



## **Thought**Works®

## ASP.NET CORE+K8S的容器化之路

张阳







# 软件的演化趋势

#### 目标

无论用什么语言,都可以在使用相同的方式运行起 来

你的应用程序能够得到最佳的部署和管理

所有的资源都是按需分配的

### 容器+编排+云

无论用什么语言,都可以在使用相同的方式运行起来 => 容器(Container)

你的应用程序能够得到最佳的部署和管理 => 编排 工具(Orchestrator)

所有的资源都是按需分配的 => 云(Cloud)

## Demo: 把<u>ASP.NET</u> Core应用程序跑在 Docker中

#### **DEMO: DOCKER**

dotnet new webapi

docker build -t hellowed:v1.

docker run -d -p 8001:80 hellowed:v1

## Demo:把Docker镜像上传到ACR( Azure container repository)

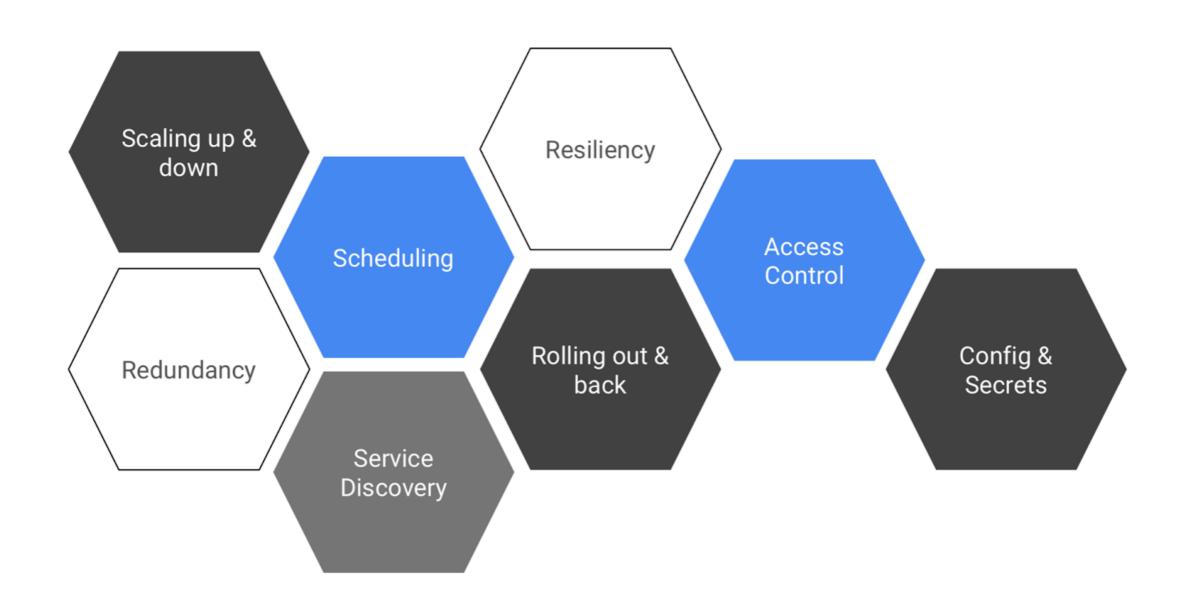
#### **DEMO:ACR**

docker login

docker tag helloweb:v1 restairline.azurecr.io/helloweb:v1

docker push <u>restairline.azurecr.io/helloweb:v1</u>

### 只有DOCKER是不够的



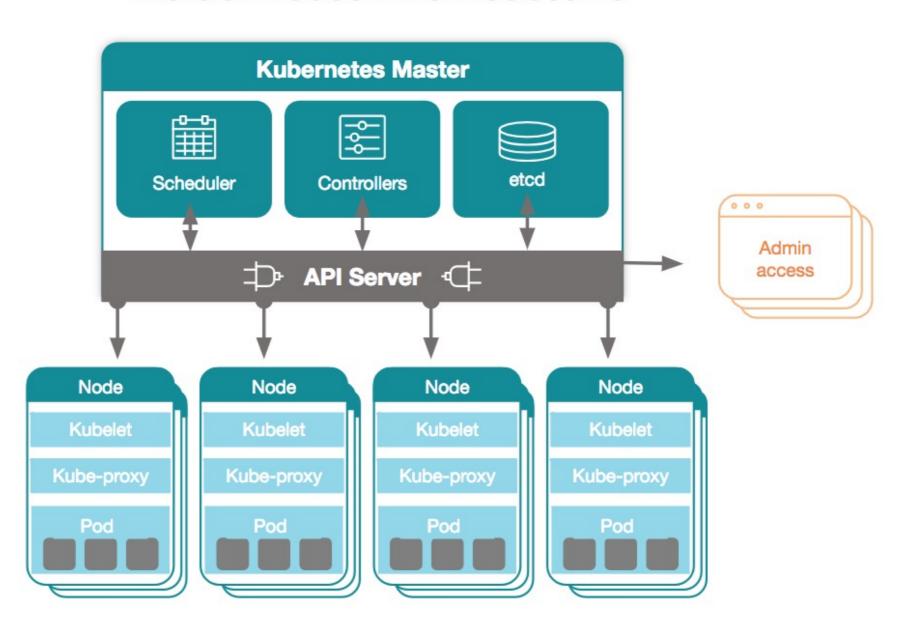
#### **KUBERNETES**

- Kubernetes是一个开源的,用于管理云平台中多个主机上的容器化的应用,Kubernetes的目标是让部署容器化的应用简单并且高效(powerful),Kubernetes提供了应用部署,规划,更新,统护的一种机制。
- 可移植,支持公有云,私有云
- 支持多个容器,包括docker
- 100% 开源, go语言编写



#### **K8S ARCHITECTURE**

#### **Kubernetes Architecture**



#### **DEMO**

## Demo:k8s

https://github.com/twzhangyang/k8s-tutorial/blob/master/tutorial1/demo1.txt

#### **DEMO: DEPLOYMENT**

kubectl run kubernetes-bootcamp --image=gcr.io/google-samples/

kubectl get pods

kubectl get deployments

**kubectl describe deployments** 

**kubectl** describe pods

kubectl exec -it \$pod\_name bash

#### **DEMO:SERVICE**

kubectl expose deployment/kubernetes-bootcamp --type="NodePort" --port 8080

kubectl get service

**kubectl describe replicasets** 

curl \$(minikube ip):\$NODE\_PORT

#### **DEMO:SCALING UP/SCALING DOWN & ROLLING UPDATE**

kubectl scale deployment/kubernetes-bootcamp --replicas=2

kubectl set image deployments/kubernetes-bootcamp kubernetes-bootcamp=jocatalin/kubernetes-bootcamp:v2

kubectl set image deployments/kubernetes-bootcamp kubernetes-bootcamp=gcr.io/google-samples/kubernetes-bootcamp:v10

kubectl rollout undo deployments/kubernetes-bootcamp

#### **DEMO**

## **Demo:AKS**

https://github.com/twzhangyang/k8s-tutorial/blob/master/tutorial2/aks.txt

#### **DEMO:AKS**

#### az aks get-versions

az aks create -g restairline-group -n restairlineCluster -k 1.13.5 --node-count 1 --node-vm-size Standard\_D2\_v2 --generate-ssh-keys

az aks get-credentials -n restairlineCluster -g restairline-group

#### kubectl cluster-info

az aks browse --resource-group restairline-group --name

kubectl create clusterrolebinding kubernetes-dashboard -n kube-system -- clusterrole=cluster-admin --serviceaccount=kube-system:kubernetes-dashboard

#### **DEMO**

## Demo:Run k8s on AKS

<a href="https://github.com/twzhangyang/k8s-tutorial/blob/master/tutorial2/deploy.txt">https://github.com/twzhangyang/k8s-tutorial/blob/master/tutorial2/deploy.txt</a>

https://github.com/twzhangyang/RestAirline/tree/master/k8s

#### **DEMO:AKS**

kubectl create secret docker-registry registry-key --docker-server=restairline.azurecr.io --docker-username=restairline --docker-password=FXnbyQuuuF+MI3J4o9Z30TDhBAOk3fel --docker-email=test@test.com

az aks create -g restairline-group -n restairlineCluster -k 1.13.5 --node-count 1 --node-vm-size Standard\_D2\_v2 --generate-ssh-keys

kubectl apply -f conf\_local.yaml

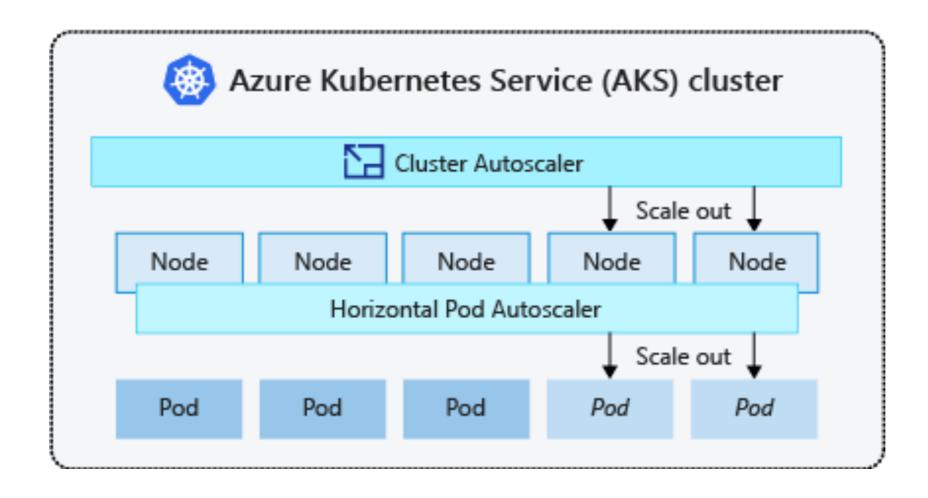
kubectl apply -f sql-data.yaml

kubectl apply -f restairlineapi.yaml

kubectl scale deployment restairline-api --replicas=3

az aks scale --name restairlineCluster --resource-group restairline-group -c 2

#### **AUTOSCALING**



#### **AZURE PIPELINES**

持续生成、测试并部署到任何平台和云

## Demo:Azure pipeline

### **HELM**

https://helm.sh/

# 谢谢!

张阳

