Kubernetes Dashboard

- What is Kubernetes Dashboard
- Install Kubernetes Dashboard
- Access Kubernetes Dashboard
- Create An Authentication Token (RBAC)

Kubernetes Dashboard

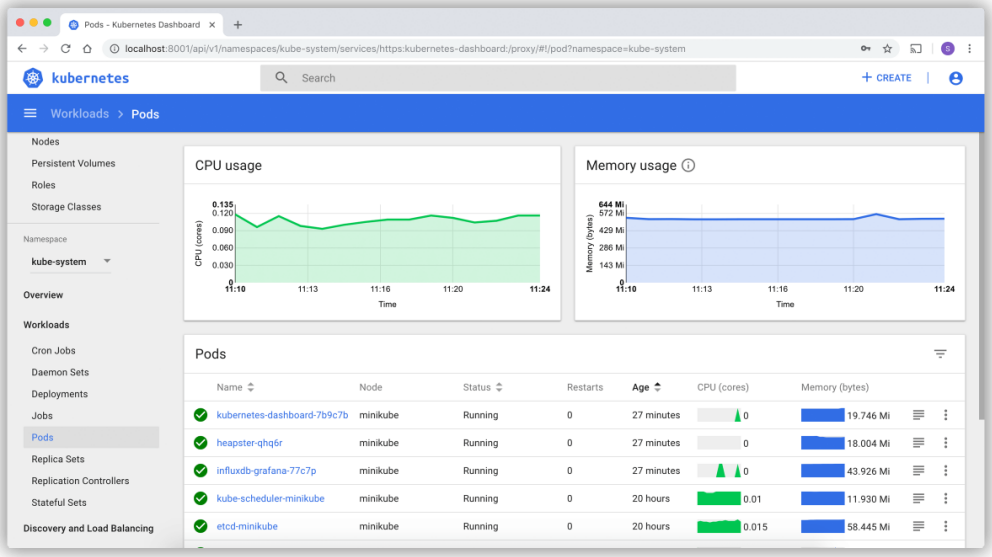
Kubernetes dashboard is the official web-based UI for Kubernetes.

We can use Dashboard to get an overview of applications running on our cluster, as well as for creating or modifying individual Kubernetes resources (such as Deployments, Jobs, DaemonSets etc).

We can scale a Deployment, initiate a rolling update, restart a pod or deploy new applications using a deploy wizard.

Dashboard also provides information on the state of Kubernetes resources in our cluster and on any errors that may have occurred.:

- Overview of the Kubernetes cluster
- Deploy applications onto your Kubernetes clusters
- Get an overview of the applications running
- Troubleshoot those applications
- Get an overview of the resources running in the cluster
- Create, modify, update and delete Kubernetes resources
- Information about the state of the Kubernetes resources running in the cluster
- Basic resource metrics including resource usage for individual Kubernetes objects



For Minikube:

minikube addons enable dashboard

minikube dashboard

Docker Desktop:

Step1: Install Kubernetes Dashboard

The Dashboard UI is not deployed by default. To deploy it, run the following command:

```
kubectl apply -f
https://raw.githubusercontent.com/kubernetes/dashboard/v2.7.0/aio/deploy/recommended.yaml
```

Note: get the latest URL from the link: https://github.com/kubernetes/dashboard

The Dashboard related objects create created in **kubenetes-dashboard** namespace. We can see the objects created for using the following command

```
kubectl get all -n kubernetes-dashboard
```

Step2: Access Kubernetes Dashboard

After we create the dashboard, we can access it using Kubectl. We need to start a proxy server between our local machine and the Kubernetes apiserver. Run the command in a separate terminal.

```
kubectl proxy
```

Kubectl proxy uses **http** for the connection between localhost and the proxy server and **https** for the connection between the proxy and apiserver.

Step3: Open the below URL in browser:

```
http://localhost:8001/api/v1/namespaces/kubernetes-dashboard/services/https:kubernetes-dashboard:/proxy/
```

Create An Authentication Token (RBAC)

Step 4. Creating sample user

We need to create a new user using **Service Account** mechanism of Kubernetes, grant this user admin permissions and login to Dashboard using bearer token tied to this user.

The objects required are of type **ServiceAccount** and **ClusterRoleBinding** in kubernetes-dashboard namespace.

dashboard-adminuser.yaml

```
apiVersion: v1
kind: ServiceAccount
metadata:
  name: admin-sa
  namespace: kubernetes-dashboard
---
```

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: admin-user-role-binding
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: cluster-admin
subjects:
- kind: ServiceAccount
  name: admin-sa
  namespace: kubernetes-dashboard
```

Run the following command for the above yaml file

```
kubectl apply -f dashboard-adminuser.yaml
```

Step5: Create a Token

```
kubectl create token admin-sa -n kubernetes-dashboard
```

Copy the token and paste it into Enter token field on the login screen of Dashboard