

Module 2: Prometheus Security and Use Cases

Use Case 1

edureka!

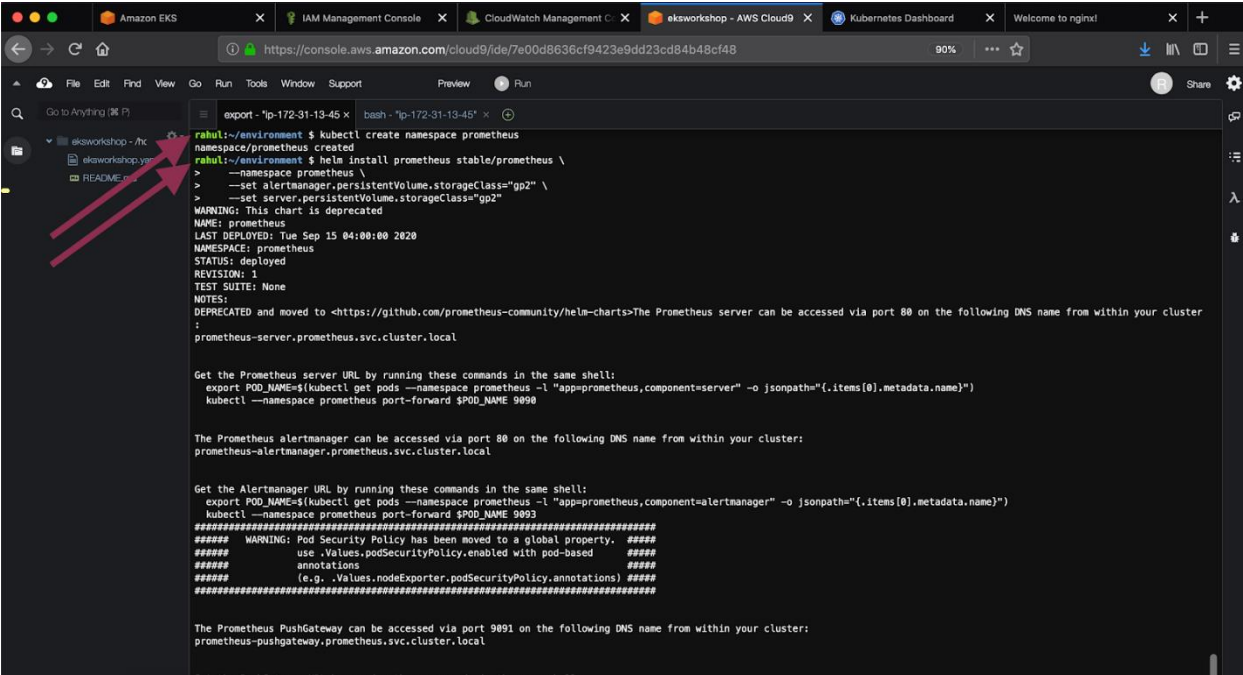
edureka!

© Brain4ce Education Solutions Pvt. Ltd.

Use Case 1: Prometheus Security and Use Cases

1. Deploy Prometheus

```
kubectl create namespace prometheus
helm install prometheus stable/prometheus \
  --namespace prometheus \
  --set alertmanager.persistentVolume.storageClass="gp2" \
  --set server.persistentVolume.storageClass="gp2"
```



```
export -lp-172-31-13-45 x bash -lp-172-31-13-45 x
rahu!:/environment $ kubectl create namespace prometheus
namespace/prometheus created
rahu!:/environment $ helm install prometheus stable/prometheus \
> --namespace prometheus \
> --set alertmanager.persistentVolume.storageClass="gp2" \
> --set server.persistentVolume.storageClass="gp2"
WARNING: This chart is deprecated
NAME: prometheus
LAST DEPLOYED: Tue Sep 15 04:00:00 2020
NAMESPACE: prometheus
STATUS: deployed
REVISION: 1
TEST SUITE: None
NOTES:
DEPRECATED and moved to <https://github.com/prometheus-community/helm-charts>The Prometheus server can be accessed via port 80 on the following DNS name from within your cluster:
prometheus-server.prometheus.svc.cluster.local

Get the Prometheus server URL by running these commands in the same shell:
export POD_NAME=$(kubectl get pods --namespace prometheus -l "app=prometheus,component=server" -o jsonpath="{.items[0].metadata.name}")
kubectl --namespace prometheus port-forward $POD_NAME 9090

The Prometheus alertmanager can be accessed via port 80 on the following DNS name from within your cluster:
prometheus-alertmanager.prometheus.svc.cluster.local

Get the Alertmanager URL by running these commands in the same shell:
export POD_NAME=$(kubectl get pods --namespace prometheus -l "app=prometheus,component=alertmanager" -o jsonpath="{.items[0].metadata.name}")
kubectl --namespace prometheus port-forward $POD_NAME 9093

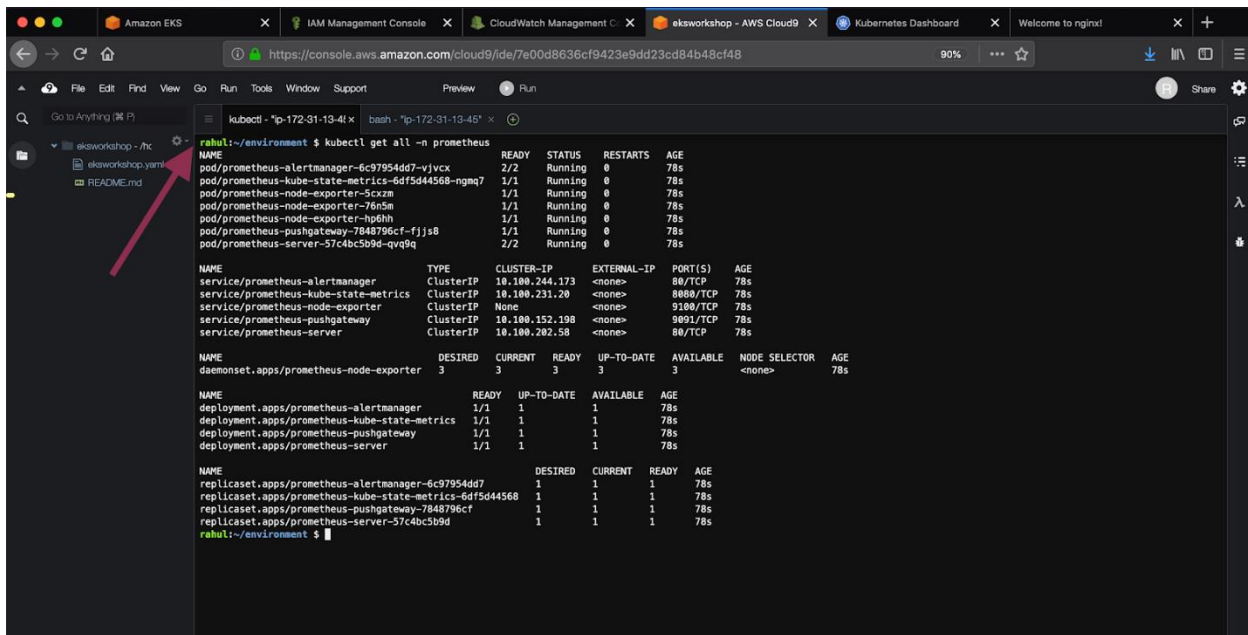
##### WARNING: Pod Security Policy has been moved to a global property. #####
##### use .Values.podSecurityPolicy.enabled with pod-based #####
##### annotations #####
##### (e.g., .Values.nodeExporter.podSecurityPolicy.annotations) #####

The Prometheus PushGateway can be accessed via port 9091 on the following DNS name from within your cluster:
prometheus-pushgateway.prometheus.svc.cluster.local

Get the PushGateway URL by running these commands in the same shell:
```

