KUBERNETES MODULE 6

Hands-on: 2

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

Connecting Prometheus

Operation 1: Create a namespace

```
kubectl create namespace <namespace name>
```

```
ubuntu@ip-172-31-32-239:~$ kubectl create namespace monitoring
namespace/monitoring created
ubuntu@ip-172-31-32-239:~$
```

Operation 2: Create a file named clusterRole.yaml and copy the content for this file given below

```
apiVersion: rbac.authorization.k8s.io/v1beta1
kind: ClusterRole
metadata:
name: prometheus
rules:
- apiGroups: [""]
resources:
- nodes
- nodes/proxy
- services
- endpoints
- pods
verbs: ["get", "list", "watch"]
- apiGroups:
- extensions
resources:
- ingresses
verbs: ["get", "list", "watch"]
- nonResourceURLs: ["/metrics"]
verbs: ["get"]
---
apiVersion: rbac.authorization.k8s.io/v1beta1
```

kubectl create –f <file name>

```
ubuntu@ip-172-31-32-239:~$ nano clusterRole.yaml
ubuntu@ip-172-31-32-239:~$ kubectl create -f clusterRole.yaml
clusterrole.rbac.authorization.k8s.io/prometheus created
clusterrolebinding.rbac.authorization.k8s.io/prometheus created
ubuntu@ip-172-31-32-239:~$
```

Operation 3: Execute the following command to create the config map in Kubernetes

```
apiVersion: v1
kind: ConfigMap
metadata:
name: prometheus-server-conf
labels:
name: prometheus-server-conf
namespace: monitoring
data:
prometheus.rules: |-
groups:
- name: devopscube demo alert
rules:
- alert: High Pod Memory
expr: sum(container_memory_usage_bytes) > 1
for: 1m
labels:
severity: slack
annotations:
summary: High Memory Usage
prometheus.yml: |-
global:
scrape_interval: 5s
evaluation_interval: 5s
```

```
ubuntu@ip-172-31-32-239:~$ kubectl create -f config-map.yaml
configmap/prometheus-server-conf 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"
```

```
^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: prometheus-service
namespace: monitoring
annotations:
prometheus.io/scrape: 'true'
prometheus.io/port:   '9090'
spec:
selector:
app: prometheus-server
type: NodePort
ports:
- port: 8080
targetPort: 9090
nodePort: 30000
```

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

```
ubuntu@ip-172-31-32-239:~$ kubectl get svc -n=monitoring
NAME                 TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)          AG
prometheus-service   NodePort    10.96.115.61    <none>        8080:30000/TCP   87
```

Check by going to ip and port