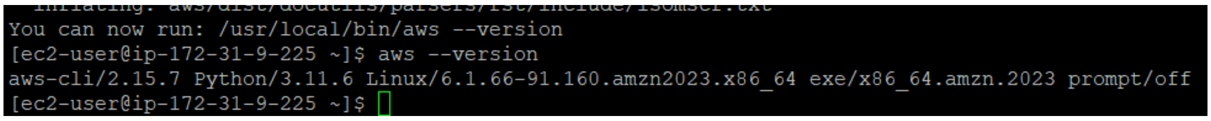
Install AWS CLI and Configure

curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"

sudo apt install unzip

unzip awscliv2.zip

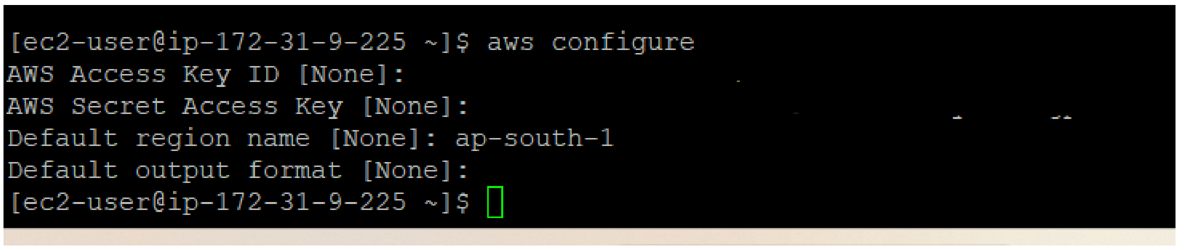
sudo ./aws/install



```
You can now run: /usr/local/bin/aws --version
[ec2-user@ip-172-31-9-225 ~]$ aws --version
aws-cli/2.15.7 Python/3.11.6 Linux/6.1.66-91.160.amzn2023.x86_64 exe/x86_64.amzn.2023 prompt/off
[ec2-user@ip-172-31-9-225 ~]$
```

Okay now after installing the AWS CLI, let's configure the ***AWS CLI*** so that it can authenticate and communicate with the AWS environment.

aws configure



```
[ec2-user@ip-172-31-9-225 ~]$ aws configure
AWS Access Key ID [None]:
AWS Secret Access Key [None]:
Default region name [None]: ap-south-1
Default output format [None]:
[ec2-user@ip-172-31-9-225 ~]$
```

Install and Setup Kubectl

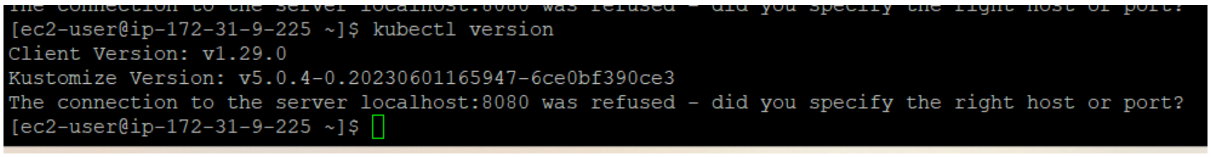
Moving forward now we need to set up the **kubectl** also onto the EC2 instance.

curl -LO "https://storage.googleapis.com/kubernetes-release/release/$(curl -s https://storage.googleapis.com/kubernetes-release/release/stable.txt)/bin/linux/amd64/kubectl"

chmod +x ./kubectl

sudo mv ./kubectl /usr/local/bin

kubectl version



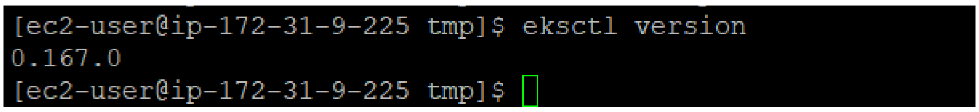
```
[ec2-user@ip-172-31-9-225 ~]$ kubectl version
Client Version: v1.29.0
Kustomize Version: v5.0.4-0.20230601165947-6ce0bf390ce3
The connection to the server localhost:8080 was refused - did you specify the right host or port?
[ec2-user@ip-172-31-9-225 ~]$
```

Install and Setup eksctl

curl --silent --location "https://github.com/weaveworks/eksctl/releases/latest/download/eksctl_$(uname -s)_amd64.tar.gz" | tar xz -C /tmp

sudo mv /tmp/eksctl /usr/local/bin

eksctl version



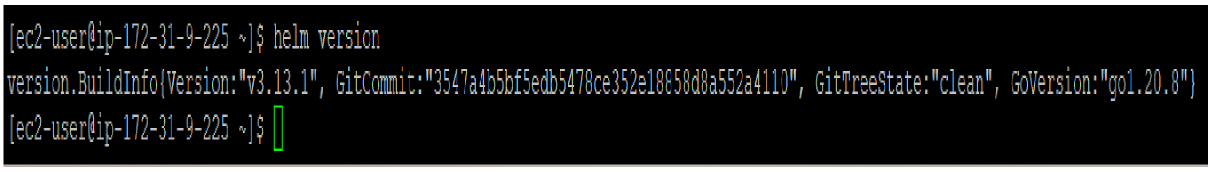
```
[ec2-user@ip-172-31-9-225 tmp]$ eksctl version
0.167.0
[ec2-user@ip-172-31-9-225 tmp]$
```

