**### Install git**

**### Install docker ce**

**### Install chocolatey**

@"%SystemRoot%\System32\WindowsPowerShell\v1.0\powershell.exe" -NoProfile -InputFormat None -ExecutionPolicy Bypass -Command "iex ((New-Object System.Net.WebClient).DownloadString('https://chocolatey.org/install.ps1'))" && SET "PATH=%PATH%;%ALLUSERSPROFILE%\chocolatey\bin"

# choco

**### Install kubectl**

choco install kubernetes-cli

# kubectl

**### Docker settings**

>> Daemon

>> Registry mirrors

>> https://registry.docker-cn.com

>> Apply

**### Clone k8s-for-docker-desktop**

git clone https://github.com/AliyunContainerService/k8s-for-docker-desktop

**### PowerShell**

cd <dir>/k8s-for-docker-desktop

set-ExecutionPolicy RemoteSigned

./load\_images.ps1

**### Docker settings**

>> Kubernetes >> Enable Kubernetes >> Kubernetes >> Apply

**### Deploy kubernetes-dashboard**

Edit >> k8s-for-docker-desktop/kubernetes-dashboard.yaml

Edit >> image: k8s.gcr.io/kubernetes-dashboard-amd64:v1.10.0

kubectl create -f kubernetes-dashboard.yaml

kubectl proxy -p 8001

http://localhost:8001/api/v1/namespaces/kube-system/services/https:kubernetes-dashboard:/proxy/#!/overview?namespace=default

**### Build k8s-demo**

>> k8s-demo/Dockerfile

FROM microsoft/dotnet:2.1-aspnetcore-runtime AS base

WORKDIR /app

EXPOSE 80

FROM microsoft/dotnet:2.1-sdk AS build

WORKDIR /src

COPY . .

RUN dotnet restore

RUN dotnet build -c Release -o /app

FROM build AS publish

RUN dotnet publish -c Release -o /app

FROM base AS final

WORKDIR /app

COPY --from=publish /app .

ENTRYPOINT ["dotnet", "k8s-demo.dll"]

docker build -t k8s-demo:latest

docker run -d -p 8085:80 k8s-demo:latest

http://localhost:8085/api/values

docker tag k8s-demo:latest <docker id>/k8s-demo:latest

docker login --username <docker id>

docker push <docker id>/k8s-demo:latest

>> k8s-demo/deploy.yaml

apiVersion: apps/v1

kind: Deployment

metadata:

name: k8s-demo

namespace: default

labels:

name: k8s-demo

spec:

replicas: 2

selector:

matchLabels:

name: k8s-demo

template:

metadata:

labels:

name: k8s-demo

spec:

containers:

- name: k8s-demo

image: <docker id>/k8s-demo:latest

ports:

- containerPort: 80

imagePullPolicy: Always

---

kind: Service

apiVersion: v1

metadata:

name: k8s-demo

namespace: default

spec:

type: NodePort

ports:

- port: 80

targetPort: 80

selector:

name: k8s-demo

**### Deploy k8s-demo**

kubectl create -f deploy.yaml

kubectl get svc -n default

http://localhost:32711/api/values

kubectl proxy -p 8001

http://localhost:8001/api/v1/namespaces/kube-system/services/https:kubernetes-dashboard:/proxy/#!/overview?namespace=default