Kubernetes Dashboard

Introduction:

So far, we have been using the Kubernetes Cluster in CLI mode and kubectl command was helping us to create, delete or update the resources.

We can also use WebUI to access our Kubernetes cluster. So, creating and modifying the resources can also be done using the WebUI.

Objective:

Create a Kubernetes WebUI

Create a Kubernetes WebUI:

Use the below GitHub link to download the file.

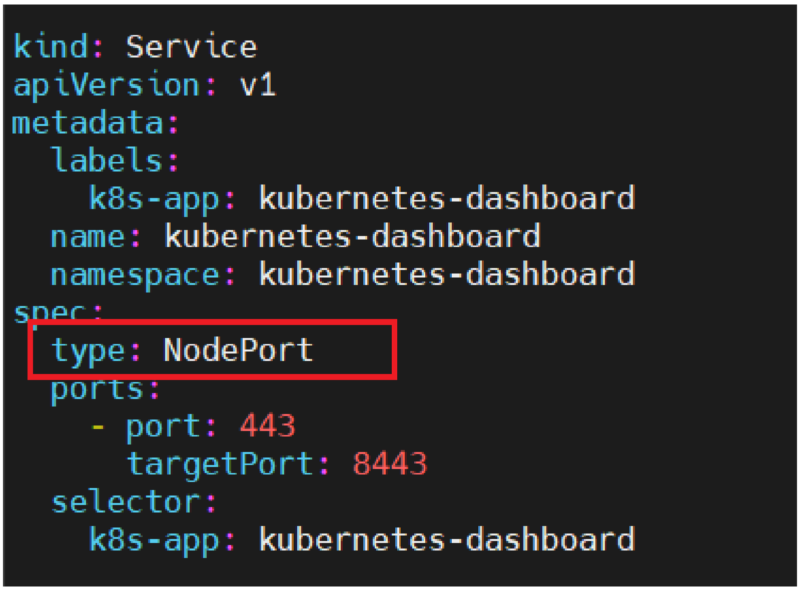
https://github.com/kubernetes/dashboard

We can download the above file first and will make some changes as we would be using **NodePort** service. So use the below command to download the file.

```
wget
https://raw.githubusercontent.com/kubernetes/dashboard/v2.7.0/aio/deploy/recommended.yaml
```

Open the downloaded file and change the service type to **NodePort** as seen in below given picture.

```
vi recommended.yaml
```

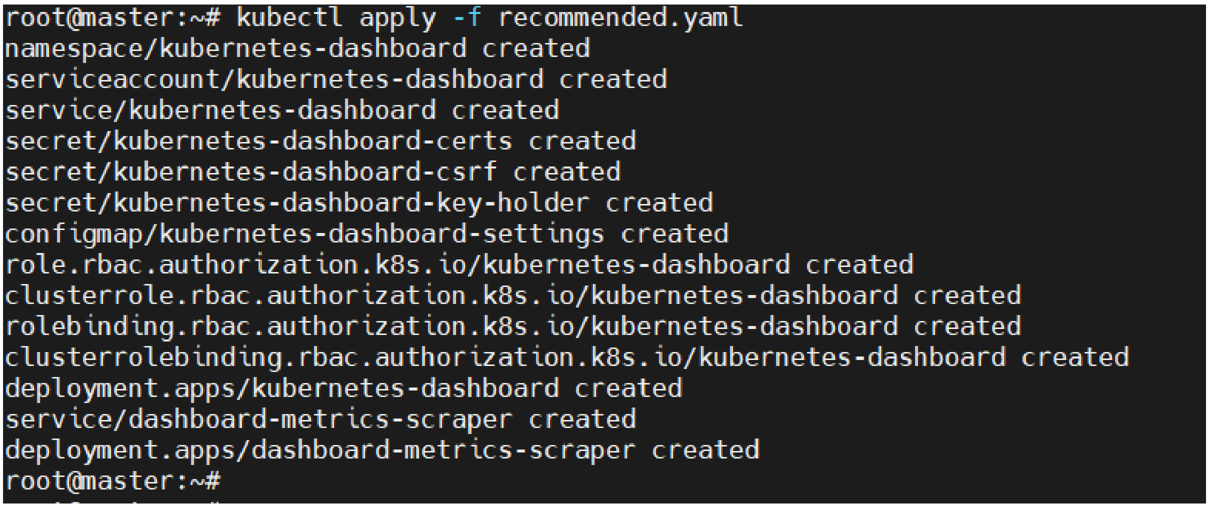


```
kind: Service
apiVersion: v1
metadata:
  labels:
    k8s-app: kubernetes-dashboard
  name: kubernetes-dashboard
  namespace: kubernetes-dashboard
spec:
  type: NodePort
  ports:
    - port: 443
      targetPort: 8443
  selector:
    k8s-app: kubernetes-dashboard
```

