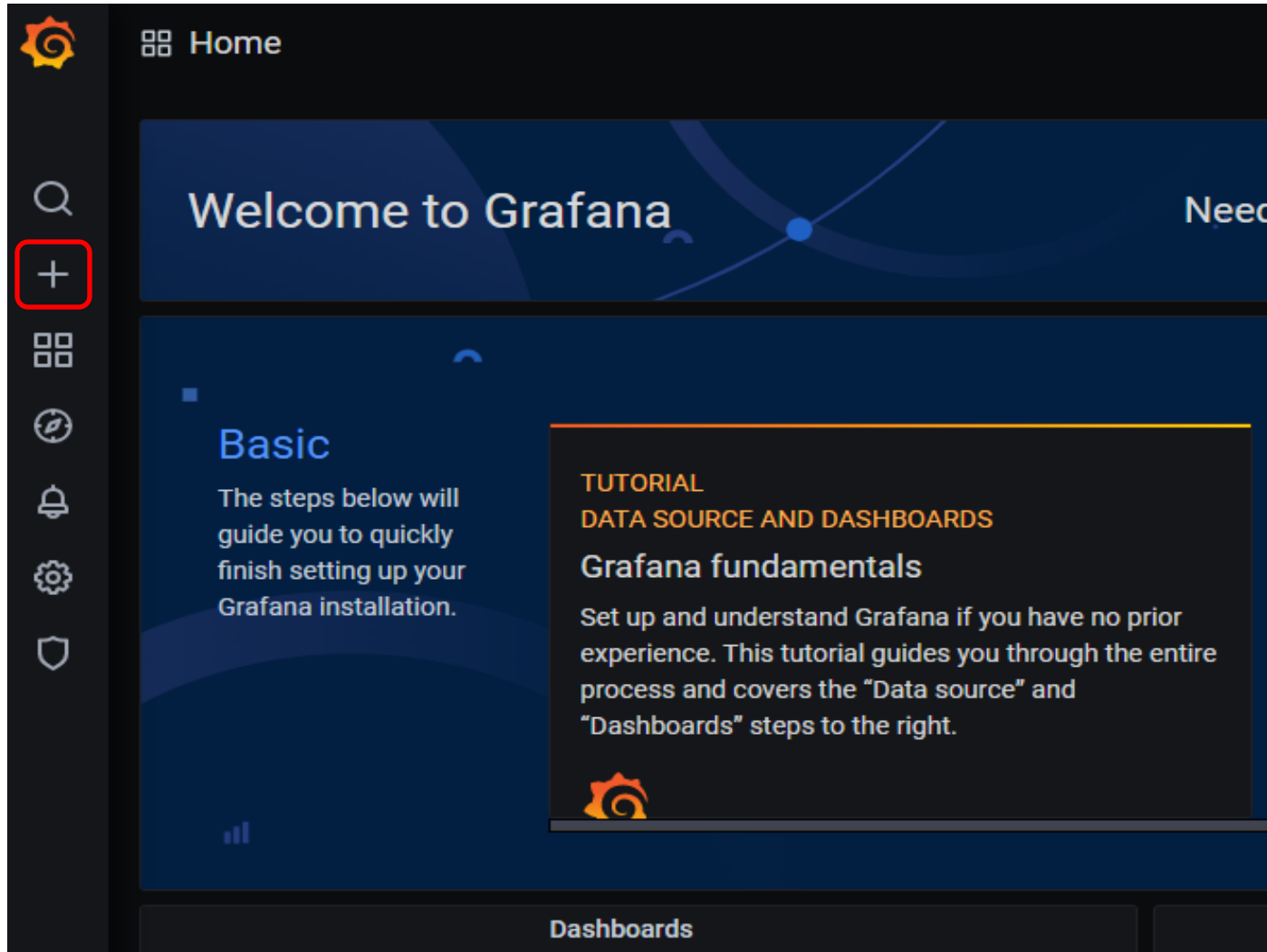
Delete

Back