Install Helm chart

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

$ chmod 700 get_helm.sh

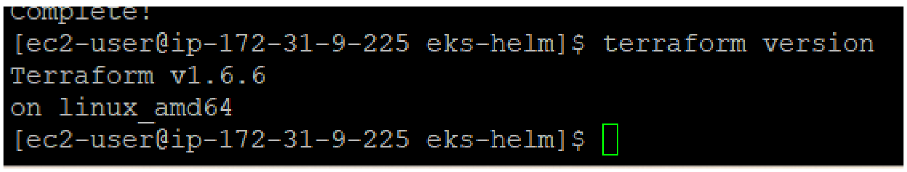
$ ./get_helm.sh



```
[ec2-user@ip-172-31-9-225 ~]$ helm version
version.BuildInfo{Version:"v3.13.1", GitCommit:"3547a4b5bf5edb5478ce352e18858d8a552a4110", GitTreeState:"clean", GoVersion:"go1.20.8"}
[ec2-user@ip-172-31-9-225 ~]$
```

This way we install all AWS CLI, kubectl, eksctl and Helm.

```
Follow below steps to install terraform on AmazonLinux.

sudo yum install -y yum-utils shadow-utils
sudo yum-config-manager --add-repo
https://rpm.releases.hashicorp.com/AmazonLinux/hashicorp.repo
sudo yum -y install terraform
```



```
Complete!
[ec2-user@ip-172-31-9-225 eks-helm]$ terraform version
Terraform v1.6.6
on linux_amd64
[ec2-user@ip-172-31-9-225 eks-helm]$
```

Creating an Amazon EKS cluster using terraform

Code available in https://github.com/ksnithya/blue-green.git

git clone https://github.com/ksnithya/blue-green.git

cd blue-green

terraform init

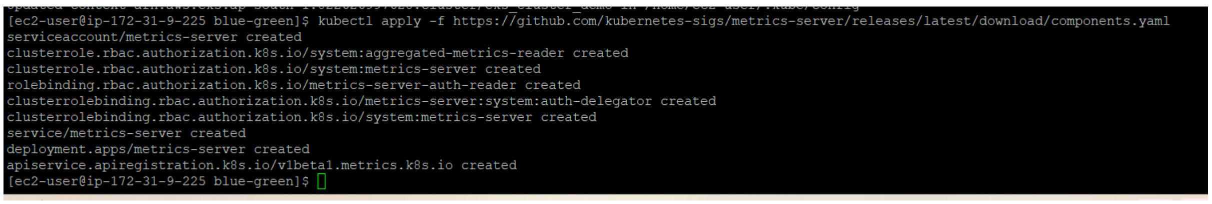
terraform plan

terraform apply

aws eks --region ap-south-1 update-kubeconfig --name eks_cluster_demo

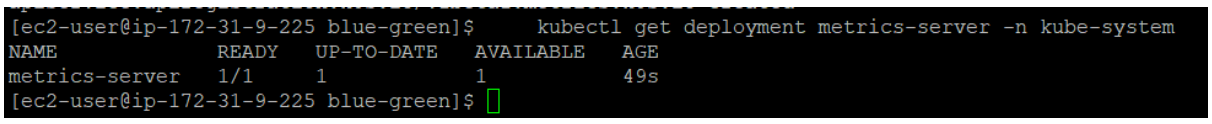
Installing the Kubernetes Metrics Server

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



```
[ec2-user@ip-172-31-9-225 blue-green]$ kubectl apply -f https://github.com/kubernetes-sigs/metrics-server/releases/latest/download/components.yaml
serviceaccount/metrics-server created
clusterrole.rbac.authorization.k8s.io/system:aggregated-metrics-reader created
clusterrole.rbac.authorization.k8s.io/system:metrics-server created
rolebinding.rbac.authorization.k8s.io/metrics-server-auth-reader created
clusterrolebinding.rbac.authorization.k8s.io/metrics-server:system:auth-delegator created
clusterrolebinding.rbac.authorization.k8s.io/system:metrics-server created
service/metrics-server created
deployment.apps/metrics-server created
apiservice.apiregistration.k8s.io/v1beta1.metrics.k8s.io created
[ec2-user@ip-172-31-9-225 blue-green]$
```

kubectl get deployment metrics-server -n kube-system



```
[ec2-user@ip-172-31-9-225 blue-green]$     kubectl get deployment metrics-server -n kube-system
NAME             READY   UP-TO-DATE   AVAILABLE   AGE
metrics-server   1/1     1            1           49s
[ec2-user@ip-172-31-9-225 blue-green]$
```

Install Prometheus

Now we install the Prometheus using the helm chart.