Now apply the changes and it will create the below resources.

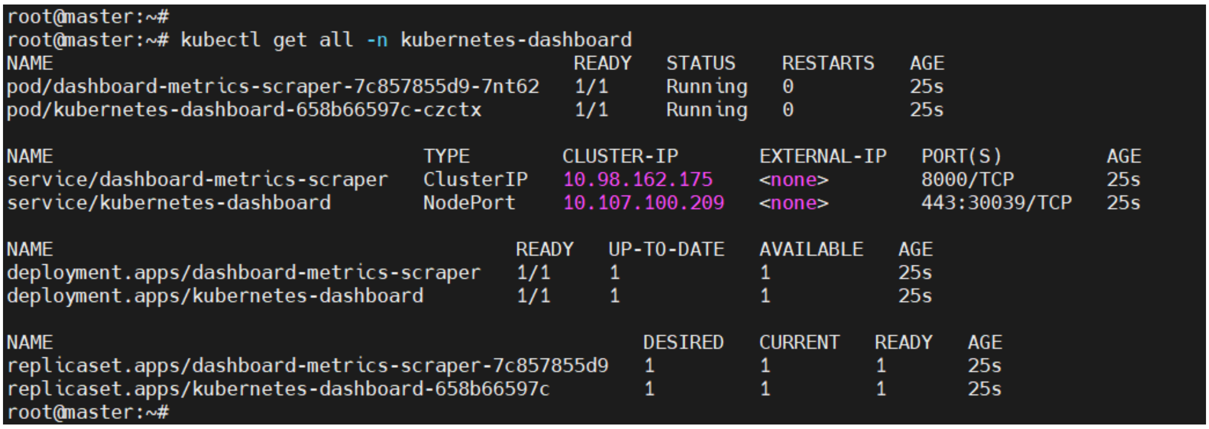
```
kubectl apply -f recommended.yaml
```

Resource Type	Resource Name
Namespace	kubernetes-dashboard
ServiceAccount	kubernetes-dashboard
Service	kubernetes-dashboard
Secret	kubernetes-dashboard-certs
Secret	kubernetes-dashboard-csrf
Secret	kubernetes-dashboard-key-holder
ConfigMap	kubernetes-dashboard-settings
Role	kubernetes-dashboard
ClusterRole	kubernetes-dashboard
RoleBinding	kubernetes-dashboard
ClusterRoleBinding	kubernetes-dashboard
Deployment	kubernetes-dashboard
Service	dashboard-metrics-scraper
Deployment	dashboard-metrics-scraper



```
root@master:~# kubectl apply -f recommended.yaml
namespace/kubernetes-dashboard created
serviceaccount/kubernetes-dashboard created
service/kubernetes-dashboard created
secret/kubernetes-dashboard-certs created
secret/kubernetes-dashboard-csrf created
secret/kubernetes-dashboard-key-holder created
configmap/kubernetes-dashboard-settings created
role.rbac.authorization.k8s.io/kubernetes-dashboard created
clusterrole.rbac.authorization.k8s.io/kubernetes-dashboard created
rolebinding.rbac.authorization.k8s.io/kubernetes-dashboard created
clusterrolebinding.rbac.authorization.k8s.io/kubernetes-dashboard created
deployment.apps/kubernetes-dashboard created
service/dashboard-metrics-scraper created
deployment.apps/dashboard-metrics-scraper created
root@master:~#
```

