1. **Install Docker Desktop 2.3.0.4**
2. **Configure Kubernetes as single node cluster running inside Docker Desktop.**
3. **Configure & Access Kubernetes Dashboard**

**Install Docker Desktop**. Instructions at: <https://hub.docker.com/editions/community/docker-ce-desktop-windows>

**Start Docker**. Docker Desktop has a single node kubernetes cluster that must be configured before we can access the kubernetes dashboard.

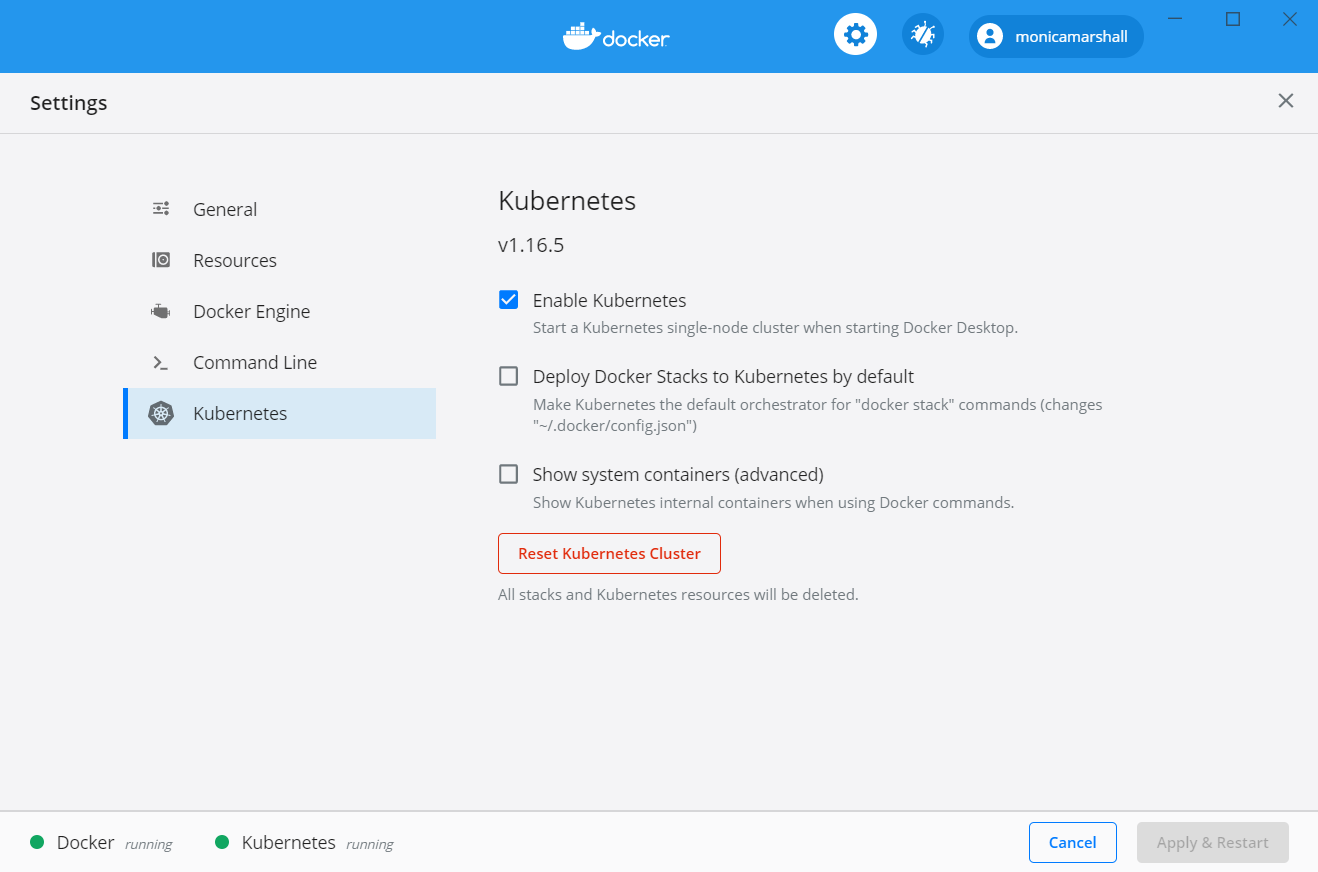
The alternative to using kubernetes shipped with Docker Desktop is minikube. Minikube is another very good option for deploying docker images to kubernetes. Minikube comes with a Docker daemon (server side) which, after experimenting with it, is inaccessible from outside the minikube environment. I have not been able to access it, receiving a connection refused error message whenever I try to access it using an external Docker client.