2. Check if Prometheus components deployed as expected

```
kubectl get all -n prometheus
```



```
rahu1:~/environment $ kubectl get all -n prometheus
NAME                                READY   STATUS    RESTARTS   AGE
pod/prometheus-alertmanager-6c97954dd7-vjvcx   1/1     Running   0           78s
pod/prometheus-kube-state-metrics-6df5d44568-ngmq7  1/1     Running   0           78s
pod/prometheus-node-exporter-5cxzm             1/1     Running   0           78s
pod/prometheus-node-exporter-76n5m             1/1     Running   0           78s
pod/prometheus-node-exporter-hp6m             1/1     Running   0           78s
pod/prometheus-pushgateway-7848796cf-fjjs8      1/1     Running   0           78s
pod/prometheus-server-57c4bc5b9d-qvq9q         2/2     Running   0           78s

NAME                                TYPE          CLUSTER-IP      EXTERNAL-IP      PORT(S)          AGE
service/prometheus-alertmanager      ClusterIP      10.100.244.173  <none>           80/TCP           78s
service/prometheus-kube-state-metrics ClusterIP      10.100.231.20  <none>           8080/TCP         78s
service/prometheus-node-exporter     ClusterIP      None           <none>           9100/TCP         78s
service/prometheus-pushgateway       ClusterIP      10.100.152.198 <none>           9091/TCP         78s
service/prometheus-server            ClusterIP      10.100.202.58  <none>           80/TCP           78s

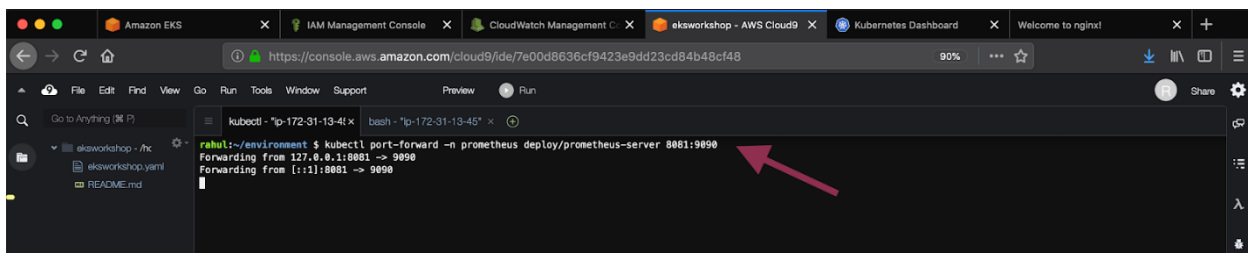
NAME                                DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE   NODE SELECTOR   AGE
daemonset.apps/prometheus-node-exporter 3          3         3       3            3          <none>          78s

NAME                                READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/prometheus-alertmanager 1/1     1            1          78s
deployment.apps/prometheus-kube-state-metrics 1/1     1            1          78s
deployment.apps/prometheus-pushgateway 1/1     1            1          78s
deployment.apps/prometheus-server 1/1     1            1          78s

NAME                                DESIRED   CURRENT   READY   AGE
replicaset.apps/prometheus-alertmanager-6c97954dd7 1          1          1       78s
replicaset.apps/prometheus-kube-state-metrics-6df5d44568 1          1          1       78s
replicaset.apps/prometheus-pushgateway-7848796cf 1          1          1       78s
replicaset.apps/prometheus-server-57c4bc5b9d 1          1          1       78s
rahu1:~/environment $
```

3. In order to access the Prometheus server URL, we are going to use the [kubectl port-forward](#) command to access the application. In Cloud9, run:

```
kubectl port-forward -n prometheus deploy/prometheus-server 8080:9090
```



```
rahu1:~/environment $ kubectl port-forward -n prometheus deploy/prometheus-server 8080:9090
Forwarding from 127.0.0.1:8081 -> 9090
Forwarding from [::]:8081 -> 9090
```

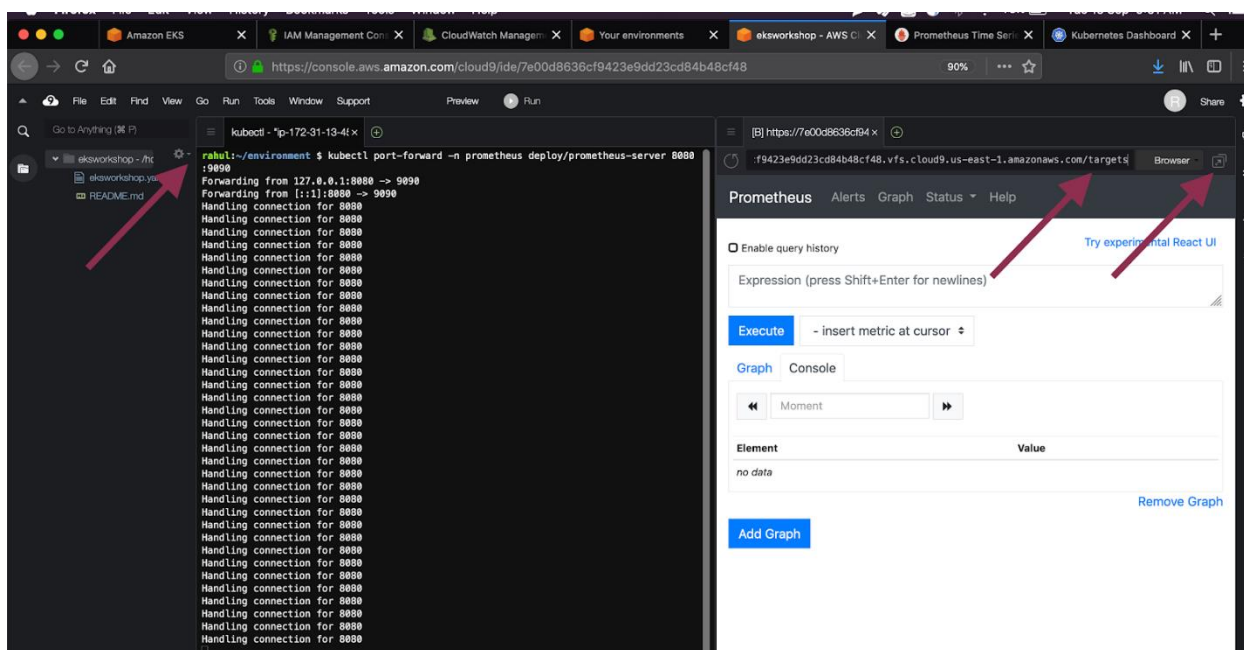
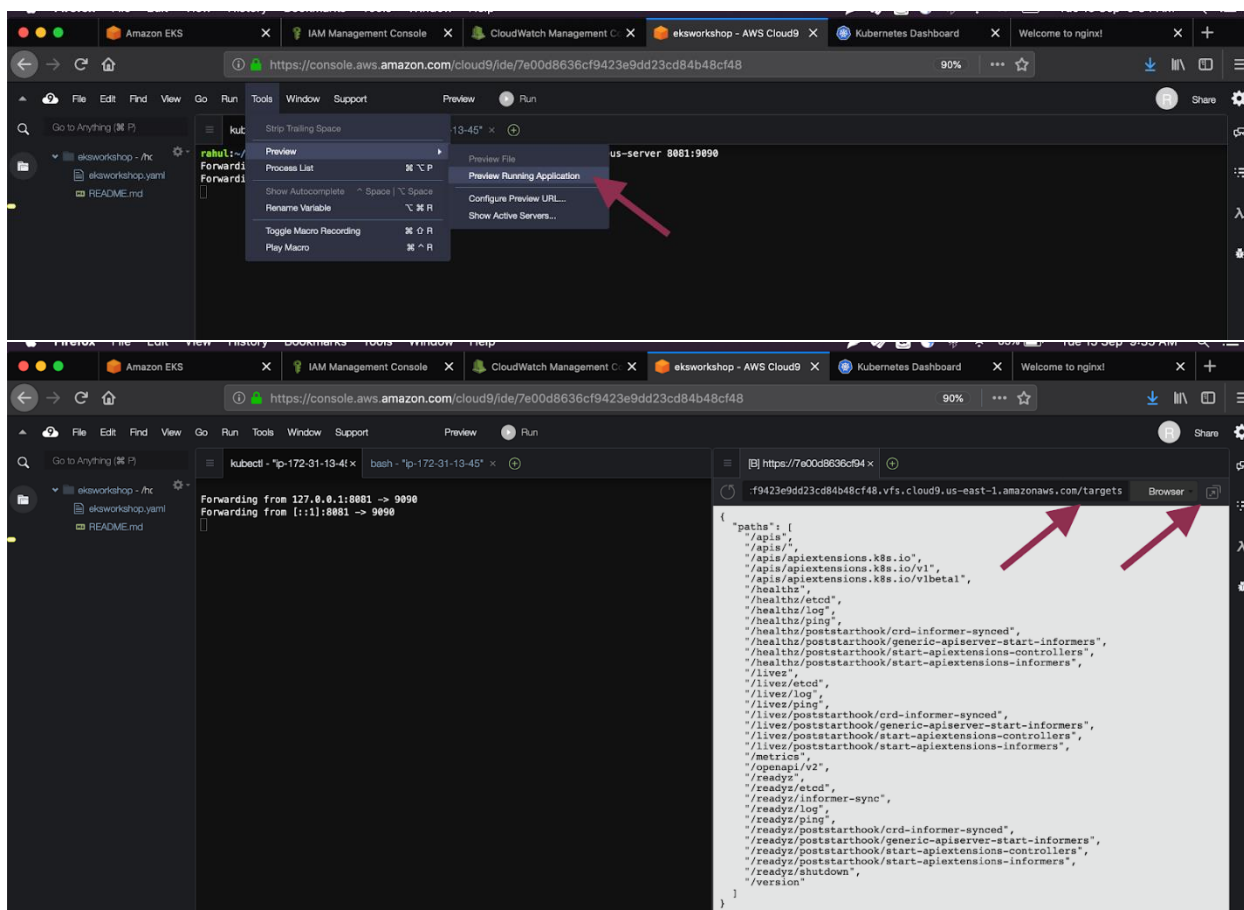
Note: In case you get an error in port forwarding saying port already in use then run the below commands to kill the process and rerun the above command. This issue could probably come because we deployed nginx earlier which used port 8080.

```
fuser -k 8080/tcp
```

In your Cloud9 environment, click Tools / Preview / Preview Running Application. Scroll to the end of the URL and append:

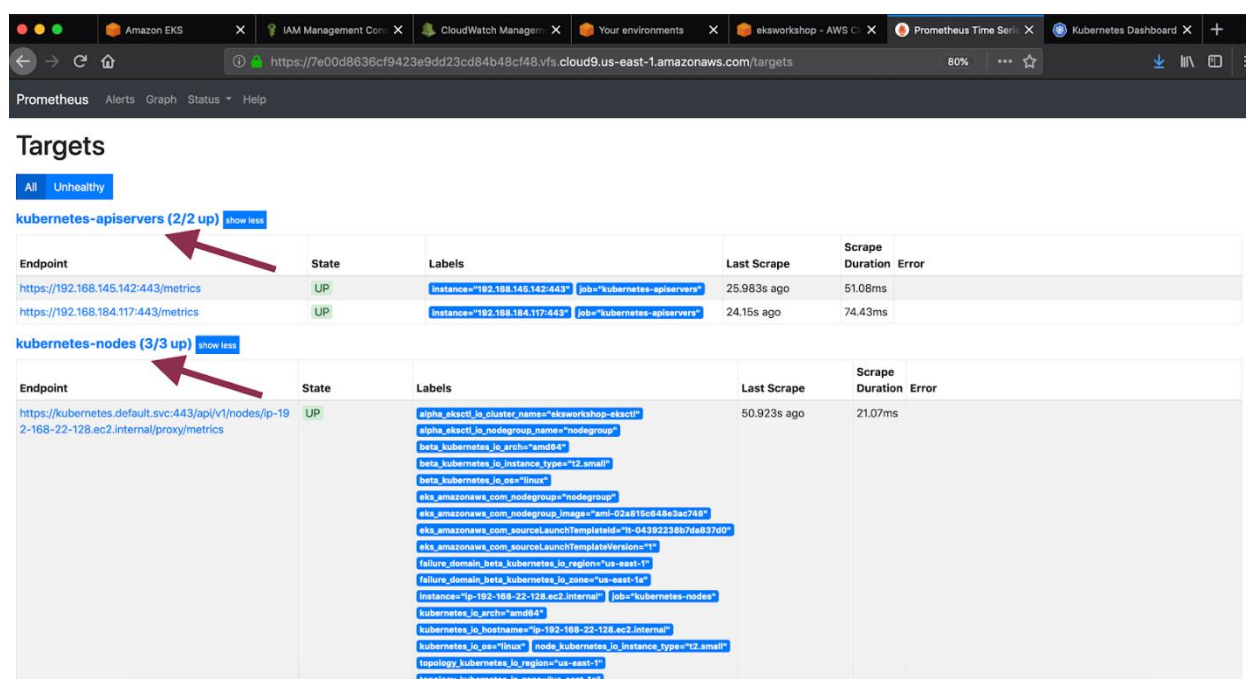
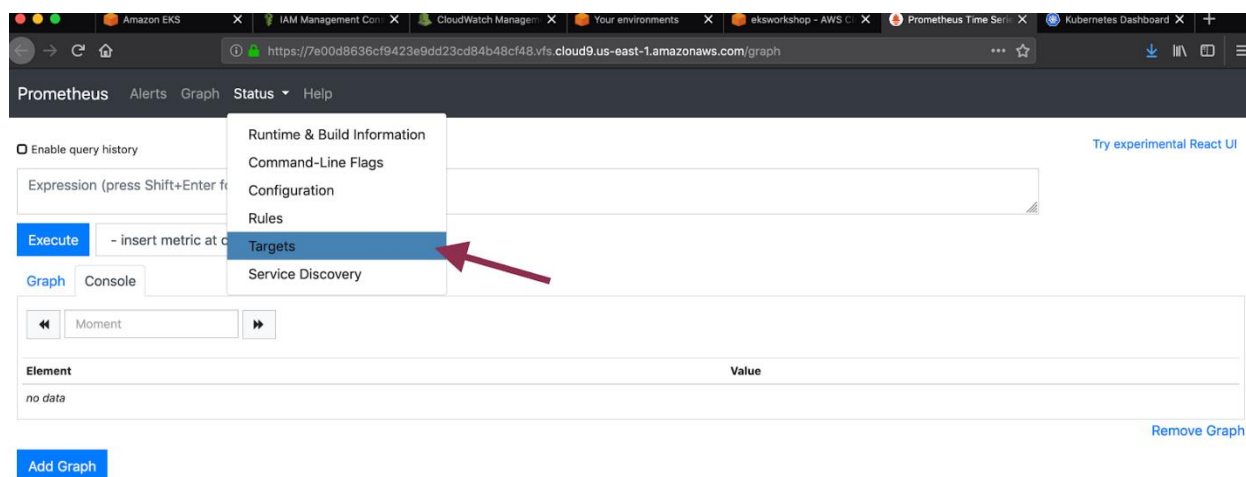
```
/targets
```

Continuous Monitoring and DevOps on AWS



Prometheus will be launched; you can see the targets in the Status Drop Down > Targets.

Continuous Monitoring and DevOps on AWS



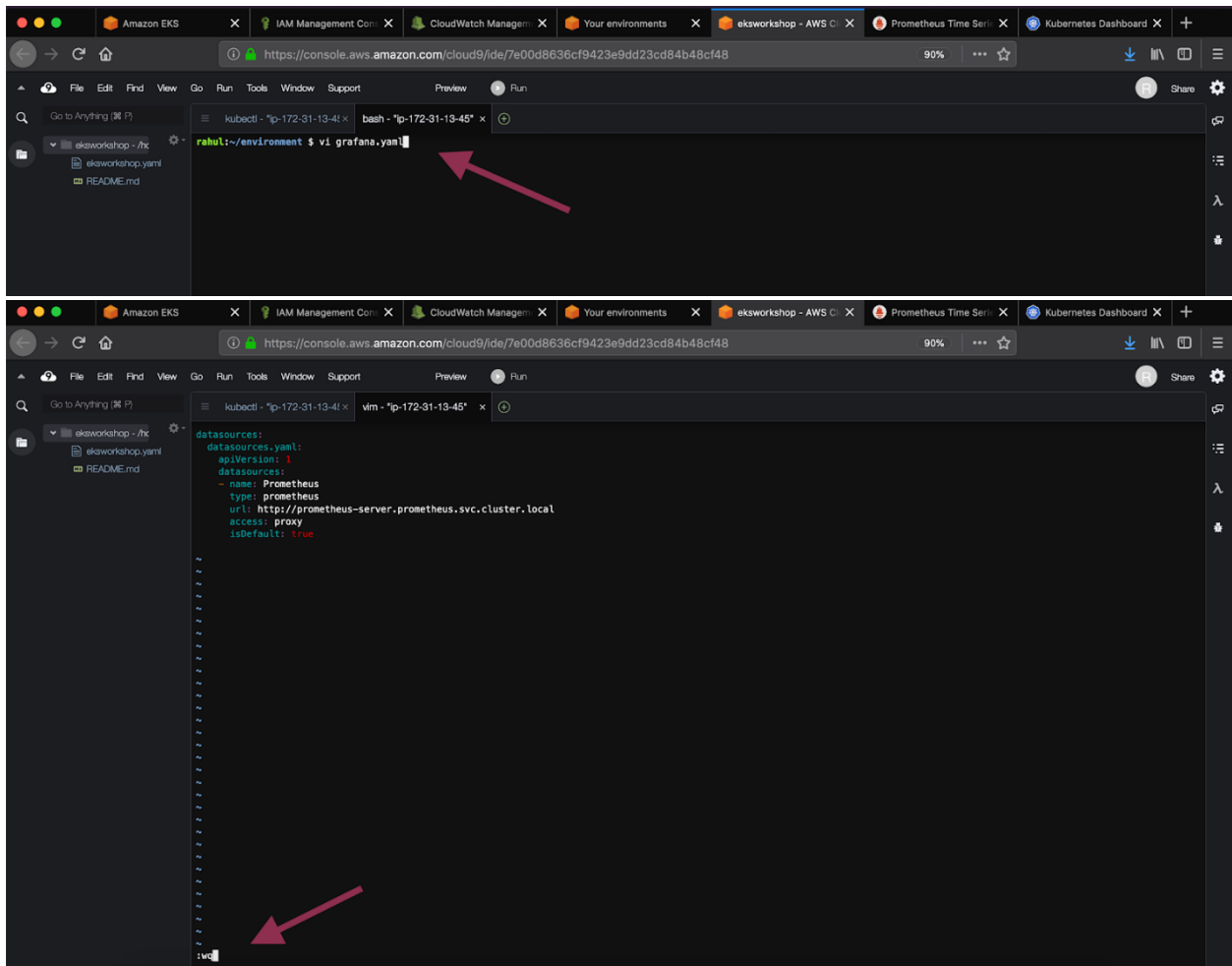
2. Deploy Grafana

We are now going to install Grafana.

