KUBERNETES DASHBOARD in DOCKER DESKTOP:

#install:

$ kubectl apply -f https://raw.githubusercontent.com/kubernetes/dashboard/v2.2.0/aio/deploy/recommended.yaml

**We will need to download the config file locally so we can edit it (make sure you are copying the most current URL from the official Github repo - do not rely on the version shown in the examples below).**

If using PowerShell:

Invoke-RestMethod -Uri https://raw.githubusercontent.com/kubernetes/dashboard/v2.2.0/aio/deploy/recommended.yaml -Outfile kubernetes-dashboard.yaml

**Add the following two lines underneath --auto-generate-certificates in kubernetes-dashboard.yaml :**

args:

- --auto-generate-certificates

- --enable-skip-login

- --disable-settings-authorizer

**Run the following command inside the directory where you downloaded the dashboard manifest file a few steps ago:**

kubectl apply -f kubernetes-dashboard.yaml

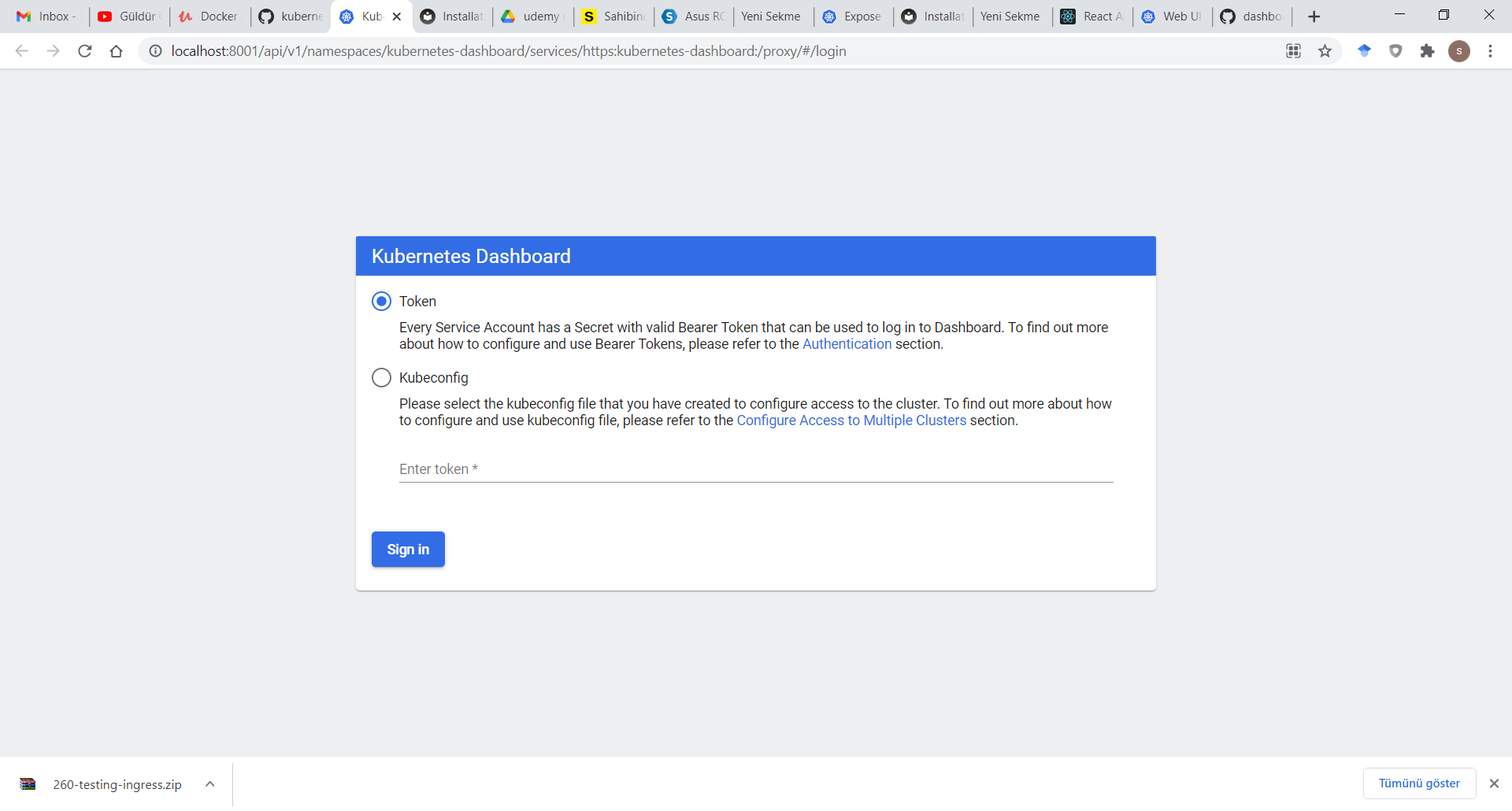
#run command in a new tab:

$ kubectl proxy

visit:

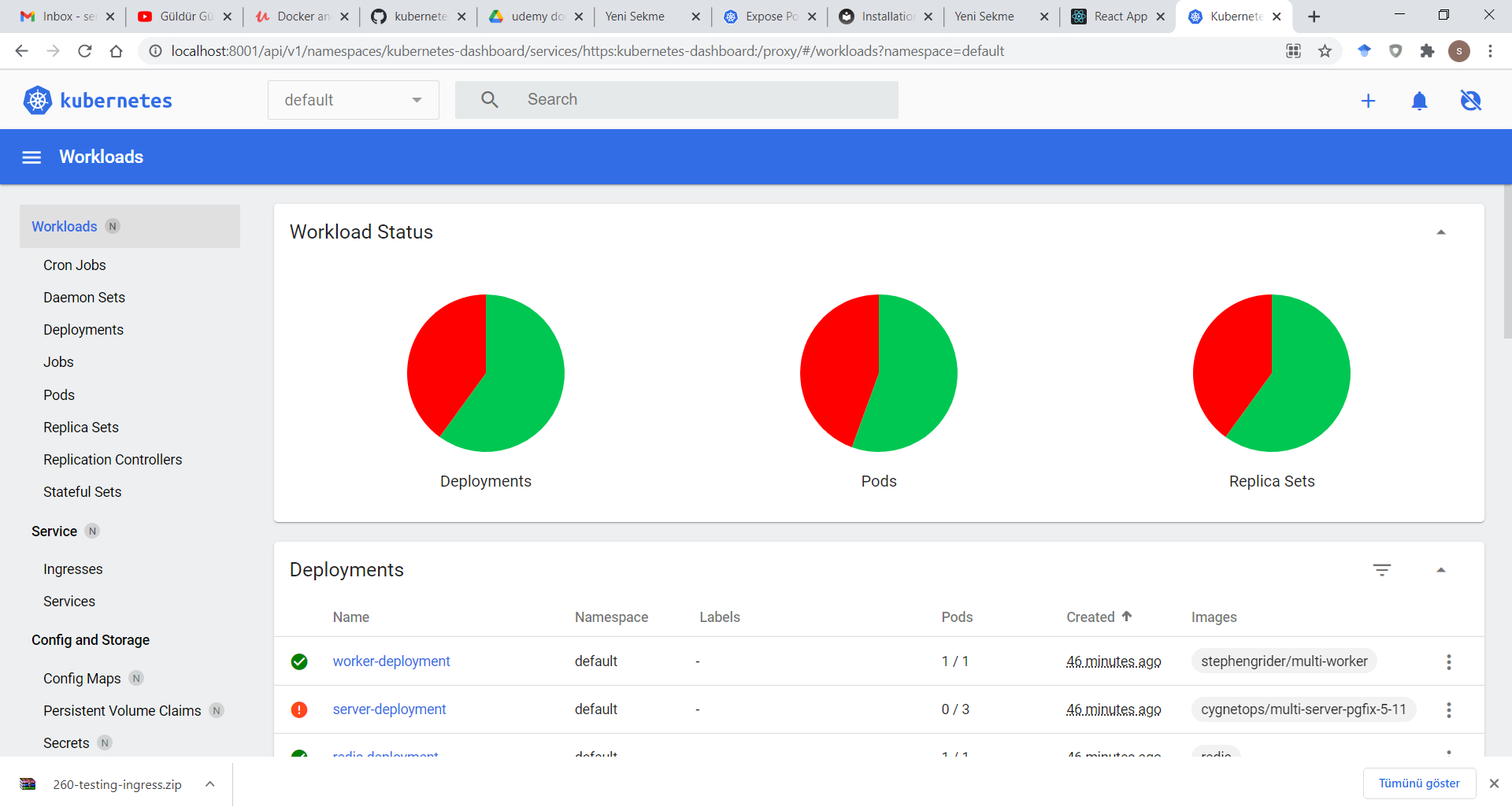
<http://localhost:8001/api/v1/namespaces/kubernetes-dashboard/services/https:kubernetes-dashboard:/proxy/>.

it will show:



**Click the "SKIP" link next to the SIGN IN button.**

then show the dashboard:



kubernetes-dashboard.yaml:

# Copyright 2017 The Kubernetes Authors.

#

# Licensed under the Apache License, Version 2.0 (the "License");

# you may not use this file except in compliance with the License.

# You may obtain a copy of the License at

#

# http://www.apache.org/licenses/LICENSE-2.0

#

# Unless required by applicable law or agreed to in writing, software

# distributed under the License is distributed on an "AS IS" BASIS,

# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

# See the License for the specific language governing permissions and

# limitations under the License.

apiVersion: v1

kind: Namespace

metadata:

name: kubernetes-dashboard

---

apiVersion: v1

kind: ServiceAccount

metadata:

labels:

k8s-app: kubernetes-dashboard

name: kubernetes-dashboard

namespace: kubernetes-dashboard

---

kind: Service

apiVersion: v1

metadata:

labels:

k8s-app: kubernetes-dashboard

name: kubernetes-dashboard

namespace: kubernetes-dashboard

spec:

ports:

- port: 443

targetPort: 8443

selector:

k8s-app: kubernetes-dashboard

---

apiVersion: v1

kind: Secret

metadata:

labels:

k8s-app: kubernetes-dashboard

name: kubernetes-dashboard-certs

namespace: kubernetes-dashboard

type: Opaque

---

apiVersion: v1

kind: Secret

metadata:

labels:

k8s-app: kubernetes-dashboard

name: kubernetes-dashboard-csrf

namespace: kubernetes-dashboard

type: Opaque

data:

csrf: ""

---

apiVersion: v1

kind: Secret

metadata:

labels:

k8s-app: kubernetes-dashboard

name: kubernetes-dashboard-key-holder

namespace: kubernetes-dashboard

type: Opaque

---

kind: ConfigMap

apiVersion: v1

metadata:

labels:

k8s-app: kubernetes-dashboard

name: kubernetes-dashboard-settings

namespace: kubernetes-dashboard

---

kind: Role

apiVersion: rbac.authorization.k8s.io/v1

metadata:

labels:

k8s-app: kubernetes-dashboard

name: kubernetes-dashboard

namespace: kubernetes-dashboard

rules:

# Allow Dashboard to get, update and delete Dashboard exclusive secrets.

- apiGroups: [""]

resources: ["secrets"]

resourceNames: ["kubernetes-dashboard-key-holder", "kubernetes-dashboard-certs", "kubernetes-dashboard-csrf"]

verbs: ["get", "update", "delete"]

# Allow Dashboard to get and update 'kubernetes-dashboard-settings' config map.

- apiGroups: [""]

resources: ["configmaps"]

resourceNames: ["kubernetes-dashboard-settings"]

verbs: ["get", "update"]

# Allow Dashboard to get metrics.

- apiGroups: [""]

resources: ["services"]

resourceNames: ["heapster", "dashboard-metrics-scraper"]

verbs: ["proxy"]

- apiGroups: [""]

resources: ["services/proxy"]

resourceNames: ["heapster", "http:heapster:", "https:heapster:", "dashboard-metrics-scraper", "http:dashboard-metrics-scraper"]

verbs: ["get"]

---

kind: ClusterRole

apiVersion: rbac.authorization.k8s.io/v1

metadata:

labels:

k8s-app: kubernetes-dashboard

name: kubernetes-dashboard

rules:

# Allow Metrics Scraper to get metrics from the Metrics server

- apiGroups: ["metrics.k8s.io"]

resources: ["pods", "nodes"]

verbs: ["get", "list", "watch"]