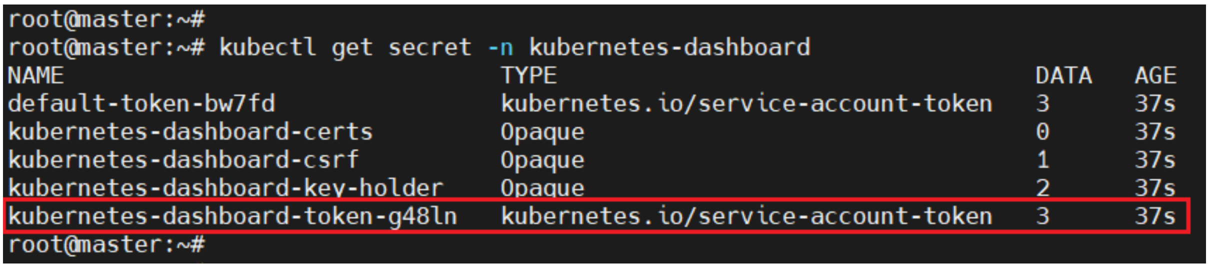
Below we can see, we have deployments, pods, services and replica sets in **Kubernetes-dashboard** namespace.



```
root@master:~#
root@master:~# kubectl get all -n kubernetes-dashboard
NAME                                             READY   STATUS    RESTARTS   AGE
pod/dashboard-metrics-scraper-7c857855d9-7nt62   1/1     Running   0          25s
pod/kubernetes-dashboard-658b66597c-czctx        1/1     Running   0          25s

NAME                                TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)         AGE
service/dashboard-metrics-scraper   ClusterIP   10.98.162.175    <none>        8000/TCP        25s
service/kubernetes-dashboard        NodePort    10.107.100.209   <none>        443:30039/TCP   25s

NAME                                        READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/dashboard-metrics-scraper   1/1     1            1           25s
deployment.apps/kubernetes-dashboard        1/1     1            1           25s

NAME                                                   DESIRED   CURRENT   READY   AGE
replicaset.apps/dashboard-metrics-scraper-7c857855d9   1         1         1       25s
replicaset.apps/kubernetes-dashboard-658b66597c        1         1         1       25s
root@master:~#
```

Below we can see that we have all the secrets created which were mentioned in our definition file.

