Create a Namespace & ClusterRole

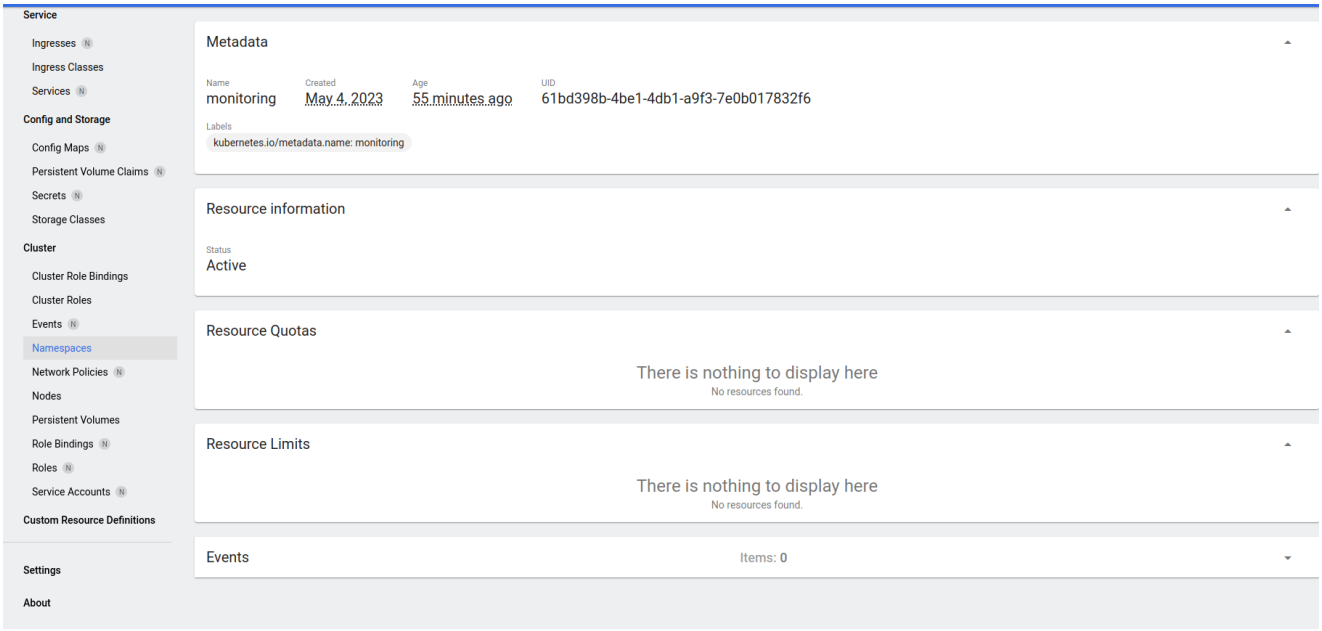
First we need to created namespace for our monitoring using the following command, after we start the minikube

```
minikube start
```

create new namespace using

```
kubectl create namespace monitoring
```

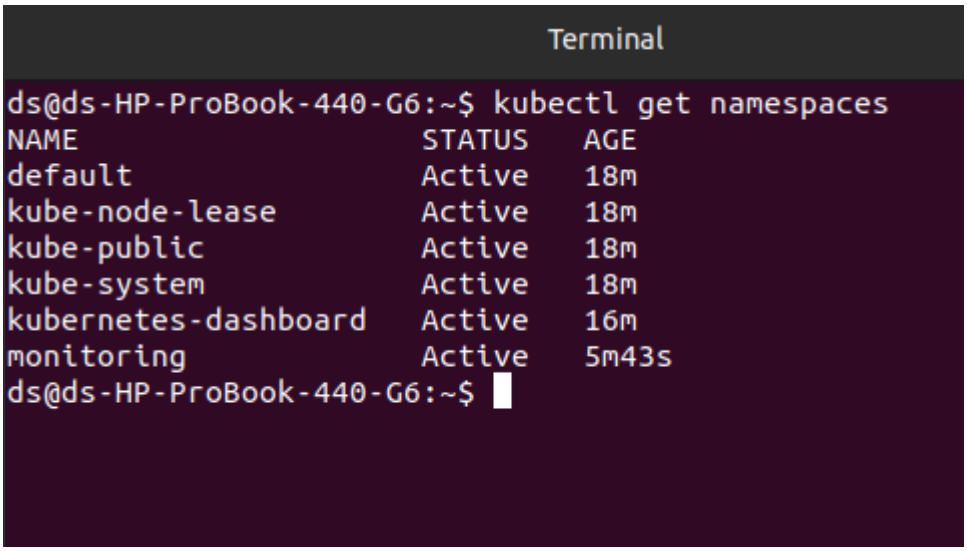
To start the dashboard for kubernetis cluster run **minikube dashboard** and navigate to **Cluster / Namespaces** and we can see that the namespace that we create.



PROF

To verify that namespace **monitoring** is created in command line type:

kubectl get namespaces



Now lets create new file caled **clusterRole.yaml**
Note: In the role, given below, you can see that we have added ***get, list, and watch*** permissions to nodes, services endpoints, pods, and ingresses. The role binding is bound to the monitoring namespace. If you have any use case to retrieve metrics from any other object, you need to add that in this cluster role.

clusterRole.yaml

PROF

clusterRole.yaml

monitoring > clusterRole.yaml > [ ] subjects > {} 0 > namespace

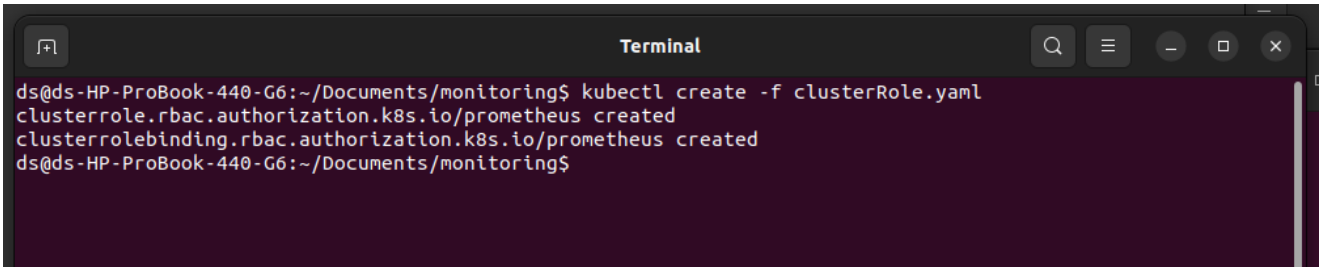
io.k8s.api.rbac.v1.ClusterRoleBinding (v1@clusterrolebinding.json) | io.k8s.api.rbac.v1.ClusterRole (v1@clusterrole.json)

```yaml
apiVersion: rbac.authorization.k8s.io/v1
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/v1
kind: ClusterRoleBinding
metadata:
  name: prometheus
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: prometheus
subjects:
- kind: ServiceAccount
  name: default
  namespace: monitoring
```

run the following command to create the role

PROF

```
kubectl create -f clusterRole.yaml
```



Create a Config Map To Externalize Prometheus Configurations

All configurations for Prometheus are part of prometheus.yaml file and all the alert rules for Alertmanager are configured in prometheus.rules.

prometheus.yaml: This is the main Prometheus configuration which holds all the scrape configs, service discovery details, storage locations, data retention configs, etc)

prometheus.rules: This file contains all the Prometheus alerting rules

Lets create new file config-map.yaml with the following content config-map.yaml

```
kubectl create -f config-map.yaml
```