**Enable Kubernetes**: Right-mouse click on the docker image, select Settings 🡪 Kubernetes. Enable Kubernetes. It will take several minutes for kubernetes to appear running with the green dot at the bottom.

Follow instructions below to set up access to the kubernetes Dashboard. Instructions are at:

<https://github.com/kubernetes/dashboard/blob/master/docs/user/access-control/creating-sample-user.md>

<https://github.com/kubernetes/dashboard>



PS C:\Users\monica> **kubectl apply -f https://raw.githubusercontent.com/kubernetes/dashboard/v2.0.0/aio/deploy/recommended.yaml**

namespace/kubernetes-dashboard unchanged

serviceaccount/kubernetes-dashboard unchanged

service/kubernetes-dashboard unchanged

secret/kubernetes-dashboard-certs unchanged

secret/kubernetes-dashboard-csrf configured

secret/kubernetes-dashboard-key-holder unchanged

configmap/kubernetes-dashboard-settings unchanged

role.rbac.authorization.k8s.io/kubernetes-dashboard unchanged

clusterrole.rbac.authorization.k8s.io/kubernetes-dashboard unchanged

rolebinding.rbac.authorization.k8s.io/kubernetes-dashboard unchanged

clusterrolebinding.rbac.authorization.k8s.io/kubernetes-dashboard unchanged

deployment.apps/kubernetes-dashboard configured

service/dashboard-metrics-scraper configured

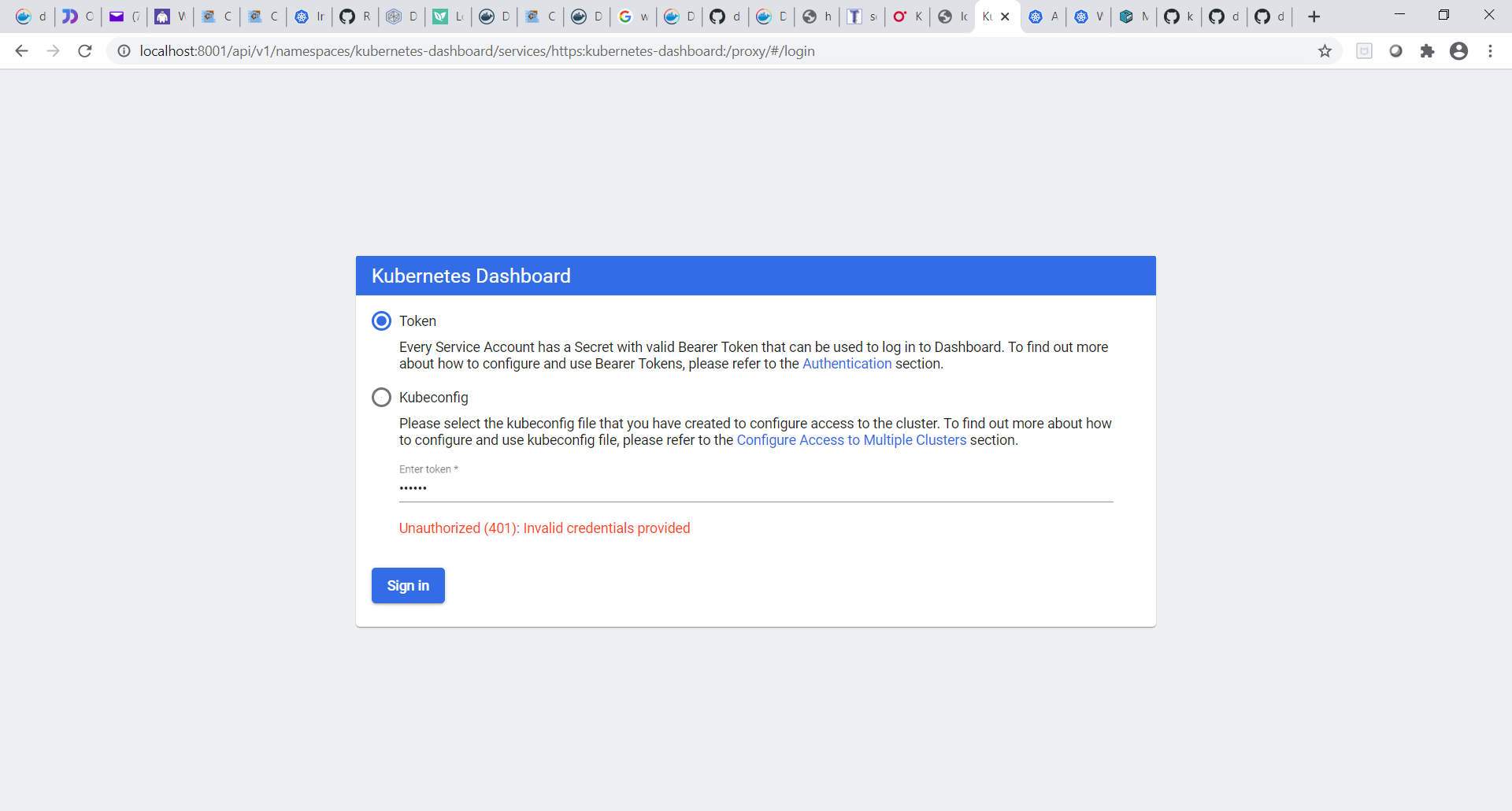
deployment.apps/dashboard-metrics-scraper configured

PS C:\Users\monica> **kubectl proxy**

Starting to serve on 127.0.0.1:8001