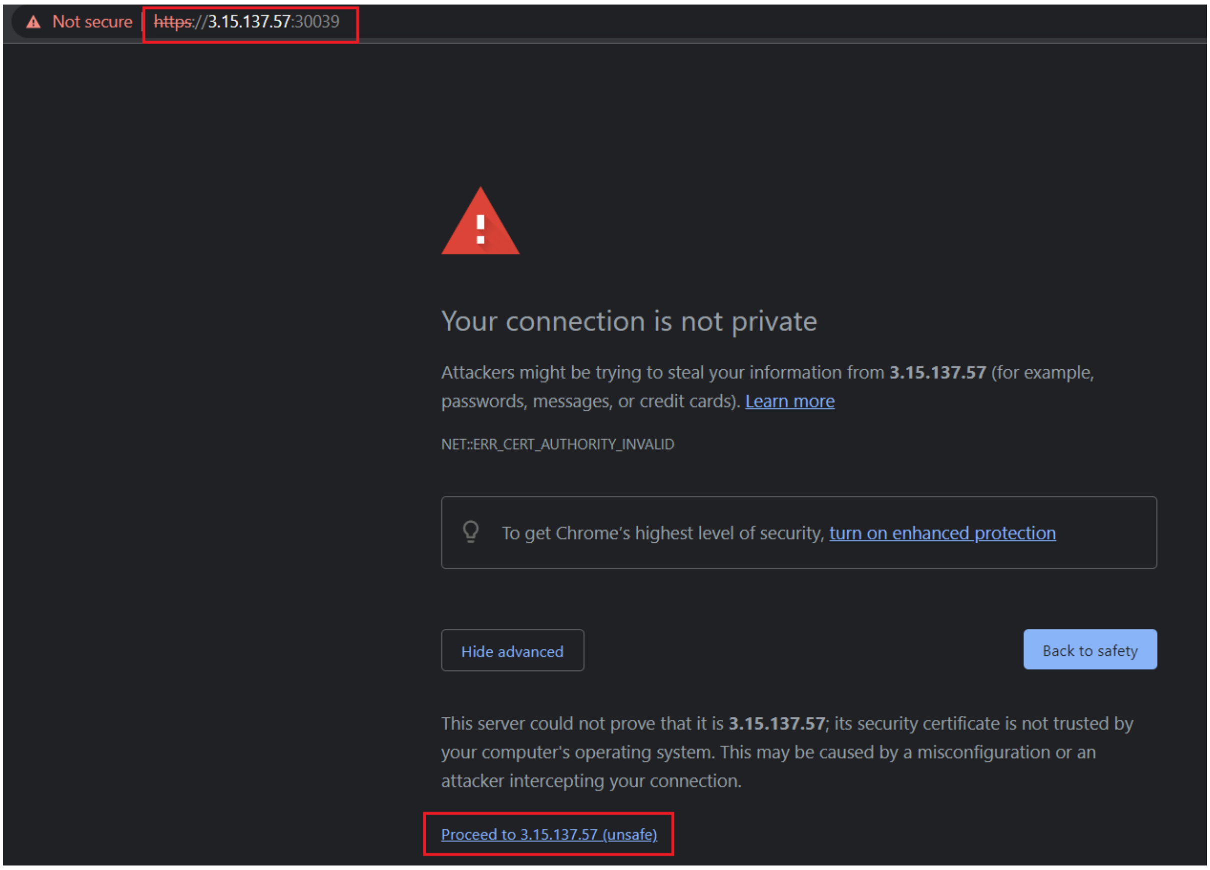
We had created a service account **kubernetes-dashboard** which will be using **kubernetes-dashboard-token-g48ln** secret.



```
root@master:~#
root@master:~# kubectl get secret -n kubernetes-dashboard
NAME                               TYPE                                  DATA   AGE
default-token-bw7fd                kubernetes.io/service-account-token   3      37s
kubernetes-dashboard-certs         Opaque                                0      37s
kubernetes-dashboard-csrf          Opaque                                1      37s
kubernetes-dashboard-key-holder    Opaque                                2      37s
kubernetes-dashboard-token-g48ln   kubernetes.io/service-account-token   3      37s
root@master:~#
```

Now let’s try to access our Cluster using our NodePort service.

Open the browse and type https://node-ip:nodeport (replace node-ip with the IP address of your node)