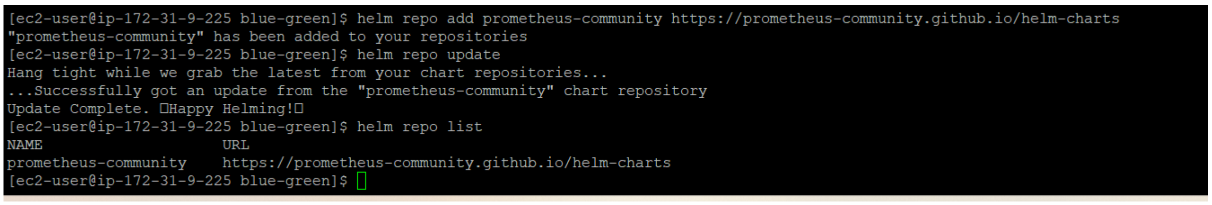
Add Prometheus helm chart repository

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

Update the helm chart repository

helm repo update

helm repo list



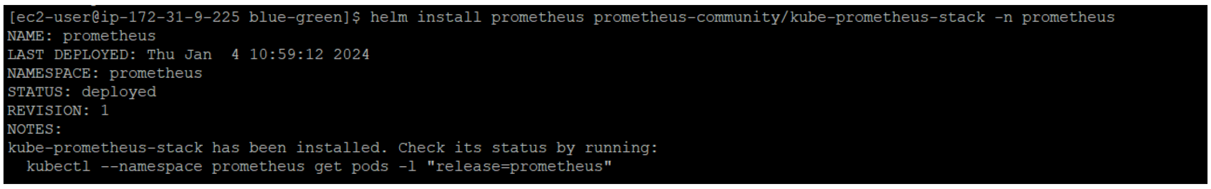
```
[ec2-user@ip-172-31-9-225 blue-green]$ helm repo add prometheus-community https://prometheus-community.github.io/helm-charts
"prometheus-community" has been added to your repositories
[ec2-user@ip-172-31-9-225 blue-green]$ helm repo update
Hang tight while we grab the latest from your chart repositories...
...Successfully got an update from the "prometheus-community" chart repository
Update Complete. □Happy Helming!□
[ec2-user@ip-172-31-9-225 blue-green]$ helm repo list
NAME                    URL
prometheus-community    https://prometheus-community.github.io/helm-charts
[ec2-user@ip-172-31-9-225 blue-green]$
```

Create `prometheus` namespace

kubectl create namespace Prometheus

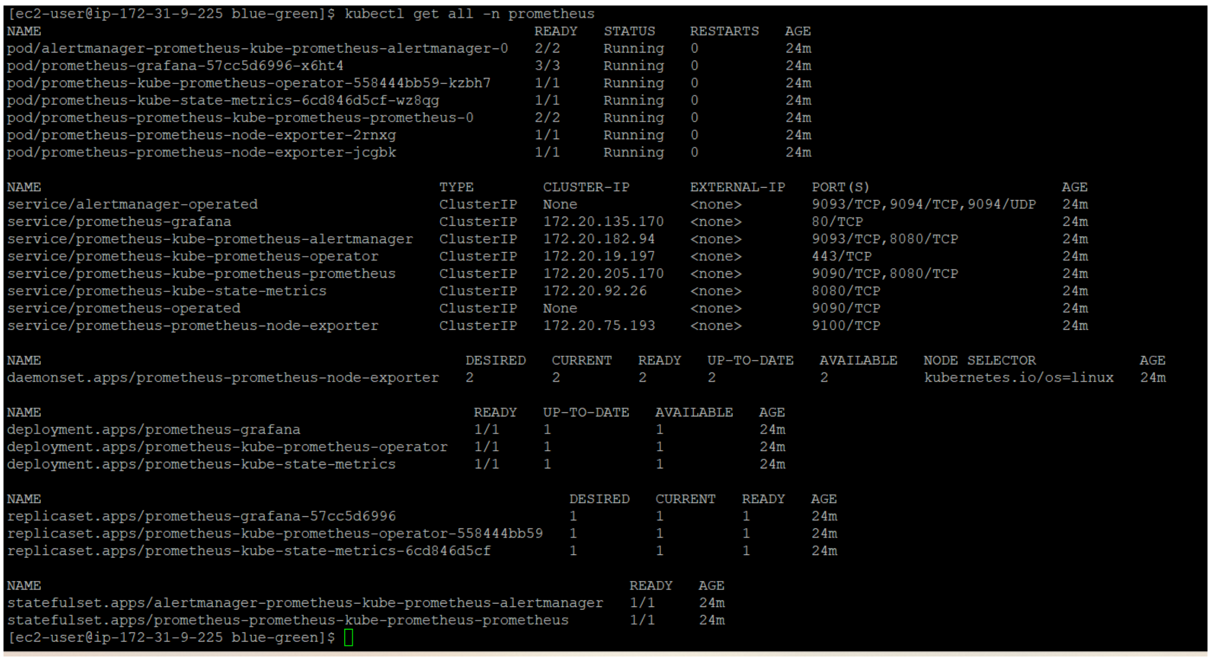
Install Prometheus

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



```
[ec2-user@ip-172-31-9-225 blue-green]$ helm install prometheus prometheus-community/kube-prometheus-stack -n prometheus
NAME: prometheus
LAST DEPLOYED: Thu Jan  4 10:59:12 2024
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=prometheus"
```