1. Create YAML file called **grafana.yaml** with following values:

```
datasources:
  datasources.yaml:
    apiVersion: 1
    datasources:
      - name: Prometheus
```

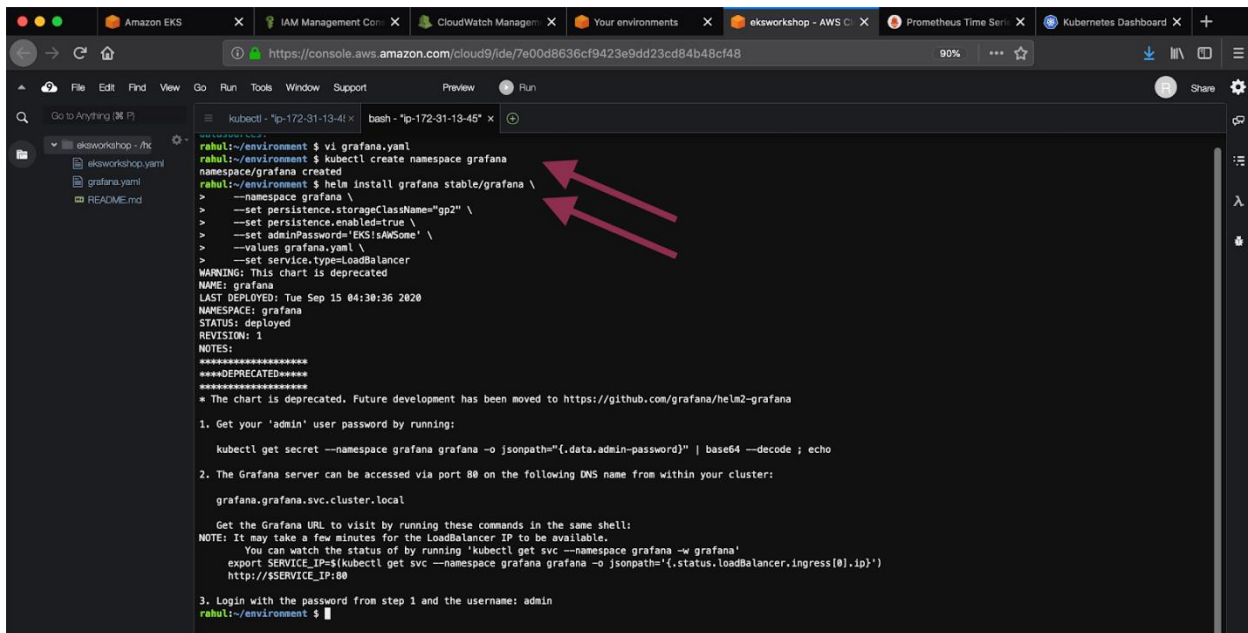
```
type: prometheus
url: http://prometheus-server.prometheus.svc.cluster.local
access: proxy
isDefault: true
```



2. Install grafana using the yaml file.

```
kubectl create namespace grafana
helm install grafana stable/grafana \
  --namespace grafana \
  --set persistence.storageClassName="gp2" \
  --set persistence.enabled=true \
  --set adminPassword='EKS!sAWSome' \
  --values grafana.yaml \
  --set service.type=LoadBalancer
```

Continuous Monitoring and DevOps on AWS



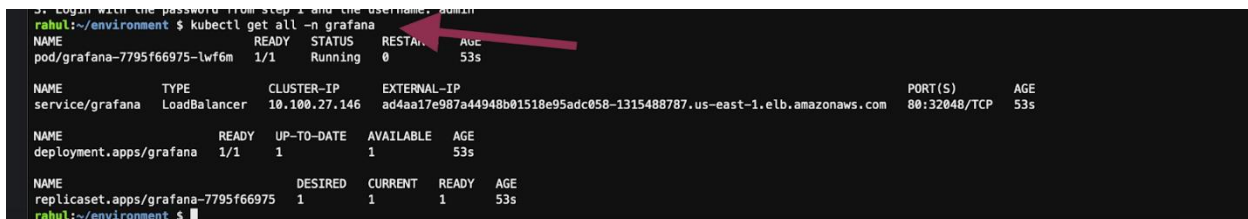
The screenshot shows the AWS console with a terminal window open in the 'eksworkshop - AWS C...' environment. The terminal displays the following commands and output:

```
kubectl -n grafana create namespace grafana
namespace/grafana created
kubectl -n grafana helm install grafana stable/grafana \
  --namespace grafana \
  --set persistence.storageClassName="gp2" \
  --set persistence.enabled=true \
  --set adminPassword='EKS!@WSome' \
  --values grafana.yaml \
  --set service.type=LoadBalancer
WARNING: This chart is deprecated
NAME: grafana
LAST DEPLOYED: Tue Sep 15 04:30:36 2020
NAMESPACE: grafana
STATUS: deployed
REVISION: 1
NOTES:
*****
****DEPRECATED****
*****
* The chart is deprecated. Future development has been moved to https://github.com/grafana/helm2-grafana
1. Get your 'admin' user password by running:
   kubectl get secret --namespace grafana 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-grafana.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 grafana -w grafana'
   export SERVICE_IP=$(kubectl get svc --namespace grafana grafana -o jsonpath='{.status.loadBalancer.ingress[0].ip}')
   http://$SERVICE_IP:80
3. Login with the password from step 1 and the username: admin
rahul:~/environment $
```

Two red arrows point to the 'namespace/grafana created' and 'helm install grafana stable/grafana' lines.

3. Run the following command to check if Grafana is deployed properly:

```
kubectl get all -n grafana
```



The screenshot shows the terminal output of the command 'kubectl get all -n grafana'. The output is as follows:

```
NAME                                READY    STATUS    RESTARTS   AGE
pod/grafana-7795f66975-lwf6m        1/1     Running   0           53s

NAME                TYPE          CLUSTER-IP    EXTERNAL-IP    PORT(S)          AGE
service/grafana     LoadBalancer  10.100.27.146  ad4aa17e987a44948b01518e95adc058-1315488787.us-east-1.elb.amazonaws.com  80:32048/TCP    53s

NAME                READY    UP-TO-DATE    AVAILABLE    AGE
deployment.apps/grafana  1/1     1              1            53s

NAME                DESIRED    CURRENT    READY    AGE
replicaset.apps/grafana  1          1          1        53s
rahul:~/environment $
```

A red arrow points to the 'RESTARTS' column in the pod table.

You can get the Grafana ELB URL using this command. Copy & Paste the value into the browser to access Grafana web UI.

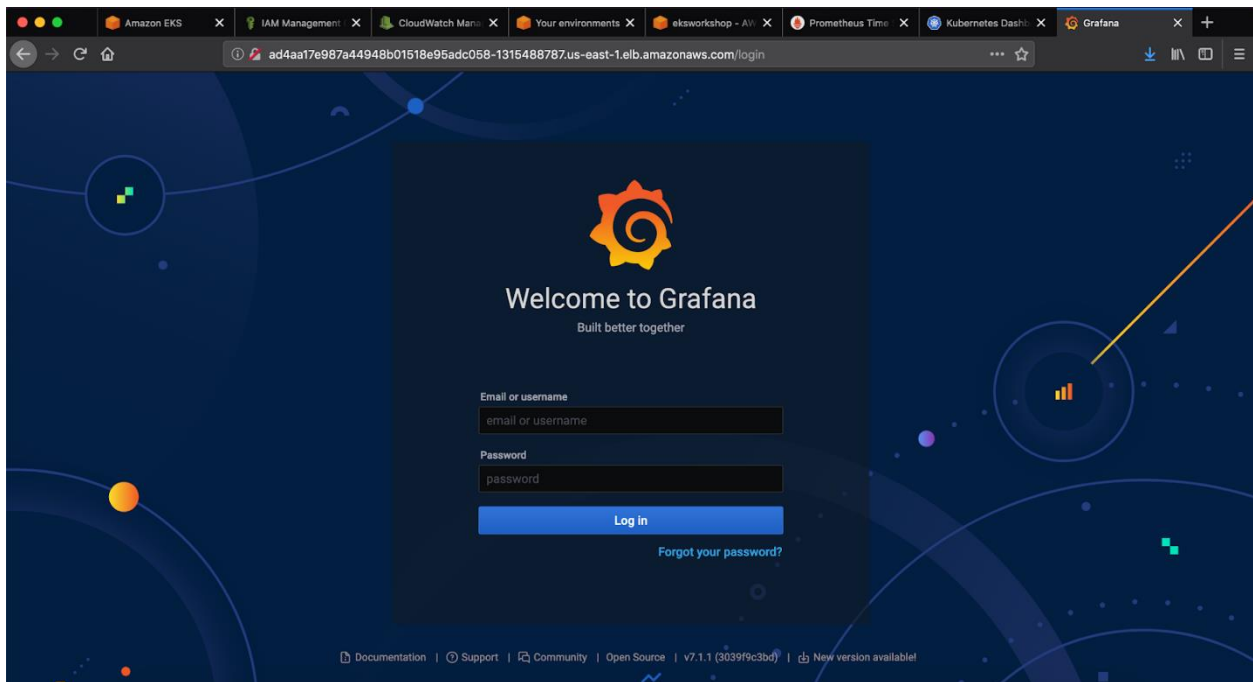
```
export ELB=$(kubectl get svc -n grafana grafana -o
jsonpath='{.status.loadBalancer.ingress[0].hostname}')
echo http://\$ELB
```



The screenshot shows the terminal output of the command 'export ELB=\$(kubectl get svc -n grafana grafana -o jsonpath='{.status.loadBalancer.ingress[0].hostname})'. The output is as follows:

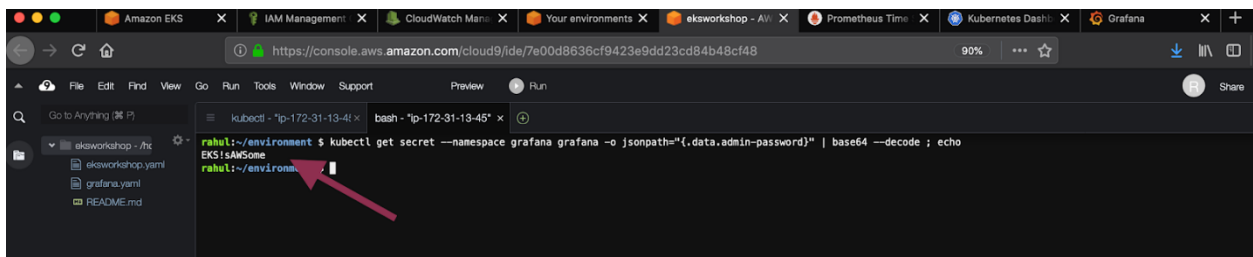
```
rahul:~/environment $ export ELB=$(kubectl get svc -n grafana grafana -o jsonpath='{.status.loadBalancer.ingress[0].hostname}')
rahul:~/environment $
rahul:~/environment $ echo "http://$ELB"
http://ad4aa17e987a44948b01518e95adc058-1315488787.us-east-1.elb.amazonaws.com
rahul:~/environment $
```

A red arrow points to the output of the 'echo' command.



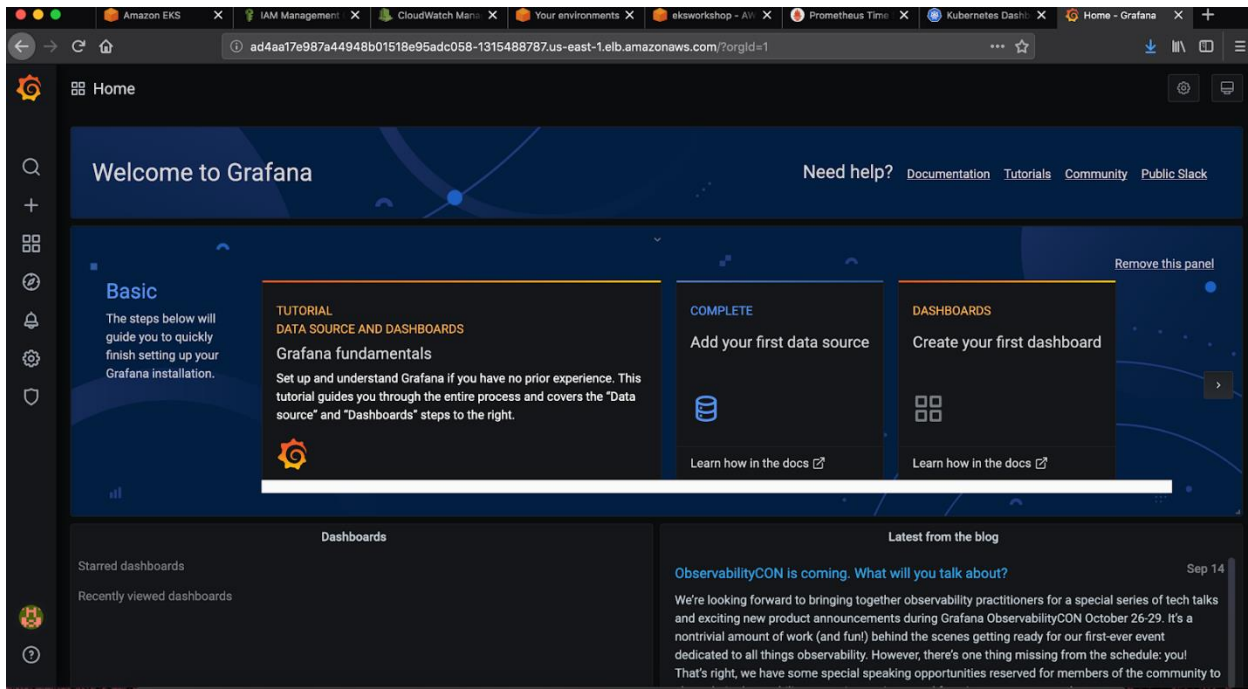
When logging in, use the username admin and get the password hash by running the following:

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



Log in to Grafana

Log in to Grafana dashboard using credentials supplied during configuration.



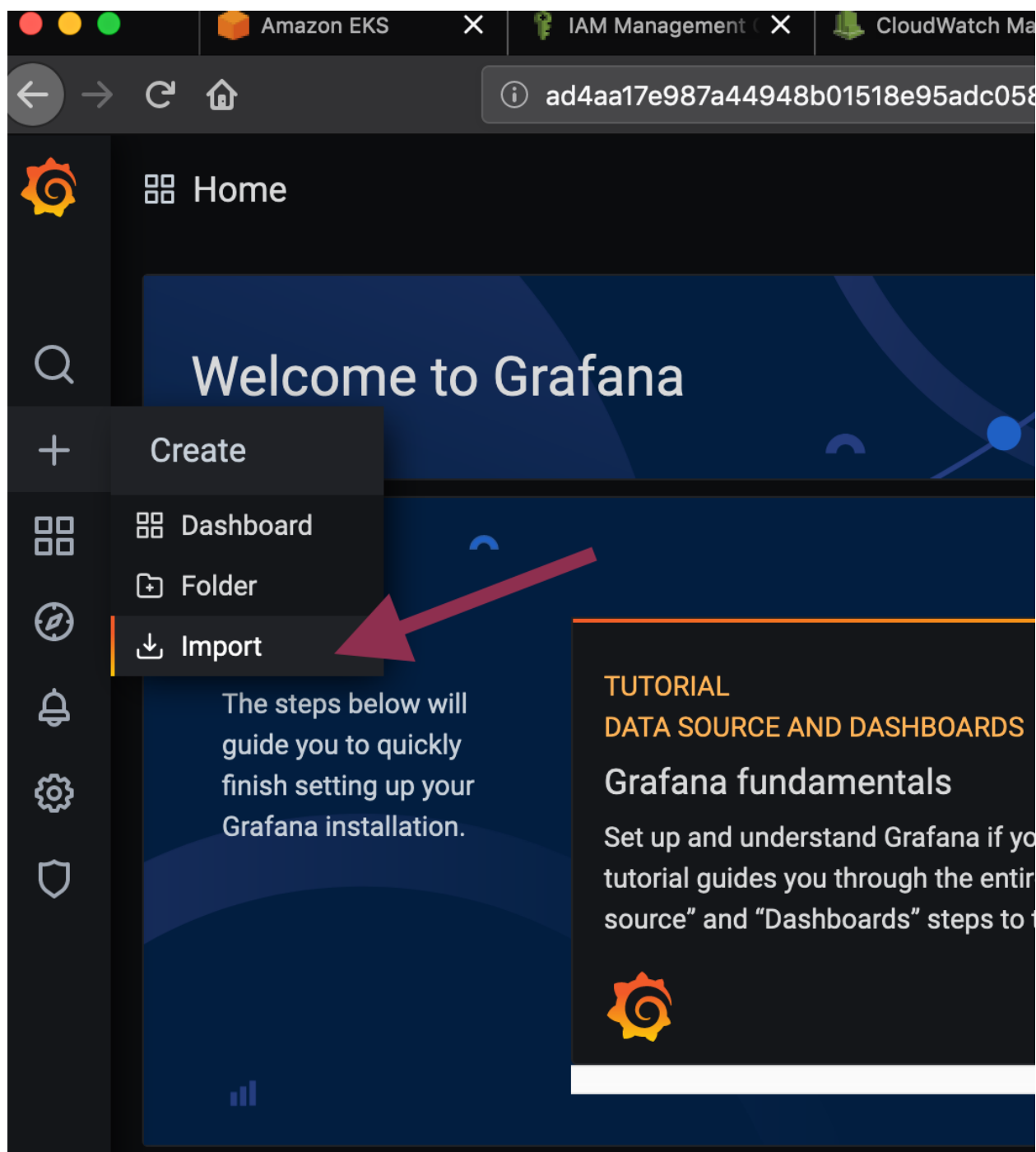
You will notice that 'Install Grafana' & 'create your first data source' are already completed. We will import a community created dashboard for this tutorial.