**Open a browser, enter this url:**

<http://localhost:8001/api/v1/namespaces/kubernetes-dashboard/services/https:kubernetes-dashboard:/proxy/#/login>



**Create admin account with appropriate rbca to access the dashboard with a valid token.**

Create 2 deployment files, one creates the admin account, the other sets the rbca for it. Then apply the files with the command: kubectl apply –f <file.yaml>

**kubedashboard-adminuser.yaml with content:**

apiVersion: v1

kind: ServiceAccount

metadata:

name: admin-user

namespace: kubernetes-dashboard

**kubedashboard-clusterrolebinding.yaml with content:**

apiVersion: rbac.authorization.k8s.io/v1

kind: ClusterRoleBinding

metadata:

name: admin-user

roleRef:

apiGroup: rbac.authorization.k8s.io

kind: ClusterRole

name: cluster-admin

subjects:

- kind: ServiceAccount

name: admin-user

namespace: kubernetes-dashboard

PS C:\data\minikubeTests\MessageBoardApp> **kubectl apply -f kubedashboard-adminuser.yaml**

serviceaccount/admin-user created

PS C:\data\minikubeTests\MessageBoardApp**> kubectl apply -f .\kubedashboard-clusterrolebinding.yaml**

clusterrolebinding.rbac.authorization.k8s.io/admin-user created

**Retrieve the token** for the admin account entering the command:

PS C:\Users\monica**> kubectl -n kubernetes-dashboard describe secret $(kubectl -n kubernetes-dashboard get secret | sls admin-user | ForEach-Object { $\_ -Split '\s+' } | Select -First 1)**

