

Dashboard

- [Dashboard - enable-skip-login](#)
- [Dashboard - Remote Access](#)
- [Dashboard Get Token](#)
- [Docker Desktop Dashboard](#)
- [Deploying the Dashboard UI](#)

Dashboard - enable-skip-login

```
microk8s.kubectl edit deployment/kubernetes-dashboard --namespace=kube-system
```

spec:

containers:

- args:

- --auto-generate-certificates

- --enable-skip-login

image: k8s.gcr.io/kubernetes-dashboard-amd64:v1.10.1

"/tmp/kubectl-edit-snu9z.yaml" 102L, 4077C

Dashboard - Remote Access

```
sudo crontab -e
```

```
@reboot sleep 60 && /snap/bin/microk8s dashboard-proxy > /tmp/dashboard-start.txt
```

Dashboard Get Token

```
sudo cat /var/snap/microk8s/current/credentials/client.config
```

Docker Desktop Dashboard

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

kubectl proxy

<http://localhost:8001/api/v1/namespaces/kubernetes-dashboard/services/https:kubernetes-dashboard:/proxy/#/workloads?namespace=default>

Erstellen und speichern Sie die folgende Definition als **s.yml** . Wenden Sie diese Konfiguration dann mit **kubectl apply -f s.yml** an

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

Erstellen und speichern Sie die folgende Definition als **r.yml** . Wenden Sie diese Konfiguration dann mit **kubectl apply -f r.yml** an

```
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
```

Führen Sie dann den folgenden Befehl aus:

```
kubectl -n kubernetes-dashboard get Secret $(kubectl -n kubernetes-dashboard get sa/admin-user -o jsonpath="{
```

Fügen Sie das Token in den vorherigen Link ein, und Sie erhalten ein funktionierendes Dashboard für Ihren lokalen Cluster.

Deploying the Dashboard UI

<https://kubernetes.io/docs/tasks/access-application-cluster/web-ui-dashboard/>

[create sample user](#)

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
```

Accessing the Dashboard UI

To protect your cluster data, Dashboard deploys with a minimal RBAC configuration by default. Currently, Dashboard only supports logging in with a Bearer Token. To create a token for this demo, you can follow our guide on [creating a sample user](#).

Warning: The sample user created in the tutorial will have administrative privileges and is for educational purposes only.

Command line proxy

You can enable access to the Dashboard using the `kubectl` command-line tool, by running the following command:

```
kubectl proxy
```

Kubectl will make Dashboard available at <http://localhost:8001/api/v1/namespaces/kubernetes-dashboard/services/https:kubernetes-dashboard:/proxy/>.

The UI can *only* be accessed from the machine where the command is executed. See `kubectl proxy --help` for more options.

Note: The kubeconfig authentication method does **not** support external identity providers or X.509 certificate-based authentication.

Welcome view

When you access Dashboard on an empty cluster, you'll see the welcome page. This page contains a link to this document as well as a button to deploy your first application. In addition, you can view which system applications are running by default in the `kube-system` [namespace](#) of your cluster, for example the Dashboard itself.