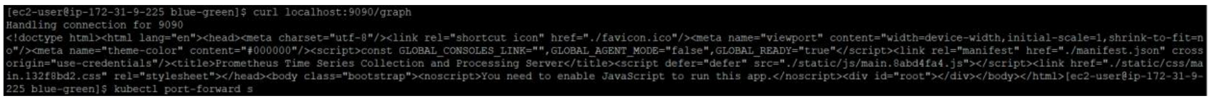
View the Prometheus dashboard by forwarding the deployment ports



```
[ec2-user@ip-172-31-9-225 blue-green]$ kubectl get all -n prometheus
NAME                                                        READY   STATUS    RESTARTS   AGE
pod/alertmanager-prometheus-kube-prometheus-alertmanager-0  2/2     Running   0          24m
pod/prometheus-grafana-57cc5d6996-x6ht4                     3/3     Running   0          24m
pod/prometheus-kube-prometheus-operator-558444bb59-kzbh7    1/1     Running   0          24m
pod/prometheus-kube-state-metrics-6cd846d5cf-wz8qg          1/1     Running   0          24m
pod/prometheus-prometheus-kube-prometheus-prometheus-0      2/2     Running   0          24m
pod/prometheus-prometheus-node-exporter-2rnxg               1/1     Running   0          24m
pod/prometheus-prometheus-node-exporter-jcgbk               1/1     Running   0          24m

NAME                                             TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)                      AGE
service/alertmanager-operated                    ClusterIP   None             <none>        9093/TCP,9094/TCP,9094/UDP   24m
service/prometheus-grafana                       ClusterIP   172.20.135.170   <none>        80/TCP                       24m
service/prometheus-kube-prometheus-alertmanager  ClusterIP   172.20.182.94    <none>        9093/TCP,8080/TCP            24m
service/prometheus-kube-prometheus-operator      ClusterIP   172.20.19.197    <none>        443/TCP                      24m
service/prometheus-kube-prometheus-prometheus    ClusterIP   172.20.205.170   <none>        9090/TCP,8080/TCP            24m
service/prometheus-kube-state-metrics            ClusterIP   172.20.92.26     <none>        8080/TCP                     24m
service/prometheus-operated                      ClusterIP   None             <none>        9090/TCP                     24m
service/prometheus-prometheus-node-exporter      ClusterIP   172.20.75.193    <none>        9100/TCP                     24m

NAME                                                 DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE   NODE SELECTOR            AGE
daemonset.apps/prometheus-prometheus-node-exporter   2         2         2       2            2           kubernetes.io/os=linux   24m

NAME                                                  READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/prometheus-grafana                    1/1     1            1           24m
deployment.apps/prometheus-kube-prometheus-operator   1/1     1            1           24m
deployment.apps/prometheus-kube-state-metrics         1/1     1            1           24m

NAME                                                             DESIRED   CURRENT   READY   AGE
replicaset.apps/prometheus-grafana-57cc5d6996                    1         1         1       24m
replicaset.apps/prometheus-kube-prometheus-operator-558444bb59   1         1         1       24m
replicaset.apps/prometheus-kube-state-metrics-6cd846d5cf         1         1         1       24m

NAME                                                                    READY   AGE
statefulset.apps/alertmanager-prometheus-kube-prometheus-alertmanager   1/1     24m
statefulset.apps/prometheus-prometheus-kube-prometheus-prometheus       1/1     24m
[ec2-user@ip-172-31-9-225 blue-green]$ 
```

kubectl port-forward statefulset.apps/prometheus-prometheus-kube-prometheus-prometheus 9090 -n prometheus &

run `curl` `localhost:9090/graph`



```
[ec2-user@ip-172-31-9-225 blue-green]$ curl localhost:9090/graph
Handling connection for 9090
<!doctype html><html lang="en"><head><meta charset="utf-8"/><link rel="shortcut icon" href="./favicon.ico"/><meta name="viewport" content="width=device-width,initial-scale=1,shrink-to-fit=n
o"/><meta name="theme-color" content="#000000"/><script>const GLOBAL_CONSOLES_LINK="",GLOBAL_AGENT_MODE="false",GLOBAL_READY="true"</script><link rel="manifest" href="./manifest.json" cross
origin="use-credentials"/><title>Prometheus Time Series Collection and Processing Server</title><script defer="defer" src="./static/js/main.8abd4fa4.js"></script><link href="./static/css/ma
in.132f8bd2.css" rel="stylesheet"></head><body class="bootstrap"><noscript>You need to enable JavaScript to run this app.</noscript><div id="root"></div></body></html>[ec2-user@ip-172-31-9-
225 blue-green]$ kubectl port-forward s
```

Install Grafana

Add the ***Grafana*** helm chart repository. Later, Update the helm chart repository.