Your connection is not private

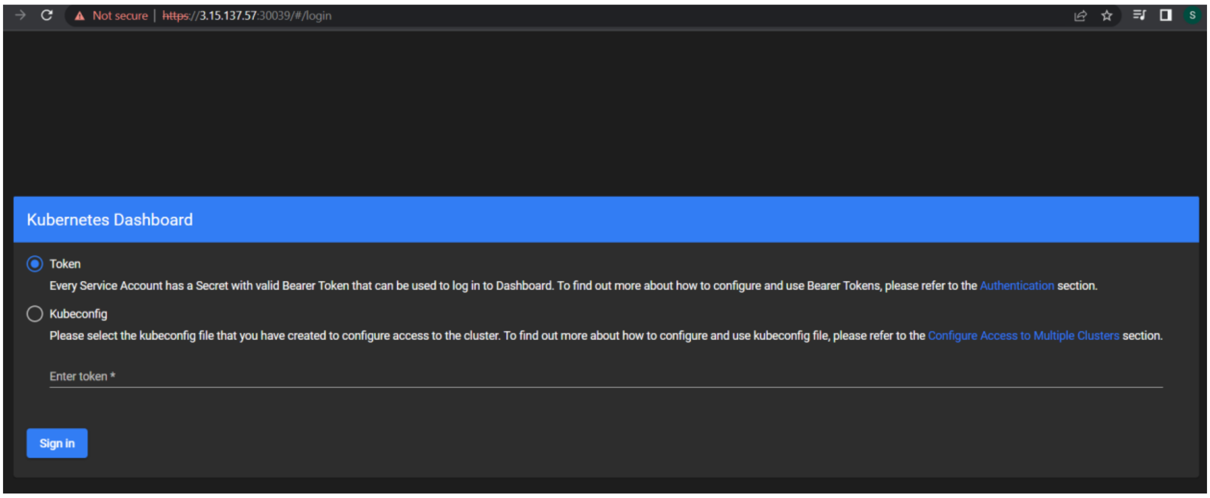
Attackers might be trying to steal your information from **3.15.137.57** (for example, passwords, messages, or credit cards). Learn more

NET::ERR_CERT_AUTHORITY_INVALID

To get Chrome's highest level of security, turn on enhanced protection

This server could not prove that it is **3.15.137.57**; its security certificate is not trusted by your computer's operating system. This may be caused by a misconfiguration or an attacker intercepting your connection.

Click the proceed button.



● Token

Every Service Account has a Secret with valid Bearer Token that can be used to log in to Dashboard. To find out more about how to configure and use Bearer Tokens, please refer to the Authentication section.

○ Kubeconfig

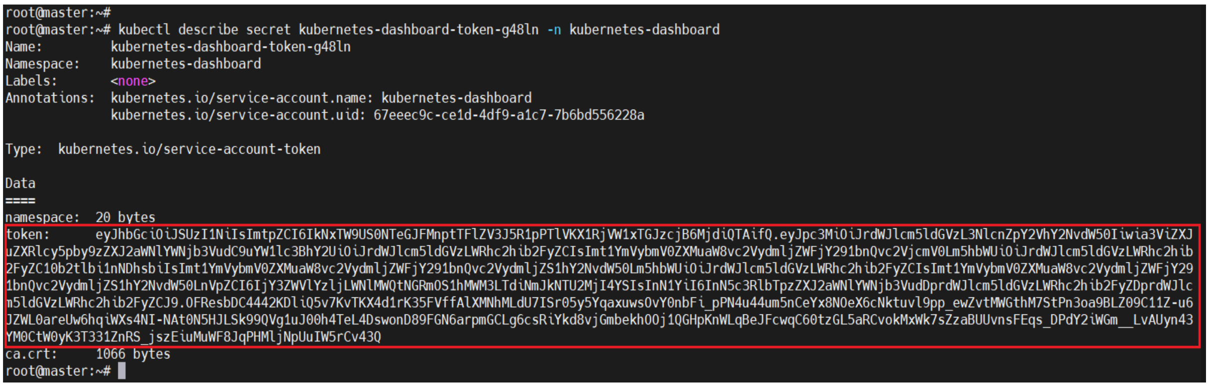
Please select the kubeconfig file that you have created to configure access to the cluster. To find out more about how to configure and use kubeconfig file, please refer to the Configure Access to Multiple Clusters section.

Enter token *

Now it is asking for the Token or KubeConfig file.

