Kubernetes Setup for Prometheus and Grafana

To quickly start all the things just do this:

SSH to your AWS instance and run the below command to deploy the granfa and prometheus deployments in a seperate namesapce.

```
$ sudo mkdir monitoring
$ cd monitoring/
$ git clone https://github.com/giantswarm/prometheus.git
$ cd prometheus/manifests/grafana
$ vim deployment.yaml
```

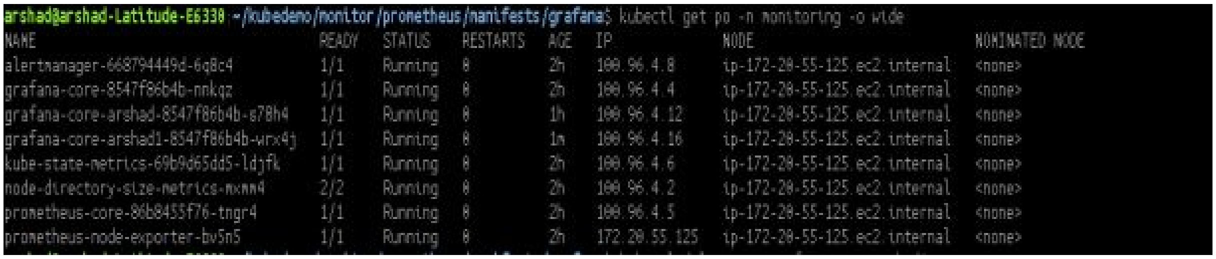
Just add your name after **grafana-core** in the below field.

```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: grafana-core-<your-name>
```

Save and exit (:wq)

$ kubectl create -f deployment.yaml
$ kubectl expose deployment grafana-core-<your-name> --type=**NodePort** --port=3000 -n monitoring

To get the details of the node where the Grafana app has been deployed do,
$ kubectl get po -n monitoring -o wide



```
arshad@arshad-Latitude-E6330 ~/kubedemo/monitor/prometheus/manifests/grafana$ kubectl get po -n monitoring -o wide
NAME                                  READY   STATUS    RESTARTS   AGE   IP              NODE                           NOMINATED NODE
alertmanager-668794449d-6q8c4         1/1     Running   0          2h    100.96.4.8      ip-172-20-55-125.ec2.internal  <none>
grafana-core-8547f86b4b-nnkqz         1/1     Running   0          2h    100.96.4.4      ip-172-20-55-125.ec2.internal  <none>
grafana-core-arshad-8547f86b4b-s78h4  1/1     Running   0          1h    100.96.4.12     ip-172-20-55-125.ec2.internal  <none>
grafana-core-arshad1-8547f86b4b-wrx4j 1/1     Running   0          1m    100.96.4.16     ip-172-20-55-125.ec2.internal  <none>
kube-state-metrics-69b9d65dd5-ldjfk   1/1     Running   0          2h    100.96.4.6      ip-172-20-55-125.ec2.internal  <none>
node-directory-size-metrics-mxmm4     2/2     Running   0          2h    100.96.4.2      ip-172-20-55-125.ec2.internal  <none>
prometheus-core-86b8455f76-tngr4      1/1     Running   0          2h    100.96.4.5      ip-172-20-55-125.ec2.internal  <none>
prometheus-node-exporter-bv5n5        1/1     Running   0          2h    172.20.55.125   ip-172-20-55-125.ec2.internal  <none>
```

Note down the NODE details.
In this example our POD has been deployed on Node
grafana-core-arshad1-1/1   Running   0 1m   100.96.4.16   **ip-172-20-55-125.ec2.internal**

Now, Login to the AWS Console and find the node **ip-172-20-55-125.ec2.internal**

Copy the Public-IP of the Node where your POD has been deployed

ACCESS Grafana through the NodePort

Remember we exposed our app through NodePort, and to check the Port on which our pod has been exposed, do.

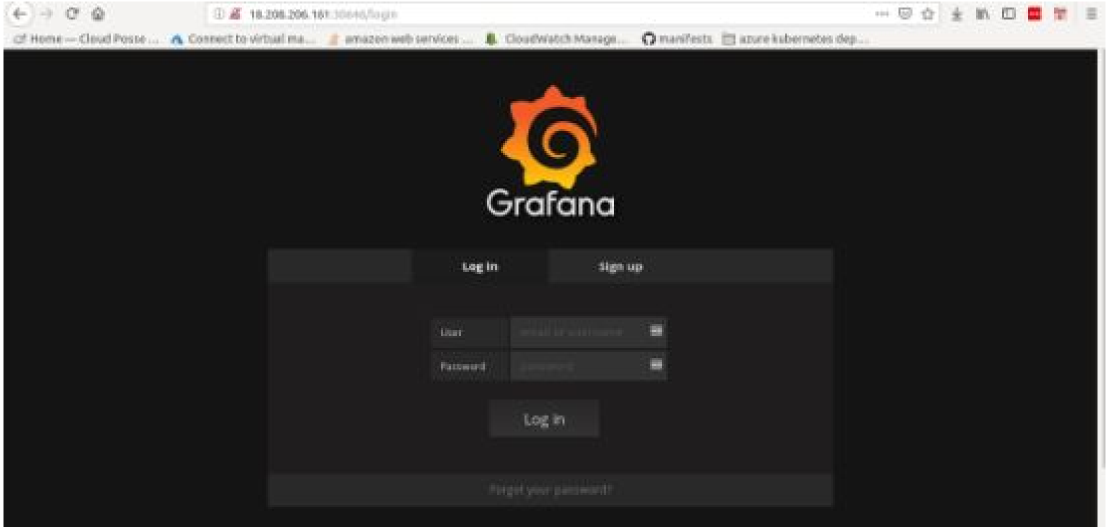
```
$ kubectl get svc -n monitoring | grep <your-name>
```

```
grafana-core-arshad1    NodePort   100.67.205.182   <none>   3000:30646/TCP   10m
```

To access the Application

http://<Public-IP of the Node>:NodePort
Example
http://18.208.206.161:30646/
Wait for few minutes before trying to access the Grafana dashboard

The default credentials are
Username : **admin**
Password : **admin**



Configure Prometheus data source for Grafana.

To find out the prometheus url.

```
$ kubectl get po -n monitoring -o wide ####Note down the NODE Name
$ kubectl get svc -n monitoring          ##### Node down the Port Prometheus is exposed to
```

Grafana UI >>> **Data Sources** >>> **Add data source**

- Name: prometheus
- Type: Prometheus
- Url: http://<Public-Ip-of-Worker-Node>:<Port Number> ##check step above
- Access : Proxy
- Add
- Import Prometheus Stats:
- **Grafana** UI >>> **Dashboards** >>> **Import**
- Grafana.net Dashboard: https://grafana.com/dashboards/737
- Load
- Prometheus: prometheus
- Import (Save & Open)

Import Kubernetes cluster monitoring

- Grafana UI / Dashboards / Import
- Grafana.net Dashboard: https://grafana.com/dashboards/315
- Load
- Prometheus: prometheus
- Save & Open