helm repo add grafana https://grafana.github.io/helm-charts

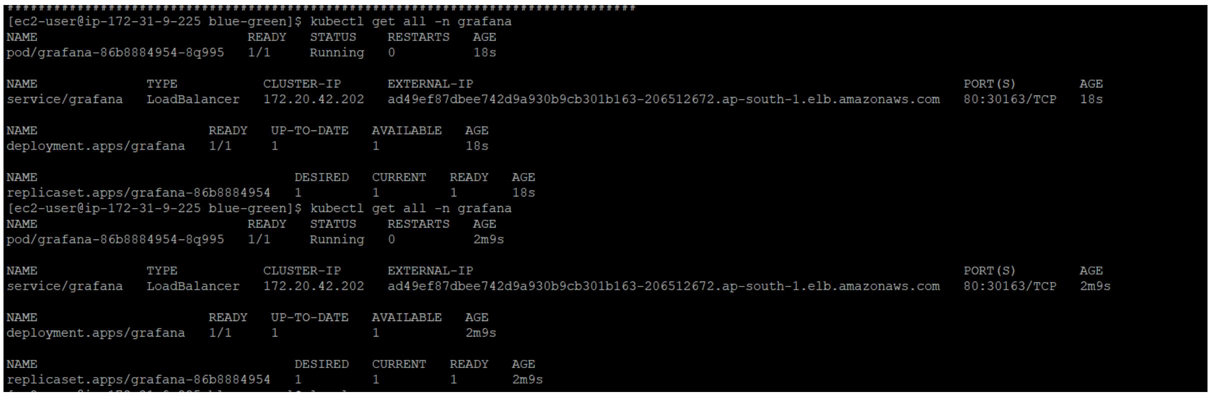
helm repo update

Create a namespace `Grafana`

kubectl create namespace Grafana

Install the Grafana

```
helm install grafana grafana/grafana \
   --namespace grafana \
   --set adminPassword='Venkat@123' \
   --set service.type=LoadBalancer
```



```
################################################################################
[ec2-user@ip-172-31-9-225 blue-green]$ kubectl get all -n grafana
NAME                           READY   STATUS    RESTARTS   AGE
pod/grafana-86b8884954-8q995   1/1     Running   0          18s

NAME              TYPE           CLUSTER-IP      EXTERNAL-IP                                                               PORT(S)        AGE
service/grafana   LoadBalancer   172.20.42.202   ad49ef87dbee742d9a930b9cb301b163-206512672.ap-south-1.elb.amazonaws.com   80:30163/TCP   18s

NAME                      READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/grafana   1/1     1            1           18s

NAME                                 DESIRED   CURRENT   READY   AGE
replicaset.apps/grafana-86b8884954   1         1         1       18s
[ec2-user@ip-172-31-9-225 blue-green]$ kubectl get all -n grafana
NAME                           READY   STATUS    RESTARTS   AGE
pod/grafana-86b8884954-8q995   1/1     Running   0          2m9s

NAME              TYPE           CLUSTER-IP      EXTERNAL-IP                                                               PORT(S)        AGE
service/grafana   LoadBalancer   172.20.42.202   ad49ef87dbee742d9a930b9cb301b163-206512672.ap-south-1.elb.amazonaws.com   80:30163/TCP   2m9s

NAME                      READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/grafana   1/1     1            1           2m9s

NAME                                 DESIRED   CURRENT   READY   AGE
replicaset.apps/grafana-86b8884954   1         1         1       2m9s
```

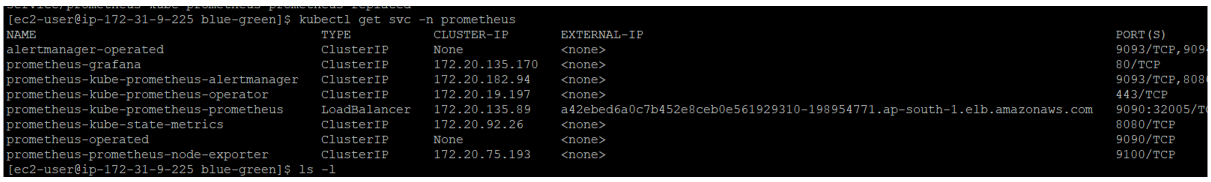
We can change the service of Prometheus to LoadBalance.

```
kubectl get service/prometheus-kube-prometheus-prometheus -n prometheus -o yaml>prometheus.yml
```

```
[ec2-user@ip-172-31-9-225 blue-green]$ cat prometheus.yml
apiVersion: v1
kind: Service
metadata:
  annotations:
    meta.helm.sh/release-name: prometheus
    meta.helm.sh/release-namespace: prometheus
  labels:
    app: kube-prometheus-stack-prometheus
    app.kubernetes.io/instance: prometheus
    app.kubernetes.io/managed-by: Helm
    app.kubernetes.io/part-of: kube-prometheus-stack
    app.kubernetes.io/version: 55.5.1
    chart: kube-prometheus-stack-55.5.1
    heritage: Helm
    release: prometheus
    self-monitor: "true"
  name: prometheus-kube-prometheus-prometheus
  namespace: prometheus
  resourceVersion: "6646"
  uid: 0e68febb-a677-49b9-86b6-85602ea04fcc
spec:
  ports:
  - name: http-web
    port: 9090
    protocol: TCP
    targetPort: 9090
  - appProtocol: http
    name: reloader-web
    port: 8080
    protocol: TCP
    targetPort: reloader-web
  selector:
    app.kubernetes.io/name: prometheus
    operator.prometheus.io/name: prometheus-kube-prometheus-prometheus
  sessionAffinity: None
  type: LoadBalancer
status:
  loadBalancer: {}
[ec2-user@ip-172-31-9-225 blue-green]$
```