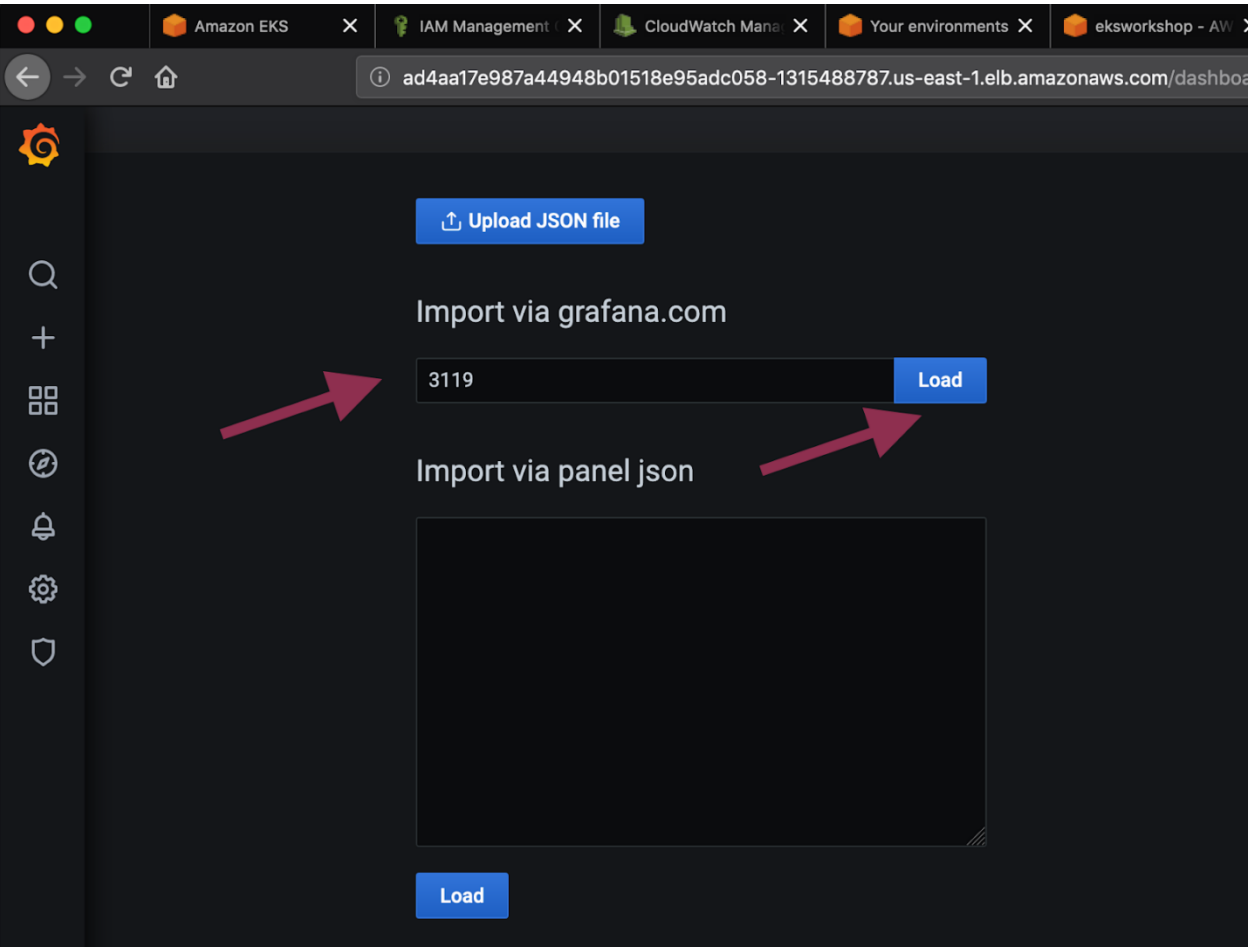
Cluster Monitoring Dashboard

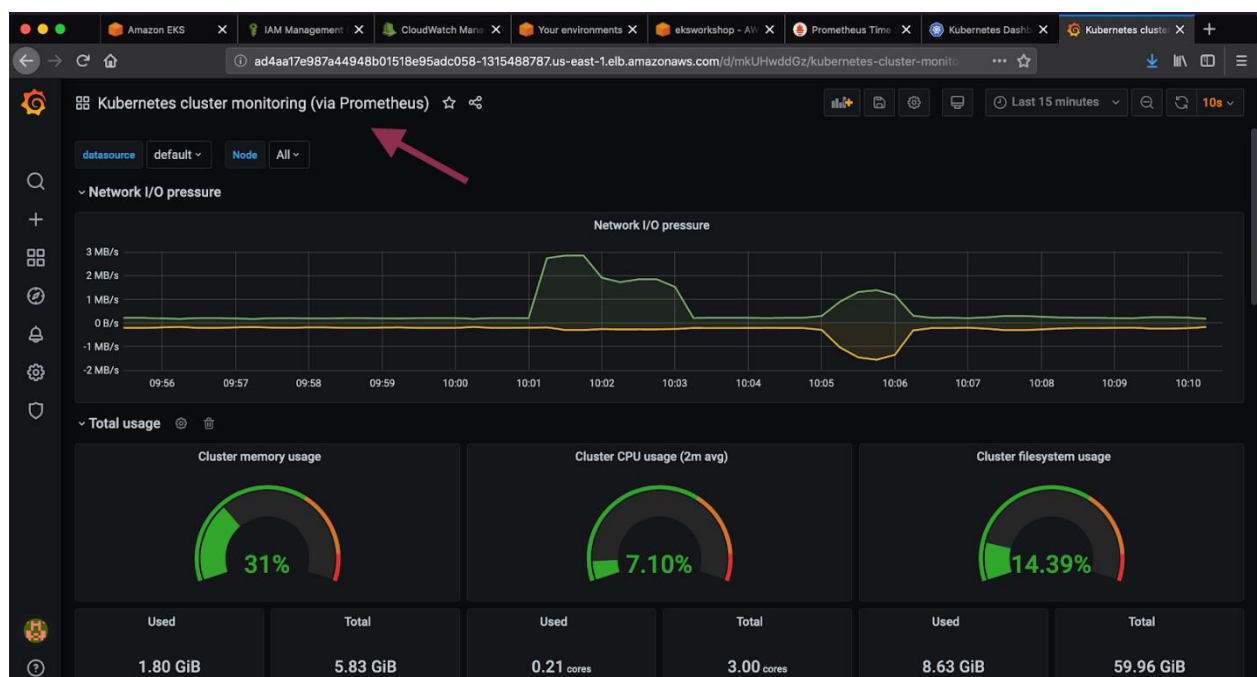
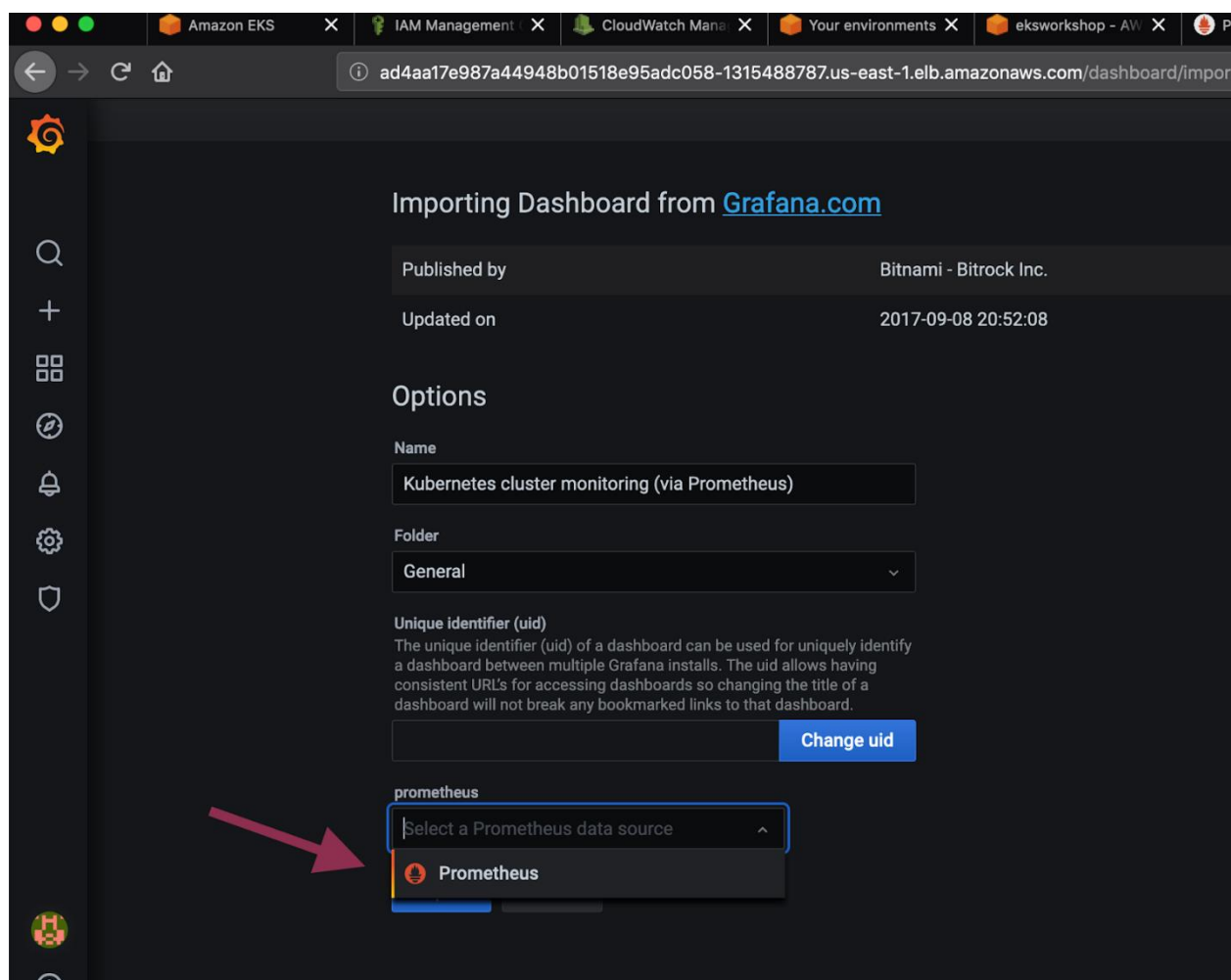
For creating a dashboard to monitor the cluster:

- Click '+' 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



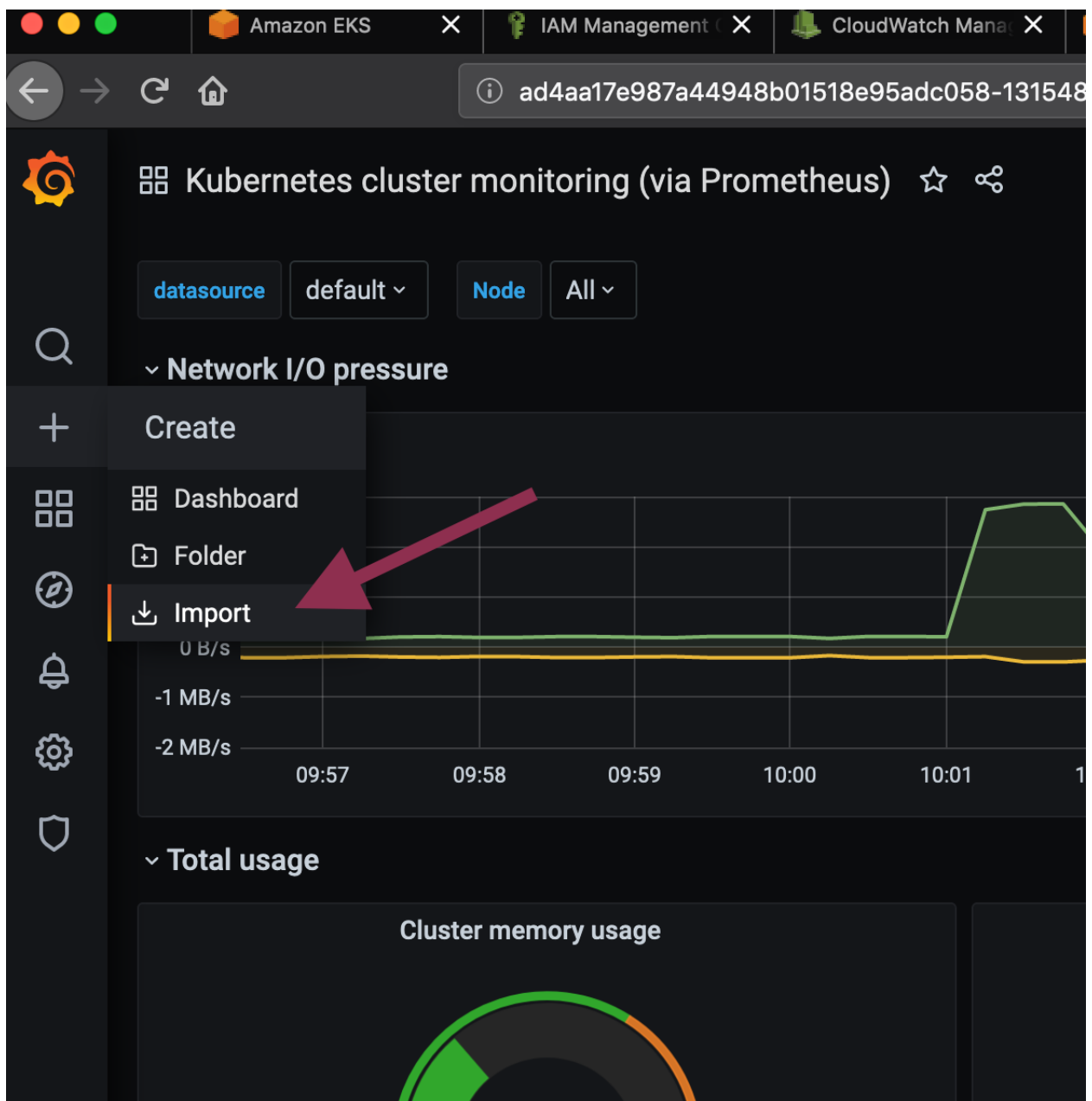


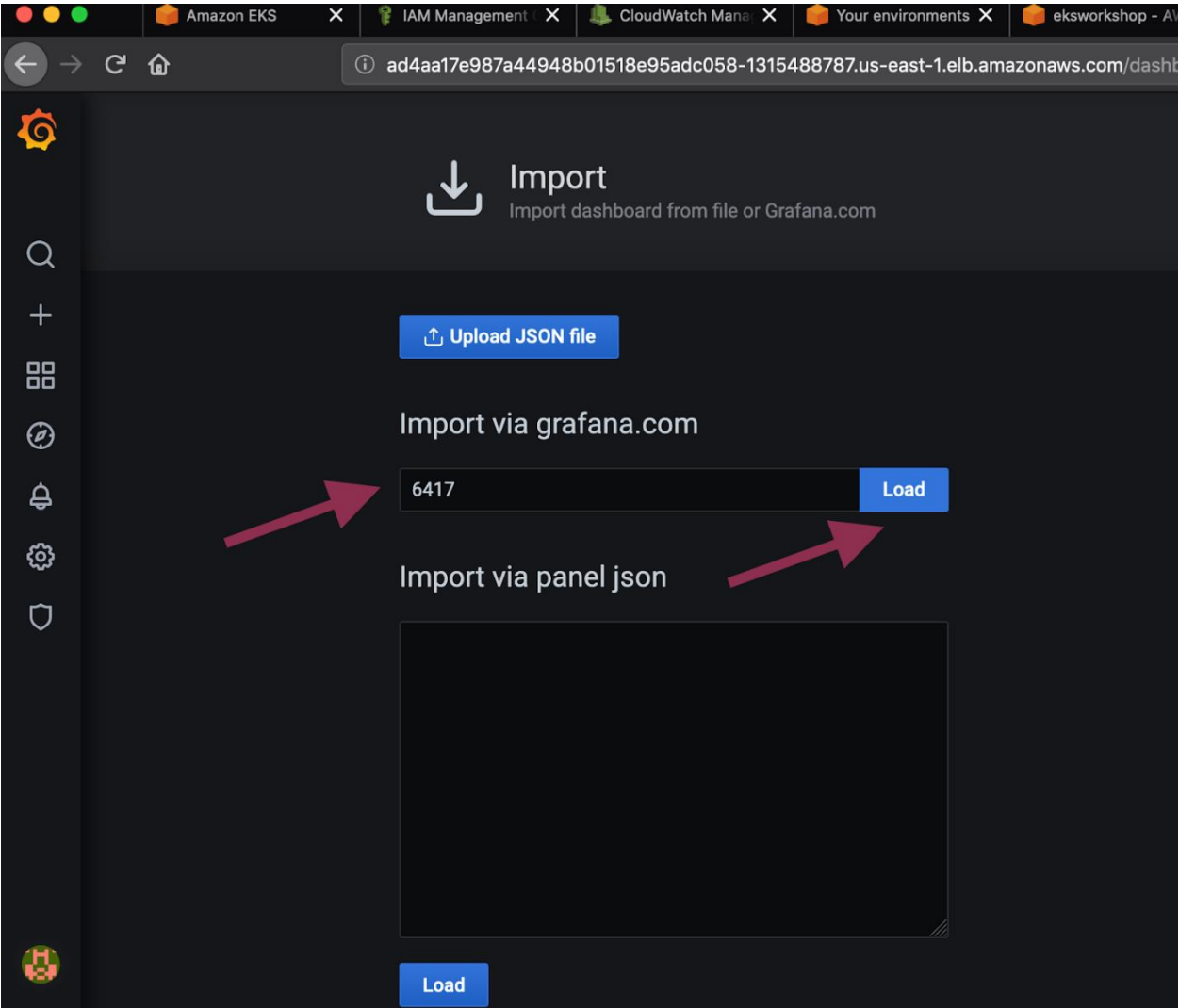


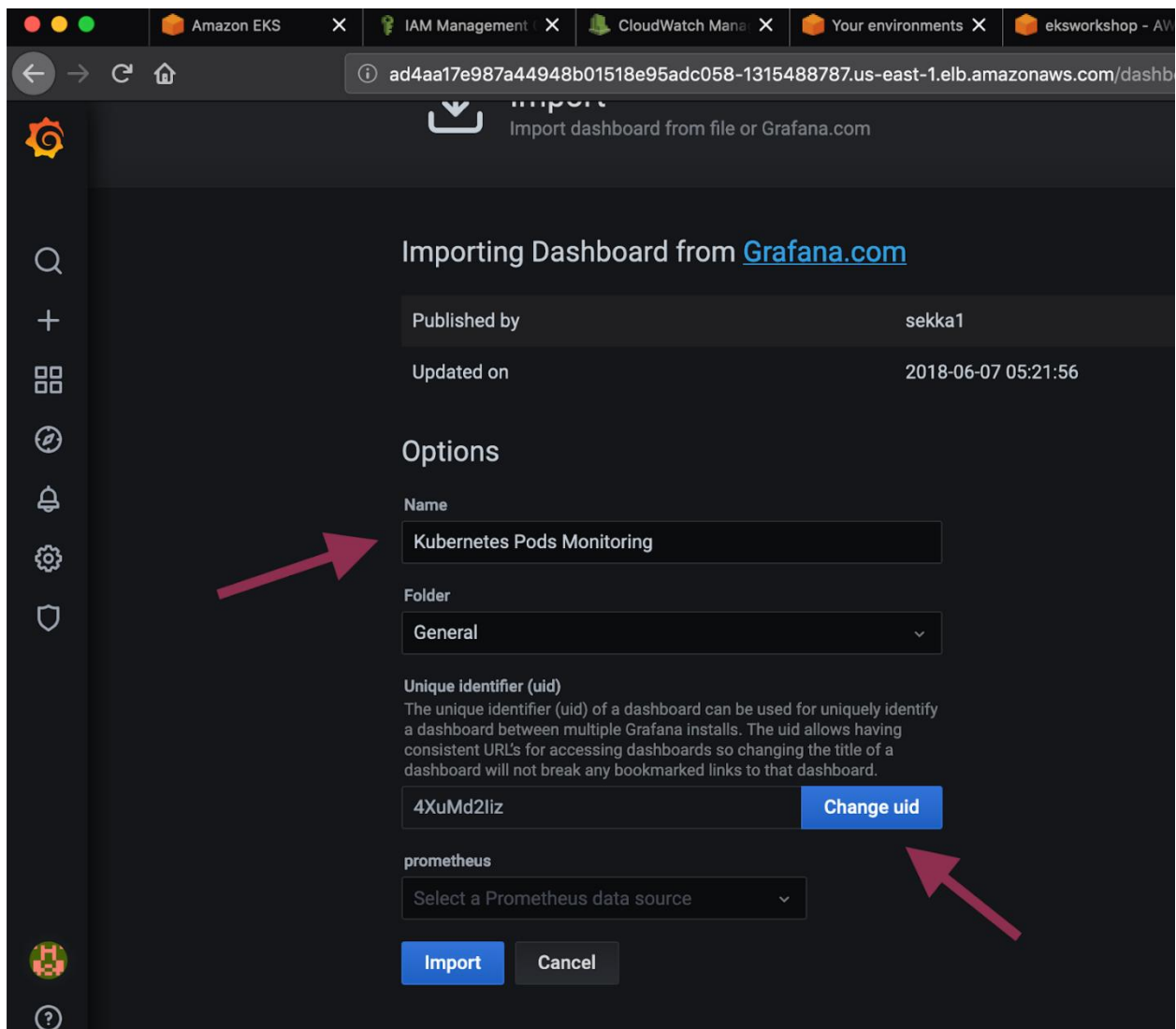
Pods Monitoring Dashboard

For creating a dashboard to monitor all the pods:

- Click '+' button on left panel and select 'Import'.
- Enter 6417 dashboard id under Grafana.com Dashboard.
- Click 'Load'.
- Enter Kubernetes Pods Monitoring as the Dashboard name.
- Click change to set the Unique identifier (uid).
- Select 'Prometheus' as the endpoint under prometheus data sources drop down.s
- Click 'Import'.







Amazon EKS X IAM Management X CloudWatch Mana X Your environments X