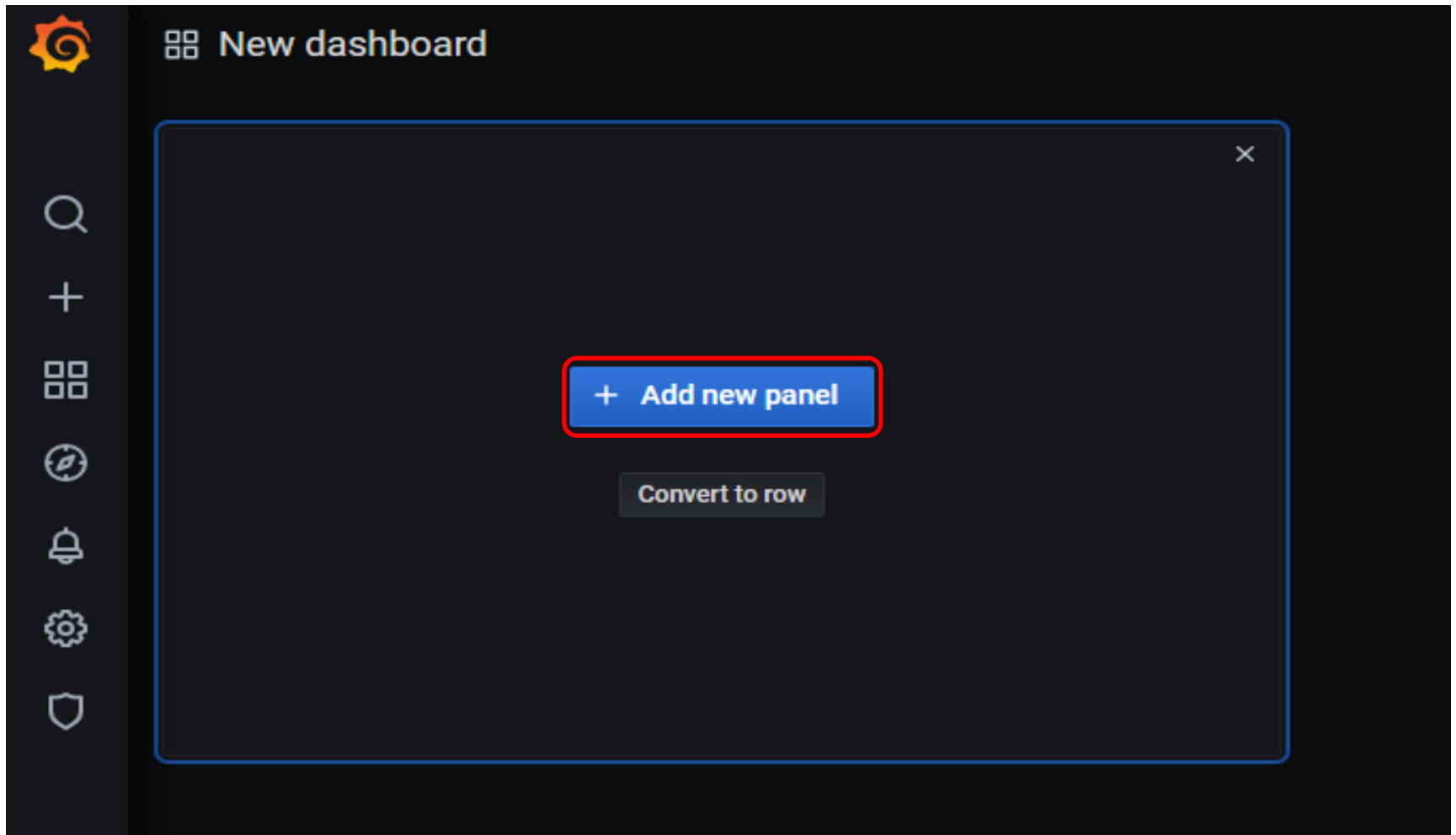
입력하면 Save & Test 버튼을 누르면 설정을 테스트하고 저장합니다.