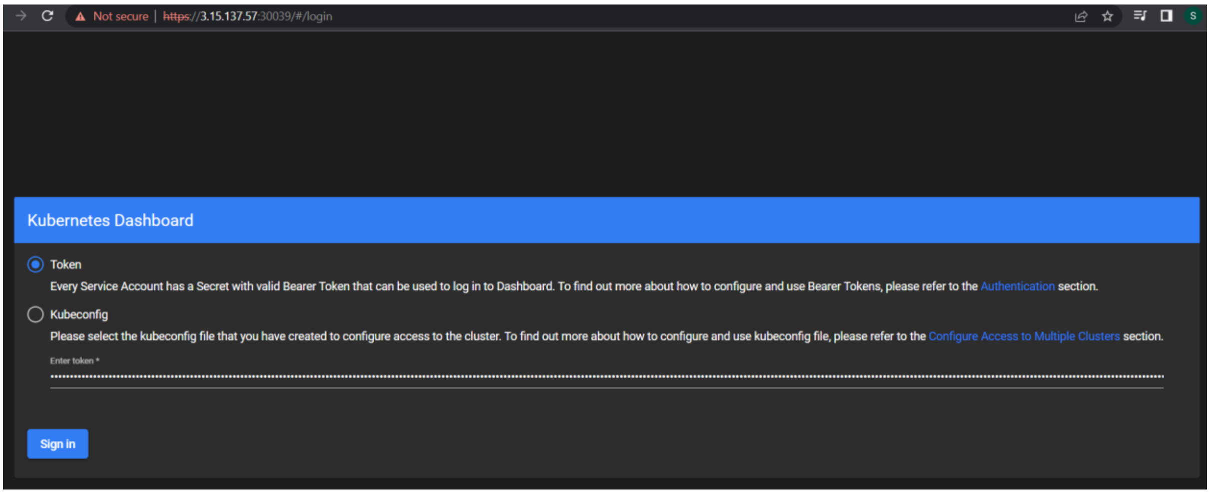
We have our token with us, let's get the token using below command.

```
kubectl describe secret kubernetes-dashboard-token-g48ln -n kubernetes-dashboard
```



```
root@master:~#
root@master:~# kubectl describe secret kubernetes-dashboard-token-g48ln -n kubernetes-dashboard
Name:         kubernetes-dashboard-token-g48ln
Namespace:    kubernetes-dashboard
Labels:       <none>
Annotations:  kubernetes.io/service-account.name: kubernetes-dashboard
              kubernetes.io/service-account.uid: 67eeec9c-ce1d-4df9-a1c7-7b6bd556228a

Type:  kubernetes.io/service-account-token

Data
====
namespace:  20 bytes
token:      eyJhbGciOiJSUzI1NiIsImtpZCI6IkNxTW9US0NTeGJFMnptTFlZV3J5R1pPTlVKX1RjVW1xTGJzcjB6MjdiQTAifQ.eyJpc3MiOiJrdWJlcm5ldGVzL3NlcnZpY2VhY2NvdW50Iiwia3ViZXJ
uZXRlcy5pby9zZXJ2aWNlYWNjb3VudC9uYW1lc3BhY2UiOiJrdWJlcm5ldGVzLWRhc2hib2FyZCIsImt1YmVybmV0ZXMuaW8vc2VydmljZWFjY291bnQvc2VjcmV0Lm5hbWUiOiJrdWJlcm5ldGVzLWRhc2hib
2FyZC10b2tlbi1nNDhsbiIsImt1YmVybmV0ZXMuaW8vc2VydmljZWFjY291bnQvc2VydmljZS1hY2NvdW50Lm5hbWUiOiJrdWJlcm5ldGVzLWRhc2hib2FyZCIsImt1YmVybmV0ZXMuaW8vc2VydmljZWFjY29
1bnQvc2VydmljZS1hY2NvdW50LnVpZCI6IjY3ZWVlYzljLWNlMWQtNGRmOS1hMWM3LTdiNmJkNTU2MjI4YSIsInN1YiI6InN5c3RlbTpzZXJ2aWNlYWNjb3VudDprdWJlcm5ldGVzLWRhc2hib2FyZDprdWJlc
m5ldGVzLWRhc2hib2FyZCJ9.0FResbDC4442KDliQ5v7KvTKX4d1rK35FVffAlXMNhMLdU7ISr05y5YqaxuwsOvY0nbFi_pPN4u44um5nCeYx8NOeX6cNktuvl9pp_ewZvtMWGthM7StPn3oa9BLZ09C11Z-u6
JZWL0areUw6hqiWXs4NI-NAt0N5HJLSk99QVg1uJ00h4TeL4DswonD89FGN6arpmGCLg6csRiYkd8vjGmbekhOOj1QGHpKnWLqBeJFcwqC60tzGL5aRCvokMxWk7sZzaBUUvnsFEqs_DPdY2iWGm__LvAUyn43
YM0CtW0yK3T331ZnRS_jszEiuMuWF8JqPHMljNpUuIW5rCv43Q
ca.crt:     1066 bytes
root@master:~#
```