ad4aa17e987a44948b01518e95adc058-1315488787.us-east-1.elb.ama

Importing Dashboard from [Grafana.com](#)

Published by sekka1

Updated on 2018-06-07

Options

Name

Kubernetes Pods Monitoring

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 URL's for accessing dashboards so changing the title of a dashboard will not break any bookmarked links to that dashboard.

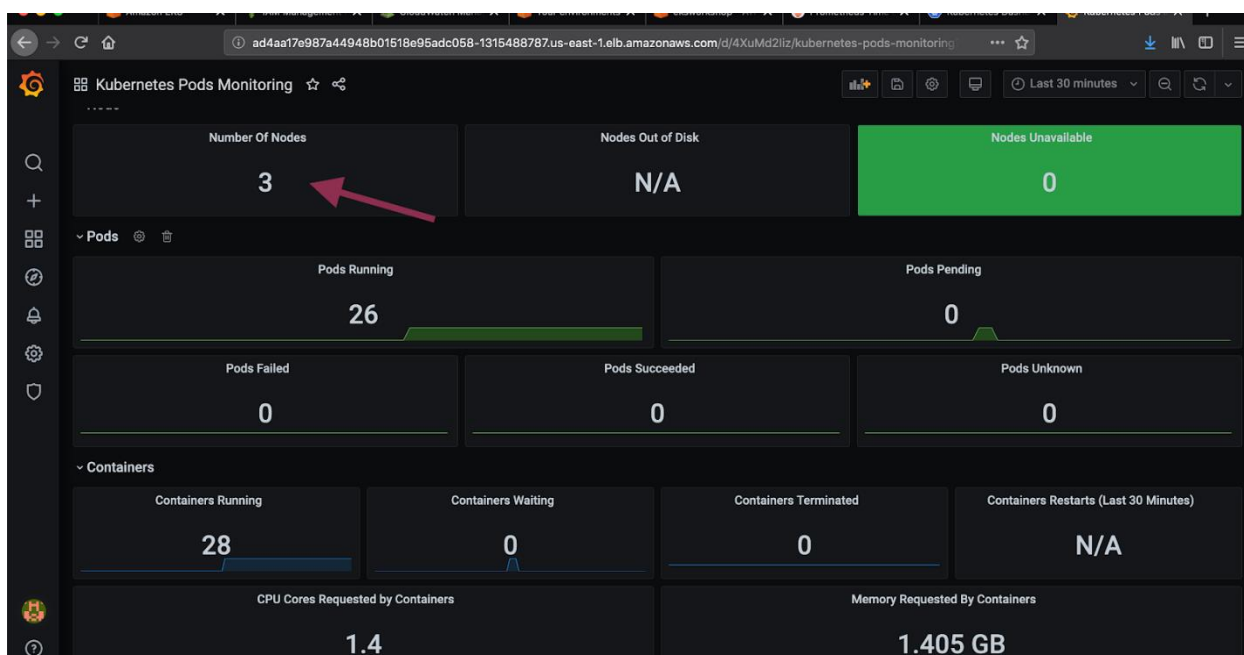
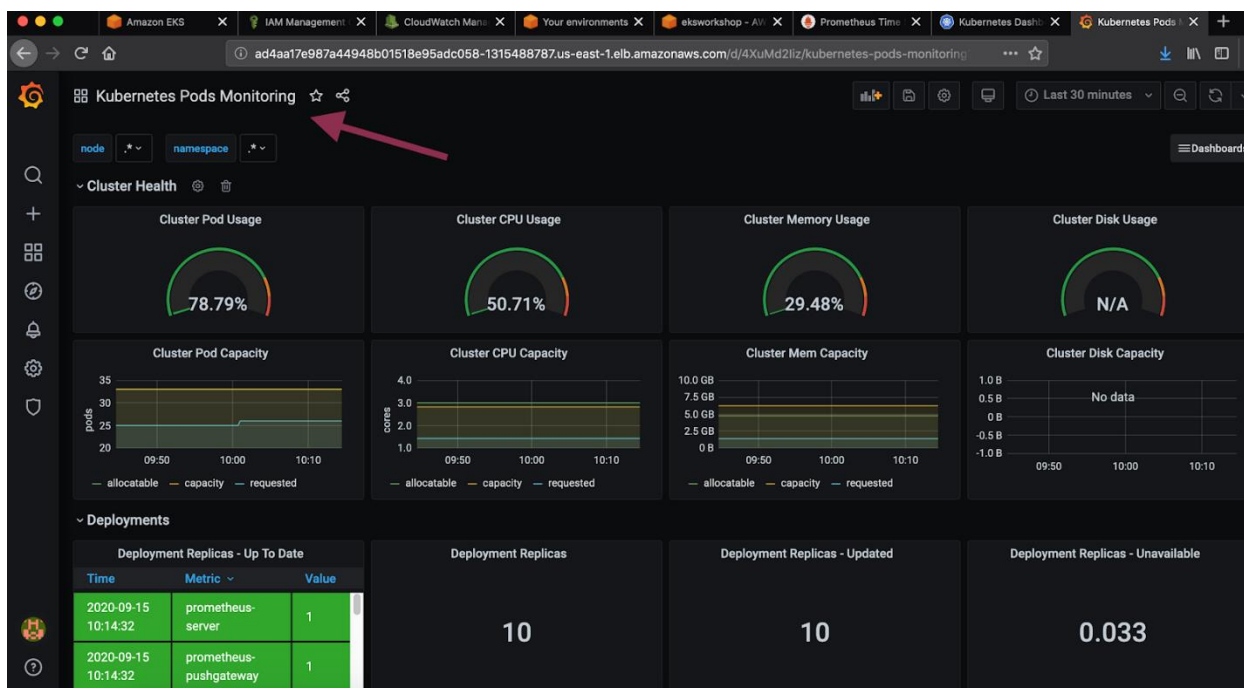
4XuMd2liz

prometheus

Select a Prometheus data source

Prometheus

Continuous Monitoring and DevOps on AWS



End of Lab Exercise. Hope you learnt something useful.