kubectl replace -f prometheus.yml –force

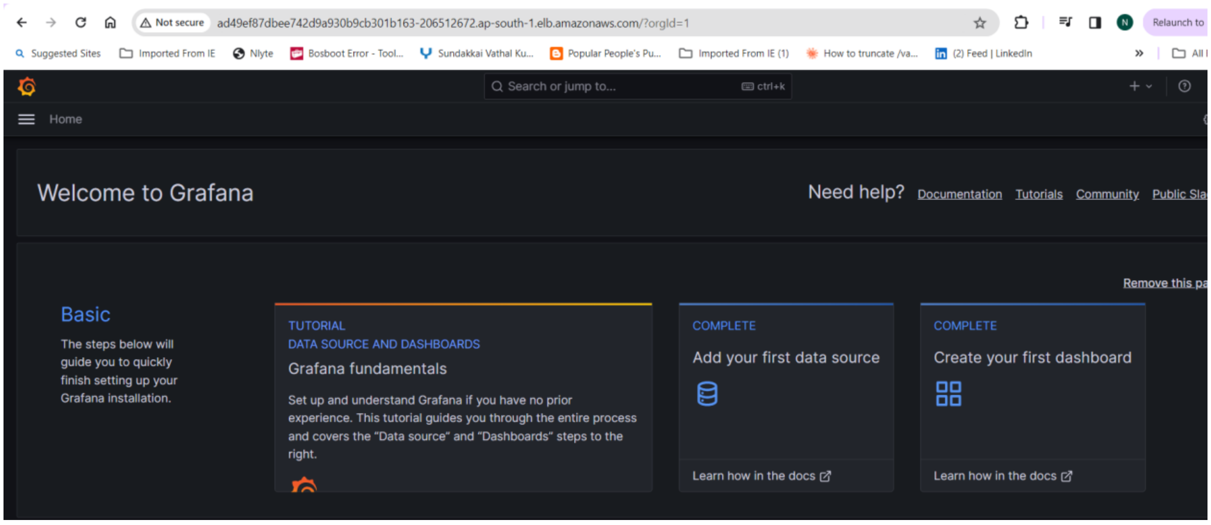
above command will replace the service to LOadbalancer.



```
[ec2-user@ip-172-31-9-225 blue-green]$ kubectl get svc -n prometheus
NAME                                      TYPE           CLUSTER-IP       EXTERNAL-IP                                                                   PORT(S)
alertmanager-operated                     ClusterIP      None             <none>                                                                        9093/TCP,909
prometheus-grafana                        ClusterIP      172.20.135.170   <none>                                                                        80/TCP
prometheus-kube-prometheus-alertmanager   ClusterIP      172.20.182.94    <none>                                                                        9093/TCP,808
prometheus-kube-prometheus-operator       ClusterIP      172.20.19.197    <none>                                                                        443/TCP
prometheus-kube-prometheus-prometheus     LoadBalancer   172.20.135.89    a42ebed6a0c7b452e8ceb0e561929310-198954771.ap-south-1.elb.amazonaws.com   9090:32005/T
prometheus-kube-state-metrics             ClusterIP      172.20.92.26     <none>                                                                        8080/TCP
prometheus-operated                       ClusterIP      None             <none>                                                                        9090/TCP
prometheus-prometheus-node-exporter       ClusterIP      172.20.75.193    <none>                                                                        9100/TCP
[ec2-user@ip-172-31-9-225 blue-green]$ ls -l
```

We can access the Graphana using Loadbalancer URL.

http:// ad49ef87dbee742d9a930b9cb301b163-206512672.ap-south-1.elb.amazonaws.com:80



Not secure ad49ef87dbee742d9a930b9cb301b163-206512672.ap-south-1.elb.amazonaws.com/?orgId=1

Home

Welcome to Grafana

Need help? Documentation Tutorials Community Public Sla

Remove this pa

Basic

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

TUTORIAL
DATA SOURCE AND DASHBOARDS
Grafana fundamentals

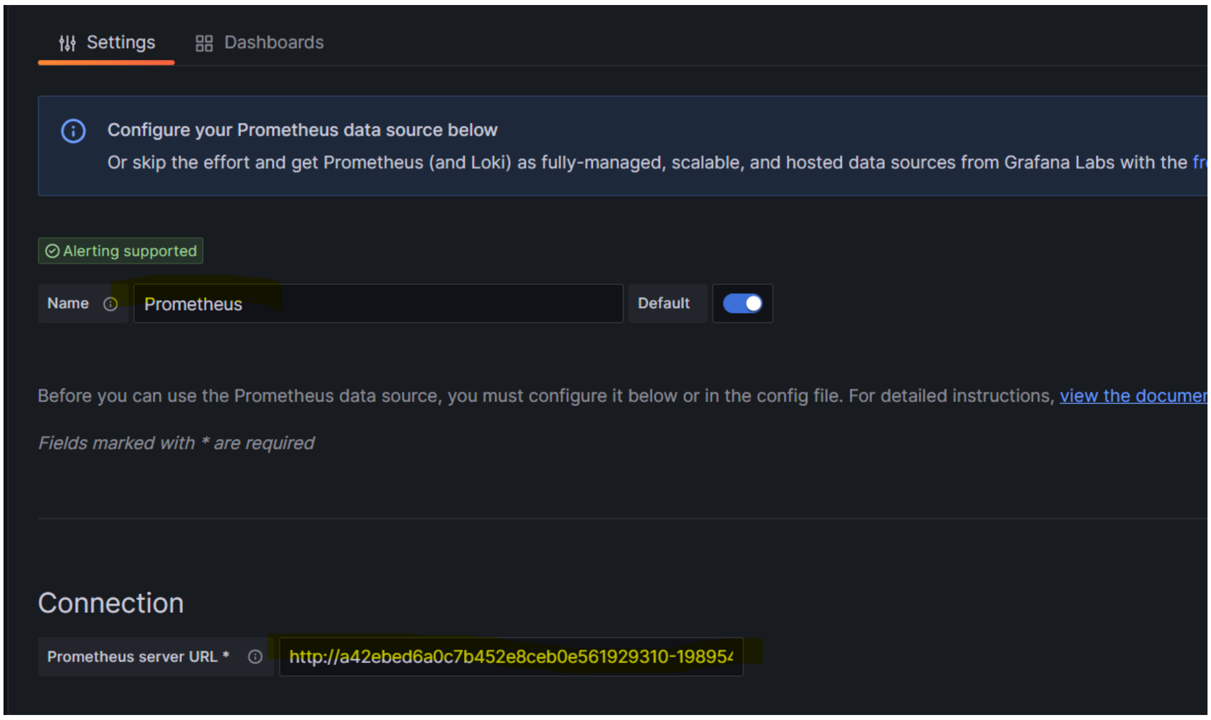
Set up and understand Grafana if you have no prior experience. This tutorial guides you through the entire process and covers the "Data source" and "Dashboards" steps to the right.

