KUBERNETES MODULE 6

Hands-on: 3

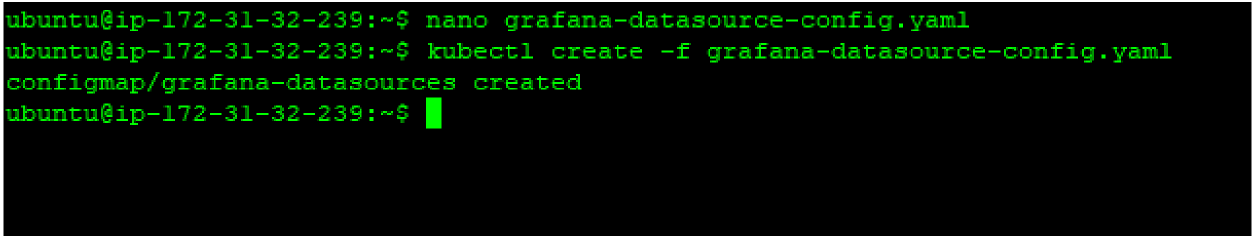
support@intellipaat.com
+91-7022374614
US: 1-800-216-8930(Toll Free)

Connecting Prometheus

Operation 1: Create file named grafana-datasource-config.yaml

```
Nano grafana-datasource-config.yaml
```

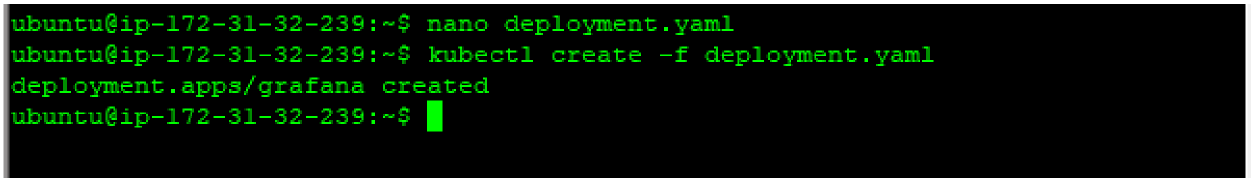
```
apiVersion: v1
kind: ConfigMap
metadata:
name: grafana-datasources
namespace: monitoring
data:
prometheus.yaml: |-
{
"apiVersion": 1,
"datasources": [
{
"access":"proxy",
"editable": true,
"name": "prometheus",
"orgId": 1,
"type": "prometheus",
"url": "http://prometheus-service.monitoring.svc:8080",
"version": 1
kubectl create –f <file name>
]
}
```



```
ubuntu@ip-172-31-32-239:~$ nano grafana-datasource-config.yaml
ubuntu@ip-172-31-32-239:~$ kubectl create -f grafana-datasource-config.yaml
configmap/grafana-datasources created
ubuntu@ip-172-31-32-239:~$
```

Operation 2: Create a file named deployment.yaml

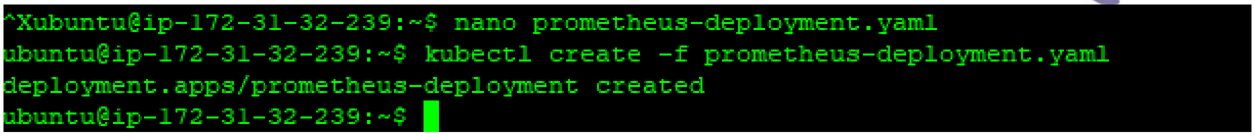
```
apiVersion: apps/v1
kind: Deployment
metadata:
name: grafana
namespace: monitoring
spec:
replicas: 1
selector:
matchLabels:
app: grafana
template:
metadata:
name: grafana
labels:
app: grafana
spec:
containers:
- name: grafana
image: grafana/grafana:latest
ports:
- name: grafana
containerPort: 3000
resources:
limits:
memory: "2Gi"
cpu: "1000m"
requests:
```



```
ubuntu@ip-172-31-32-239:~$ nano deployment.yaml
ubuntu@ip-172-31-32-239:~$ kubectl create -f deployment.yaml
deployment.apps/grafana created
ubuntu@ip-172-31-32-239:~$
```

Operation 4: Create a file named prometheus-deployment.yaml and copy the following contents onto the file. In this configuration, we are mounting the Prometheus config map as a file inside /etc/prometheus. It uses the official Prometheus image from docker hub.

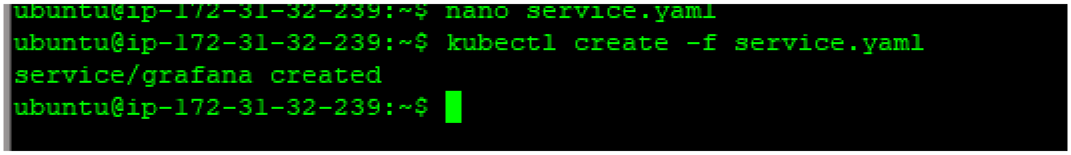
```
apiVersion: apps/v1
kind: Deployment
metadata:
name: prometheus-deployment
namespace: monitoring
spec:
replicas: 1
selector:
matchLabels:
app: prometheus-server
template:
metadata:
labels:
app: prometheus-server
spec:
containers:
- name: prometheus
image: prom/prometheus:v2.12.0
args:
- "--config.file=/etc/prometheus/prometheus.yml"
- "--storage.tsdb.path=/prometheus/"
ports:
```



```
^Xubuntu@ip-172-31-32-239:~$ nano prometheus-deployment.yaml
ubuntu@ip-172-31-32-239:~$ kubectl create -f prometheus-deployment.yaml
deployment.apps/prometheus-deployment created
ubuntu@ip-172-31-32-239:~$
```

Operation 5: To access the Prometheus dashboard over a IP or a DNS name, you need to expose it as Kubernetes service. So let's create a service

```
apiVersion: v1
kind: Service
metadata:
name: grafana
namespace: monitoring
annotations:
prometheus.io/scrape: 'true'
prometheus.io/port:   '3000'
spec:
selector:
app: grafana
type: NodePort
ports:
- port: 3000
targetPort: 3000
nodePort: 32000
```



```
ubuntu@ip-172-31-32-239:~$ nano service.yaml
ubuntu@ip-172-31-32-239:~$ kubectl create -f service.yaml
service/grafana created
ubuntu@ip-172-31-32-239:~$
```

Check by going to ip and port