03

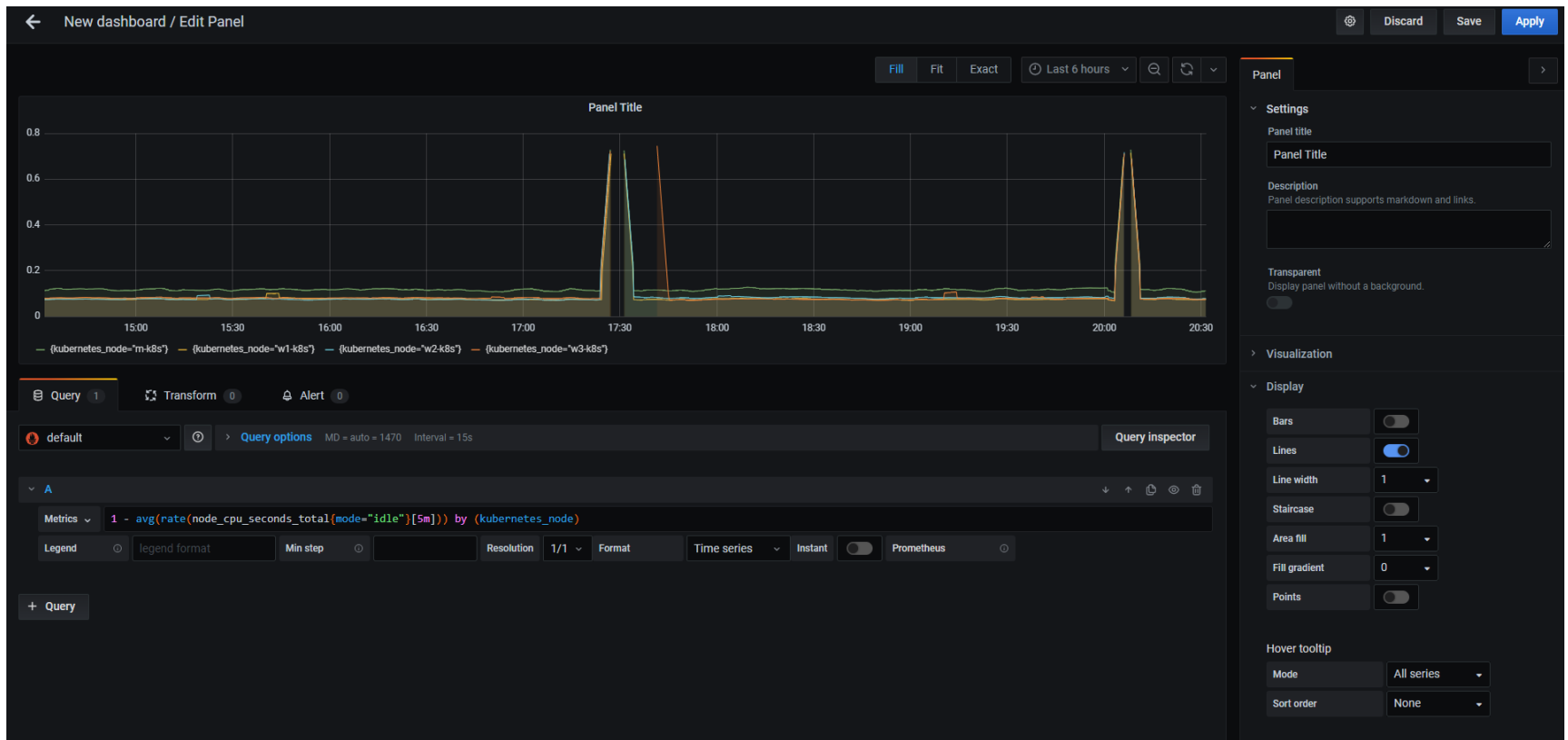
프로메테우스를
데이터 소스로
구성하기



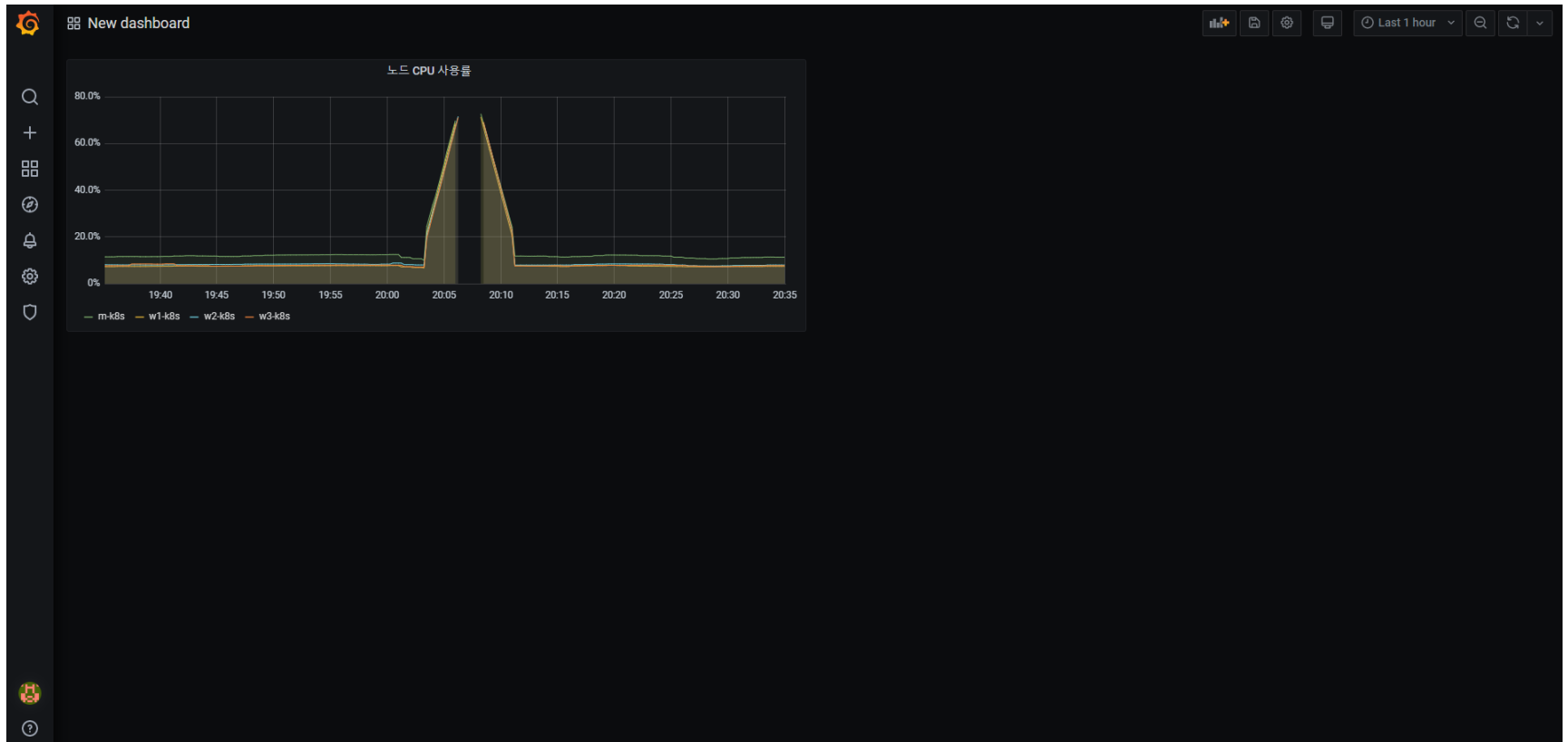
왼쪽 메뉴에서 + > DashBoard를 선택합니다.



대시보드를 생성하는 화면에서 + Add new panel 버튼을 누릅니다.



프로메테우스로부터 메트릭 값을 읽어 들이기



생성된 노드 CPU 사용률 패널을 확인합니다.



노드와 파드 데이터를 하나씩 불러와서 시각화 합니다.

04 완성



**THANK
YOU**