COMPLETE
Add your first data source
Learn how in the docs

COMPLETE
Create your first dashboard
Learn how in the docs

Now we can add our Prometheus to it.

Home -> connections -> Datasource - > Add



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 fr

Alerting supported

Name: Prometheus   Default: on

Before you can use the Prometheus data source, you must configure it below or in the config file. For detailed instructions, view the documer

*Fields marked with * are required*

Connection

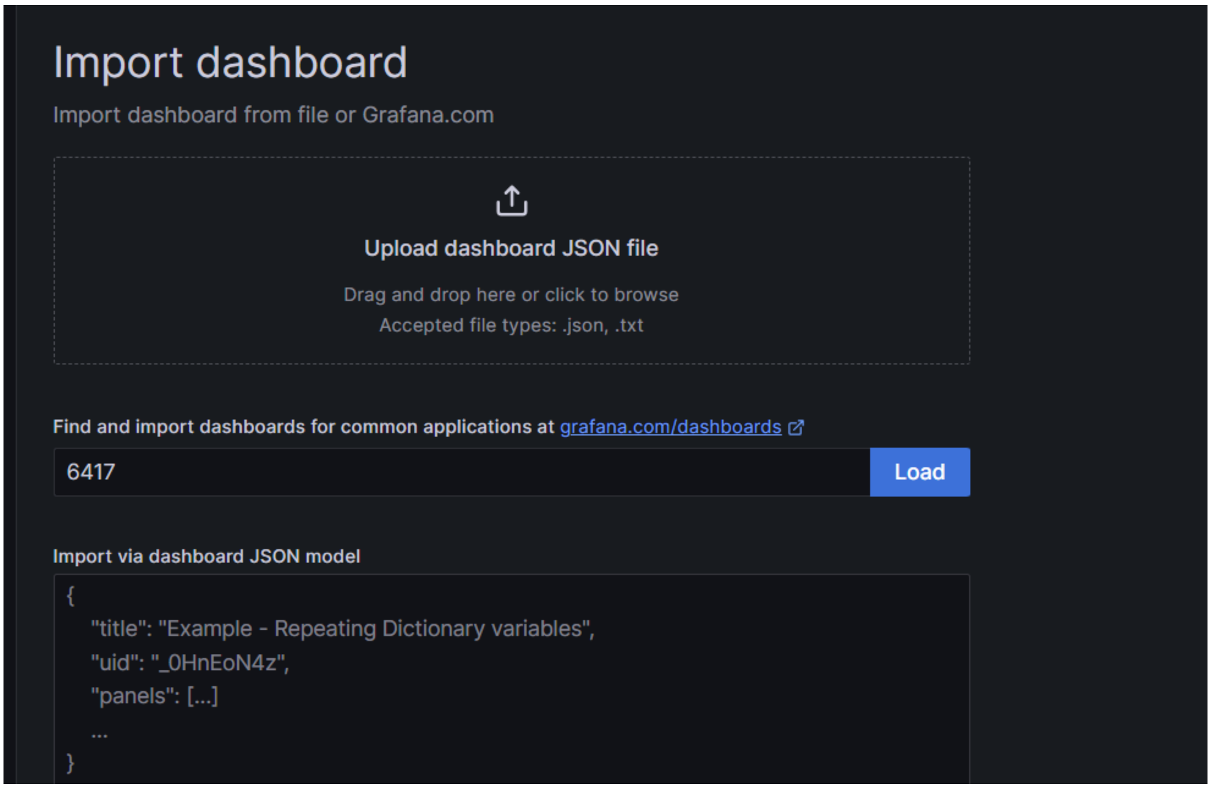
Prometheus server URL *: http://a42ebed6a0c7b452e8ceb0e561929310-19895

Give name and Prometheus service Loadbalancer URL.

Now we create dashboard.

Home -> dashboard -> New -> Import(we use existing Grafana dashboard)

“6417” dashboard of Kubernetes.