Now copy the above token and paste in the token field.



Kubernetes Dashboard

● Token

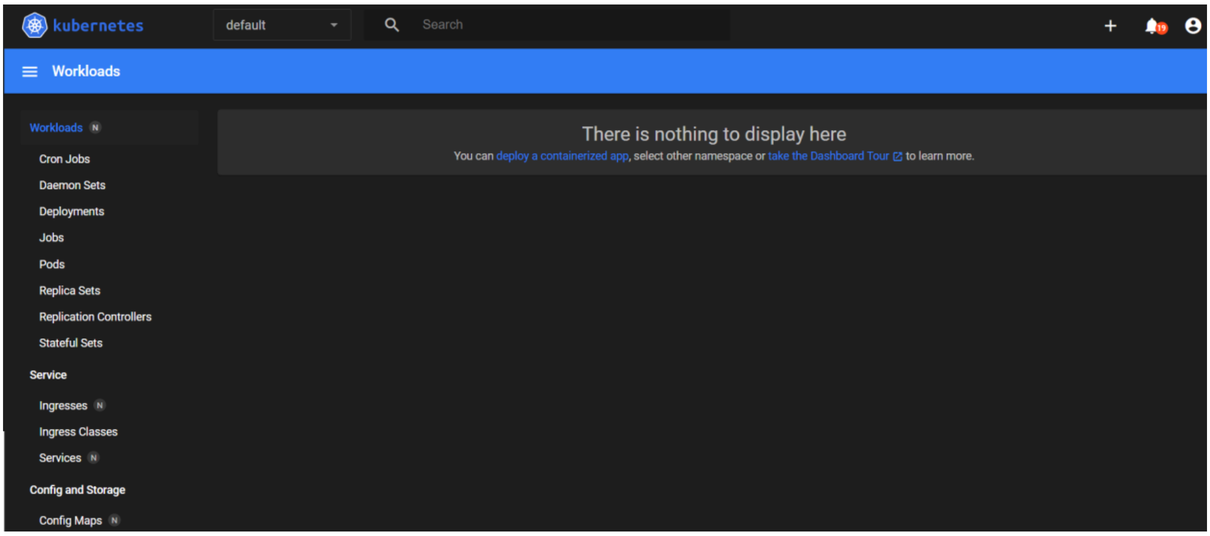
Every Service Account has a Secret with valid Bearer Token that can be used to log in to Dashboard. To find out more about how to configure and use Bearer Tokens, please refer to the Authentication section.

○ Kubeconfig

Please select the kubeconfig file that you have created to configure access to the cluster. To find out more about how to configure and use kubeconfig file, please refer to the Configure Access to Multiple Clusters section.

Enter token * ••

Now press **Sign in.**



There is nothing to display here

You can deploy a containerized app, select other namespace or take the Dashboard Tour to learn more.

Now our WebUI is ready.