---

apiVersion: rbac.authorization.k8s.io/v1

kind: RoleBinding

metadata:

labels:

k8s-app: kubernetes-dashboard

name: kubernetes-dashboard

namespace: kubernetes-dashboard

roleRef:

apiGroup: rbac.authorization.k8s.io

kind: Role

name: kubernetes-dashboard

subjects:

- kind: ServiceAccount

name: kubernetes-dashboard

namespace: kubernetes-dashboard

---

apiVersion: rbac.authorization.k8s.io/v1

kind: ClusterRoleBinding

metadata:

name: kubernetes-dashboard

roleRef:

apiGroup: rbac.authorization.k8s.io

kind: ClusterRole

name: kubernetes-dashboard

subjects:

- kind: ServiceAccount

name: kubernetes-dashboard

namespace: kubernetes-dashboard

---

kind: Deployment

apiVersion: apps/v1

metadata:

labels:

k8s-app: kubernetes-dashboard

name: kubernetes-dashboard

namespace: kubernetes-dashboard

spec:

replicas: 1

revisionHistoryLimit: 10

selector:

matchLabels:

k8s-app: kubernetes-dashboard

template:

metadata:

labels:

k8s-app: kubernetes-dashboard

spec:

containers:

- name: kubernetes-dashboard

image: kubernetesui/dashboard:v2.2.0

imagePullPolicy: Always

ports:

- containerPort: 8443

protocol: TCP

args:

- --auto-generate-certificates

- --enable-skip-login

- --disable-settings-authorizer

- --namespace=kubernetes-dashboard

# Uncomment the following line to manually specify Kubernetes API server Host

# If not specified, Dashboard will attempt to auto discover the API server and connect

# to it. Uncomment only if the default does not work.

# - --apiserver-host=http://my-address:port

volumeMounts:

- name: kubernetes-dashboard-certs

mountPath: /certs

# Create on-disk volume to store exec logs

- mountPath: /tmp

name: tmp-volume

livenessProbe:

httpGet:

scheme: HTTPS

path: /

port: 8443

initialDelaySeconds: 30

timeoutSeconds: 30

securityContext:

allowPrivilegeEscalation: false

readOnlyRootFilesystem: true

runAsUser: 1001

runAsGroup: 2001

volumes:

- name: kubernetes-dashboard-certs

secret:

secretName: kubernetes-dashboard-certs

- name: tmp-volume

emptyDir: {}

serviceAccountName: kubernetes-dashboard

nodeSelector:

"kubernetes.io/os": linux

# Comment the following tolerations if Dashboard must not be deployed on master

tolerations:

- key: node-role.kubernetes.io/master

effect: NoSchedule

---

kind: Service

apiVersion: v1

metadata:

labels:

k8s-app: dashboard-metrics-scraper

name: dashboard-metrics-scraper

namespace: kubernetes-dashboard

spec:

ports:

- port: 8000

targetPort: 8000

selector:

k8s-app: dashboard-metrics-scraper

---

kind: Deployment

apiVersion: apps/v1

metadata:

labels:

k8s-app: dashboard-metrics-scraper

name: dashboard-metrics-scraper

namespace: kubernetes-dashboard

spec:

replicas: 1

revisionHistoryLimit: 10

selector:

matchLabels:

k8s-app: dashboard-metrics-scraper

template:

metadata:

labels:

k8s-app: dashboard-metrics-scraper

annotations:

seccomp.security.alpha.kubernetes.io/pod: 'runtime/default'

spec:

containers:

- name: dashboard-metrics-scraper

image: kubernetesui/metrics-scraper:v1.0.6

ports:

- containerPort: 8000

protocol: TCP

livenessProbe:

httpGet:

scheme: HTTP

path: /

port: 8000

initialDelaySeconds: 30

timeoutSeconds: 30

volumeMounts:

- mountPath: /tmp

name: tmp-volume

securityContext:

allowPrivilegeEscalation: false

readOnlyRootFilesystem: true

runAsUser: 1001

runAsGroup: 2001

serviceAccountName: kubernetes-dashboard

nodeSelector:

"kubernetes.io/os": linux

# Comment the following tolerations if Dashboard must not be deployed on master

tolerations:

- key: node-role.kubernetes.io/master

effect: NoSchedule

volumes:

- name: tmp-volume

emptyDir: {}