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

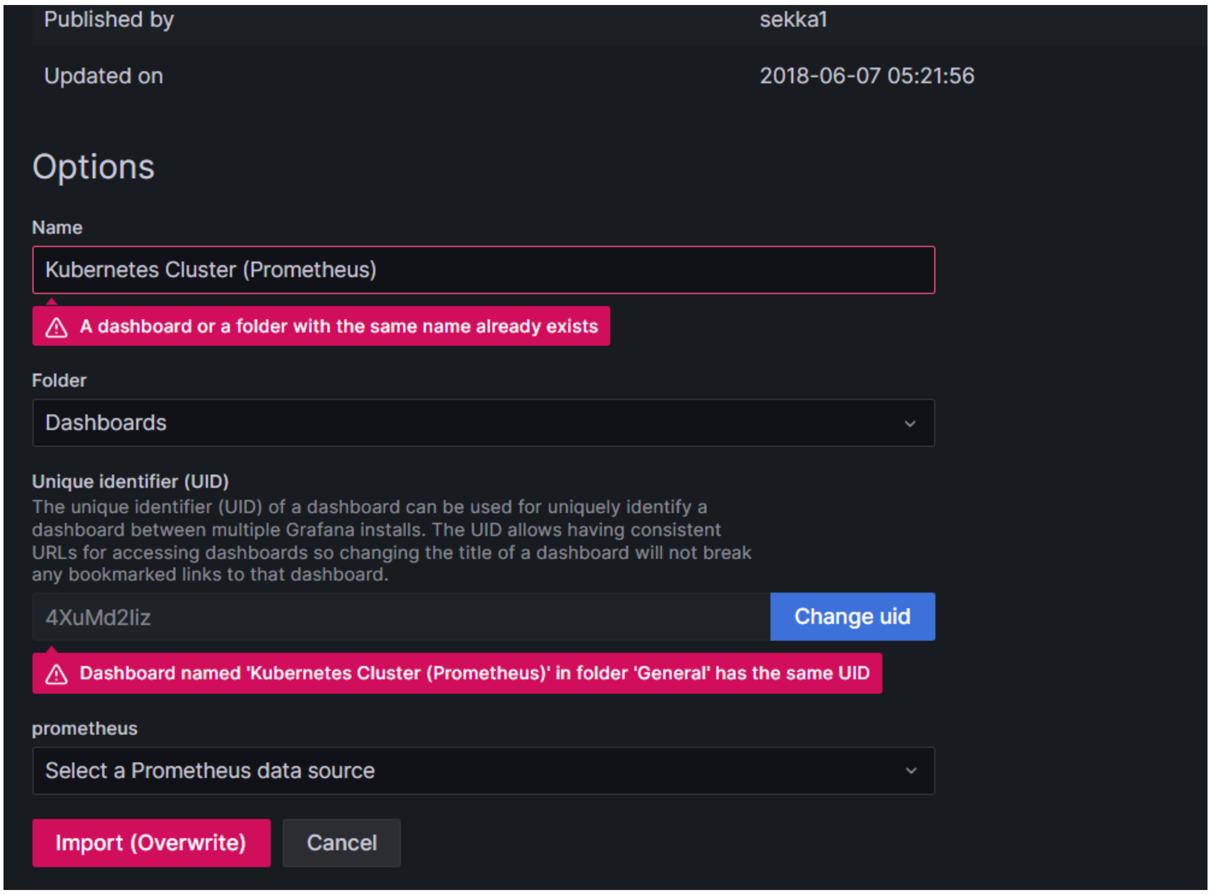
6417

Load

Import via dashboard JSON model

```
{
   "title": "Example - Repeating Dictionary variables",
   "uid": "_0HnEoN4z",
   "panels": [...]
   ...
}
```

Give name, Select the Prometheus we have created. Then import it.



Published by sekka1

Updated on 2018-06-07 05:21:56

Options

Name

Kubernetes Cluster (Prometheus)

⚠ A dashboard or a folder with the same name already exists

Folder

Dashboards

Unique identifier (UID)

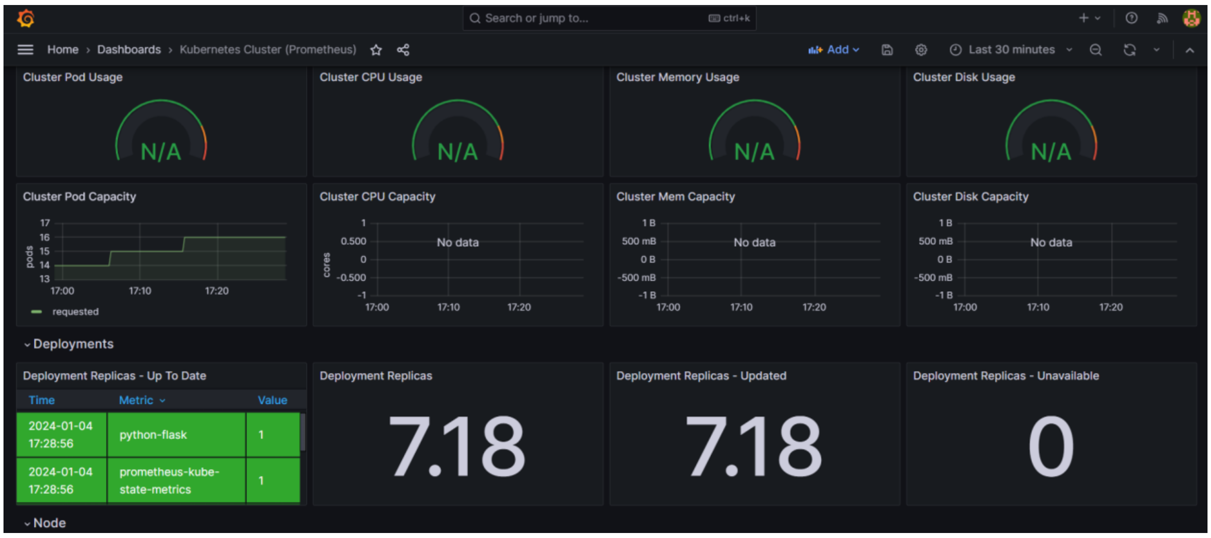
The unique identifier (UID) of a dashboard can be used for uniquely identify a dashboard between multiple Grafana installs. The UID allows having consistent URLs for accessing dashboards so changing the title of a dashboard will not break any bookmarked links to that dashboard.

4XuMd2liz

⚠ Dashboard named 'Kubernetes Cluster (Prometheus)' in folder 'General' has the same UID

prometheus

Select a Prometheus data source



Deployment Replicas - Up To Date

Time	Metric	Value
2024-01-04 17:28:56	python-flask	1
2024-01-04 17:28:56	prometheus-kube-state-metrics	1