## Create the images using docker-compose

**docker-compose build**

We should have 2 images

1. demo-webapp
2. demo-webapi

## Deploying Resources

kubectl create -f frontend-webapp.yaml

kubectl create -f backend-webapi.yaml

***# Force replace, delete and then re-create the resource. Will cause a service outage.***

kubectl replace --force -f ./frontend-webapp.yaml

## Scaling Resources

#increase the # of pods to 3

kubectl scale --replicas=3 deployment/demowebapp

kubectl scale --replicas=3 deployment/demowebapi

## Interacting with running Pods

kubectl logs <<podName>>

***# Attach to Running Container***

kubectl exec <<my-pod>> -it sh

kubectl top pod POD\_NAME --containers

## Dashboard

Install Dashboard Service (1.10.0 version can skip auth)

kubectl apply -f https://raw.githubusercontent.com/kubernetes/dashboard/v1.10.0/src/deploy/recommended/kubernetes-dashboard.yaml

### Start Proxy

Kubectl proxy

### Launch Dashboard

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

cheat sheet

<https://kubernetes.io/docs/reference/kubectl/cheatsheet/#creating-objects>

## Pull an image from a Private Registry

<https://kubernetes.io/docs/tasks/configure-pod-container/pull-image-private-registry/>

Create a secret named **acr** to hold the Container registry credentials

**kubectl create secret docker-registry acr --docker-server=<your-registry-server> --docker-username=<your-name> --docker-password=<your-pword> --docker-email=<your-email>**

in VS2017, publish projects WebApi and WebApp to the same container registry