Name: default-token-jbctz

Namespace: kubernetes-dashboard

Labels: <none>

Annotations: kubernetes.io/service-account.name: default

kubernetes.io/service-account.uid: e83b757f-83db-4f21-9d2d-097165f2813f

Type: kubernetes.io/service-account-token

Data

====

ca.crt: 1025 bytes

namespace: 20 bytes

token: eyJhbGciOiJSUzI1NiIsImtpZCI6Im8tYWVhWlZHVktfYjJYdE9xVjdJMUhsOFlxV2t3ZEtVOGJ6bzFESS1RMkUifQ..UWdn2Uu1Y60p5gZB4YwWj4GCHvb550tvrtvbn-Whoz5F6IwxHIOlgylwIiNkPaJOqv0GgcDiKWB3T08\_Y1bWyjjA5IbksvprofWVdOxpmSL0NSFvRKXbTWAsga\_ZWn7tyY7a4v4z-v0rNbKvF3-edyUpVilW22kM-\_WpP26tlv9fJZzgzOXlv-uwG3XnUGGoWOb4O3H8h0tKJdOE1rFQkipgVvnupVuOKxPKfLdX6-mL43cJ6xb7oULWX7R67K4axD\_pUvb4ZNxjdrK1Xe5-xYSnBZL19NnHfg8eFeu5A\_gyaticGwuOU3dqOWmgz9yUJi-jq78jiK-Bgu9o2yhKvA

Name: kubernetes-dashboard-certs

Namespace: kubernetes-dashboard

Labels: k8s-app=kubernetes-dashboard

Annotations:

Type: Opaque

Data

====

Name: kubernetes-dashboard-csrf

Namespace: kubernetes-dashboard

Labels: k8s-app=kubernetes-dashboard

Annotations:

Type: Opaque

Data

====

csrf: 256 bytes

Name: kubernetes-dashboard-key-holder

Namespace: kubernetes-dashboard

Labels: k8s-app=kubernetes-dashboard

Annotations:

Type: Opaque

Data

====

priv: 1675 bytes

pub: 459 bytes

Name: kubernetes-dashboard-token-wb9wj

Namespace: kubernetes-dashboard

Labels: <none>

Annotations: kubernetes.io/service-account.name: kubernetes-dashboard

kubernetes.io/service-account.uid: 15a9c47b-06a1-4c26-b533-d4da2c3f389a

Type: kubernetes.io/service-account-token

Data

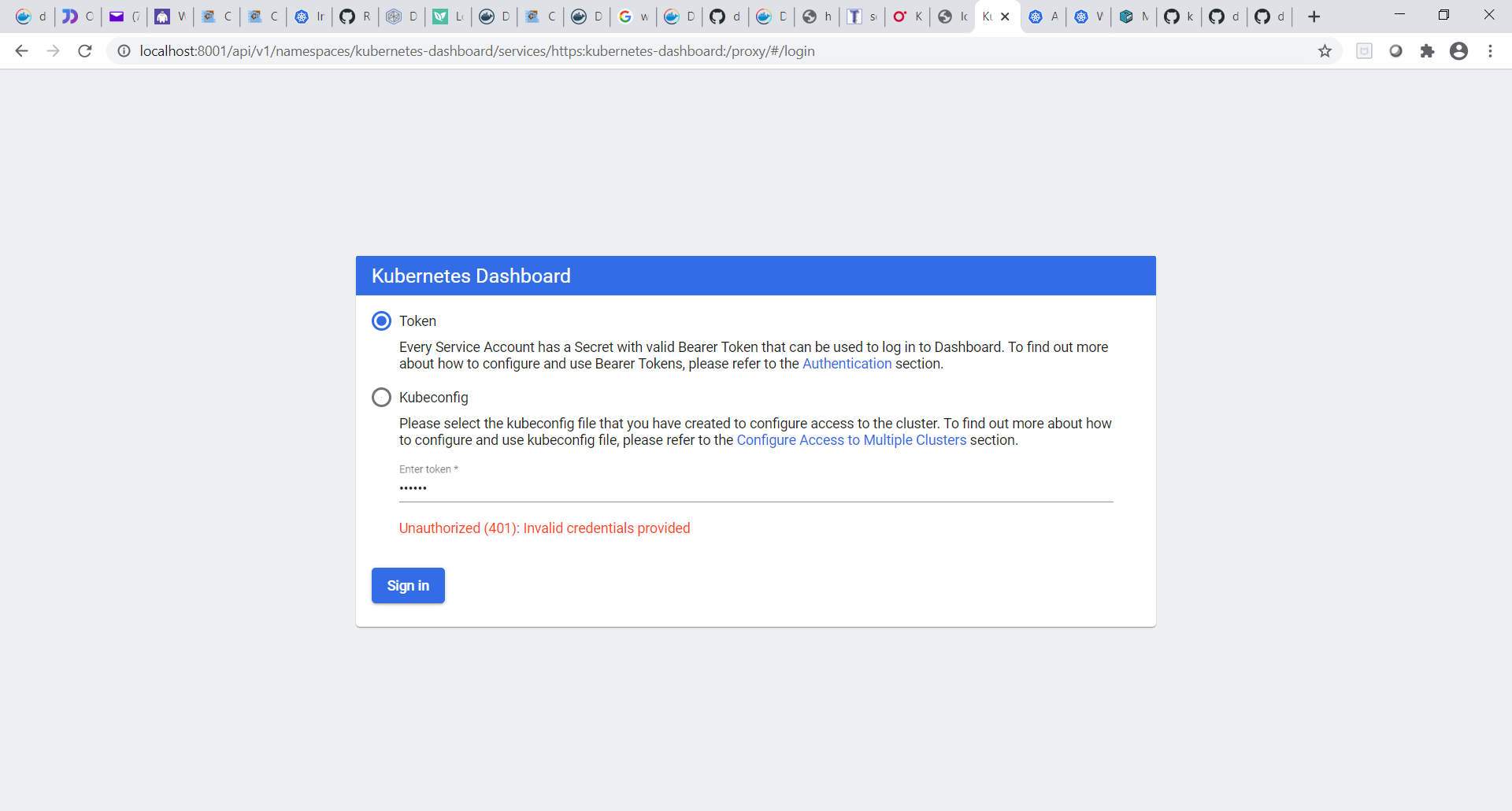
====

ca.crt: 1025 bytes

namespace: 20 bytes

token: eyJhbGciOiJSUzI1NiIsImtpZCI6Im8tYWVhWlZHVktfYjJYdE9xVjdJMUhsOFlxV2t3ZEtVOGJ6bzFESS1RMkUifQ..HIAs9xPQNCLe8geiP5GOiX0OM2ijZ2oQ9RF9ngZDBiKkZthWOu4R3\_KVWUMo4qb7yJCGICvJk1g7ywH\_Dc3WVXnRj6qy7UtfoOAbYPDoALB4kYuf-M3XnwP1dLKrUwMUEM-Dw77lfrx3eMU86NpyvNt1B2U1Iw2mMAojePQvg7zd7g9QE6kGzmbZ4WmOnCt\_jtoiQrvTYZj0NJkrknrrdb6mGAakpNwEkdTDnmrkgUOg\_Cw4cPdkPVMudIYOOu72ySsWJn6k1vHdhL2C-3i0jg2Scwj2K5CPzE-6K0eihCEl-a\_0GvvzIZwn2e1yB4eRgDqUdz9Azq5MreSJqhhFhg

**Copy this token and enter it in the login screen:**



After this you will be able to **access the Kubernetes Dashboard**:

