Docker, Containers, Registry and Kubernetes

Thursday, August 8, 2019

9:55 PM

Tutorial: <https://docs.microsoft.com/en-us/azure/aks/tutorial-kubernetes-prepare-app> (chapters 1-7)

C:\Users\DC-LAPTOP>**git clone** [**https://github.com/Azure-Samples/azure-voting-app-redis.git**](https://github.com/Azure-Samples/azure-voting-app-redis.git)

C:\Users\DC-LAPTOP>**cd azure-voting-app-redis**

C:\Users\DC-LAPTOP\azure-voting-app-redis>

**Create container images**

C:\Users\DC-LAPTOP\azure-voting-app-redis>**docker-compose up -d**

**Deploy and use Azure Container Registry**

**Create an Azure Container Registry**

C:\Users\DC-LAPTOP\azure-voting-app-redis>**az group create --name dc1-RGRP --location westus**

{

"id": "/subscriptions/dee29fa2-d695-439c-80a5-158ce469882b/resourceGroups/dc1-RGRP",

"location": "westus",

"managedBy": null,

"name": "dc1-RGRP",

"properties": {

"provisioningState": "Succeeded"

},

"tags": null,

"type": null

}

C:\Users\DC-LAPTOP\azure-voting-app-redis>**az acr create --resource-group dc1-RGRP --name dc1AzureCubernetesRegistry --sku Basic**

{

"adminUserEnabled": false,

"creationDate": "2019-08-09T02:29:06.095267+00:00",

"id": "/subscriptions/dee29fa2-d695-439c-80a5-158ce469882b/resourceGroups/dc1-RGRP/providers/Microsoft.ContainerRegistry/registries/dc1AzureCubernetesRegistry",

"location": "westus",

"loginServer": "dc1azurecubernetesregistry.azurecr.io",

"name": "dc1AzureCubernetesRegistry",

"networkRuleSet": null,

"provisioningState": "Succeeded",

"resourceGroup": "dc1-RGRP",

"sku": {

"name": "Basic",

"tier": "Basic"

},

"status": null,

"storageAccount": null,

"tags": {},

"type": "Microsoft.ContainerRegistry/registries"

}

**Log in to the container registry**

C:\Users\DC-LAPTOP\azure-voting-app-redis>**az acr login --name dc1AzureCubernetesRegistry**

Login Succeeded

C:\Users\DC-LAPTOP\azure-voting-app-redis>**docker images**

REPOSITORY TAG IMAGE ID CREATED SIZE

dc1azurecubernetesregistry.azurecr.io/azure-vote-front v2 e0c1200e7a3f 47 hours ago 952MB

azure-vote-front latest e0c1200e7a3f 47 hours ago 952MB

dc1containerregistry/azure-vote-front v1 07f91077cc84 11 days ago 952MB

dc1azurecontainerregistry.azurecr.io/azure-vote-front v1 07f91077cc84 11 days ago 952MB

dc1azurecubernetesregistry.azurecr.io/azure-vote-front v1 07f91077cc84 11 days ago 952MB

dc1containerregistry.azurecr.io/azure-vote-front v1 07f91077cc84 11 days ago 952MB

redis latest 598a6f110d01 4 weeks ago 118MB

tiangolo/uwsgi-nginx-flask python3.6 0e5630c7a817 2 months ago 952MB

C:\Users\DC-LAPTOP\azure-voting-app-redis>**az acr list --resource-group dc1-RGRP --query "[].{acrLoginServer:loginServer}" --output table**

AcrLoginServer

-------------------------------------

dc1azurecubernetesregistry.azurecr.io

**Tag a container image**

C:\Users\DC-LAPTOP\azure-voting-app-redis>**docker tag azure-vote-front dc1azurecubernetesregistry.azurecr.io/azure-vote-front:v1**

C:\Users\DC-LAPTOP\azure-voting-app-redis>**docker images**

REPOSITORY TAG IMAGE ID CREATED SIZE

dc1azurecubernetesregistry.azurecr.io/azure-vote-front v1 e0c1200e7a3f 47 hours ago 952MB

dc1azurecubernetesregistry.azurecr.io/azure-vote-front v2 e0c1200e7a3f 47 hours ago 952MB

azure-vote-front latest e0c1200e7a3f 47 hours ago 952MB

dc1containerregistry/azure-vote-front v1 07f91077cc84 11 days ago 952MB

dc1azurecontainerregistry.azurecr.io/azure-vote-front v1 07f91077cc84 11 days ago 952MB

dc1containerregistry.azurecr.io/azure-vote-front v1 07f91077cc84 11 days ago 952MB

dc1azurecubernetesregistry.azurecr.io/azure-vote-front <none> 07f91077cc84 11 days ago 952MB

redis latest 598a6f110d01 4 weeks ago 118MB

tiangolo/uwsgi-nginx-flask python3.6 0e5630c7a817 2 months ago 952MB

**Push images to registry**

**docker push dc1azurecubernetesregistry.azurecr.io/azure-vote-front:v1**

C:\Users\DC-LAPTOP\azure-voting-app-redis>**az acr repository list --name dc1azurecubernetesregistry --output table**

Result

----------------

azure-vote-front

C:\Users\DC-LAPTOP\azure-voting-app-redis>**az acr repository show-tags --name dc1AzureCubernetesRegistry --repository azure-vote-front --output table**

Result

--------

v1

**Deploy an Azure Kubernetes Service (AKS) cluster**

**Create a service principal**

C:\Users\DC-LAPTOP\azure-voting-app-redis>**az ad sp create-for-rbac --skip-assignment**

{

"appId": "8346f74f-4904-4c19-abd7-63c7cc42c8a6",

"displayName": "azure-cli-2019-08-09-02-58-51",

"name": "<http://azure-cli-2019-08-09-02-58-51>",

"password": "d8d230e6-5981-410e-9e91-08272beac1cc",

"tenant": "6bdf9621-fbe5-401d-9c2e-8c9ed7555112"

}

C:\Users\DC-LAPTOP\azure-voting-app-redis>**az acr show --resource-group dc1-RGRP --name dc1AzureCubernetesRegistry --query "id" --output tsv**

/subscriptions/dee29fa2-d695-439c-80a5-158ce469882b/resourceGroups/dc1-RGRP/providers/Microsoft.ContainerRegistry/registries/dc1AzureCubernetesRegistry

**Configure ACR authentication**

C:\Users\DC-LAPTOP\azure-voting-app-redis>**az role assignment create --assignee "8346f74f-4904-4c19-abd7-63c7cc42c8a6" --scope "/subscriptions/dee29fa2-d695-439c-80a5-158ce469882b/resourceGroups/dc1-RGRP/providers/Microsoft.ContainerRegistry/registries/dc1AzureCubernetesRegistry" --role acrpull**

{

"canDelegate": null,

"id": "/subscriptions/dee29fa2-d695-439c-80a5-158ce469882b/resourceGroups/dc1-RGRP/providers/Microsoft.ContainerRegistry/registries/dc1AzureCubernetesRegistry/providers/Microsoft.Authorization/roleAssignments/023a775f-fc09-43f5-aae2-aacab6e80d1e",

"name": "023a775f-fc09-43f5-aae2-aacab6e80d1e",

"principalId": "e5eff6b7-9358-405e-b972-872ef9f30487",

"principalType": "ServicePrincipal",

"resourceGroup": "dc1-RGRP",

"roleDefinitionId": "/subscriptions/dee29fa2-d695-439c-80a5-158ce469882b/providers/Microsoft.Authorization/roleDefinitions/7f951dda-4ed3-4680-a7ca-43fe172d538d",

"scope": "/subscriptions/dee29fa2-d695-439c-80a5-158ce469882b/resourceGroups/dc1-RGRP/providers/Microsoft.ContainerRegistry/registries/dc1AzureCubernetesRegistry",

"type": "Microsoft.Authorization/roleAssignments"

}

**Create a Kubernetes cluster**

C:\Users\DC-LAPTOP\azure-voting-app-redis>**az aks create --resource-group dc1-RGRP --name dc1AKSCluster --node-count 1 --service-principal "8346f74f-4904-4c19-abd7-63c7cc42c8a6" --client-secret "d8d230e6-5981-410e-9e91-08272beac1cc" --generate-ssh-keys**

{

"aadProfile": null,

"addonProfiles": null,

"agentPoolProfiles": [

{

"availabilityZones": null,

"count": 1,

"enableAutoScaling": null,

"maxCount": null,

"maxPods": 110,

"minCount": null,

"name": "nodepool1",

"orchestratorVersion": "1.12.8",

"osDiskSizeGb": 100,

"osType": "Linux",

"provisioningState": "Succeeded",

"type": "AvailabilitySet",

"vmSize": "Standard\_DS2\_v2",

"vnetSubnetId": null

}

],

"apiServerAuthorizedIpRanges": null,

"dnsPrefix": "dc1AKSClus-dc1-RGRP-dee29f",

"enablePodSecurityPolicy": null,

"enableRbac": true,

"fqdn": "dc1aksclus-dc1-rgrp-dee29f-87446ccc.hcp.westus.azmk8s.io",

"id": "/subscriptions/dee29fa2-d695-439c-80a5-158ce469882b/resourcegroups/dc1-RGRP/providers/Microsoft.ContainerService/managedClusters/dc1AKSCluster",

"identity": null,

"kubernetesVersion": "1.12.8",

"linuxProfile": {

"adminUsername": "azureuser",

"ssh": {

"publicKeys": [

{

"keyData": "ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDCQg3PKbTnCZ2IJMPGeJSf1J+btg5BPJoM8m+qTVqlPIS/cAkMGEaA2fV8CMY9YKnCJqPizmSmx/GX0zR4pBSmujAJmz3SWJx3BANpLIDbUfsc115PapvXml4mjx2nqUBdDvDqj1VtNvI8KHL3jR+BnbR+DJTJ9yr3i9jGLfRWSOOLex5s3ipM8vFqnaTrK/dUVxjiEFxpJRx/NqCQXgJGp+nWV/zeYinxSRhO3Grc3CsLiMo1IKpqvpt79mlrNn3pYrUC9CD3Snwy/dvMtsJk3Mx/LplTO0cK+D19vGrhNm16GDEZ4s6q9pyZUVZYw4IW10tdYng9KB2AIClnBbRT"

}

]

}

},

"location": "westus",

"maxAgentPools": 1,

"name": "dc1AKSCluster",

"networkProfile": {

"dnsServiceIp": "10.0.0.10",

"dockerBridgeCidr": "172.17.0.1/16",

"loadBalancerSku": "basic",

"networkPlugin": "kubenet",

"networkPolicy": null,

"podCidr": "10.244.0.0/16",

"serviceCidr": "10.0.0.0/16"

},

"nodeResourceGroup": "MC\_dc1-RGRP\_dc1AKSCluster\_westus",

"provisioningState": "Succeeded",

"resourceGroup": "dc1-RGRP",

"servicePrincipalProfile": {

"clientId": "8346f74f-4904-4c19-abd7-63c7cc42c8a6",

"secret": null

},

"tags": null,

"type": "Microsoft.ContainerService/ManagedClusters",

"windowsProfile": null

}

**Create a Kubernetes cluster**

**az aks install-cli**

C:\Users\DC-LAPTOP\azure-voting-app-redis>**az aks get-credentials --resource-group dc1-RGRP --name dc1AKSCluster**

A different object named dc1AKSCluster already exists in your kubeconfig file.

Overwrite? (y/n): y

A different object named clusterUser\_dc1-RGRP\_dc1AKSCluster already exists in your kubeconfig file.

Overwrite? (y/n): y

Merged "dc1AKSCluster" as current context in C:\Users\DC-LAPTOP\.kube\config

**Connect to cluster using kubectl**

**az aks get-credentials --resource-group dc1-RGRP --name** [**dc1AKSCluster**](https://portal.azure.com/#@davescheemagmail.onmicrosoft.com/resource/subscriptions/dee29fa2-d695-439c-80a5-158ce469882b/resourceGroups/dc1-RGRP/providers/Microsoft.ContainerService/managedClusters/dc1AKSCluster)

C:\Users\DC-LAPTOP\azure-voting-app-redis>**kubectl get nodes**

NAME STATUS ROLES AGE VERSION

aks-nodepool1-10061322-0 Ready agent 12m v1.12.8

**Run applications in Azure Kubernetes Service (AKS)**

**Update the manifest file**

C:\Users\DC-LAPTOP\azure-voting-app-redis>**az acr list --resource-group dc1-RGRP --query "[].{acrLoginServer:loginServer}" --output table**

AcrLoginServer

-------------------------------------

dc1azurecubernetesregistry.azurecr.io

**Deploy the application**

C:\Users\DC-LAPTOP\azure-voting-app-redis>**kubectl apply -f C:\Users\DC-LAPTOP\azure-voting-app-redis\azure-vote-all-in-one-redis.yaml**

deployment.apps/azure-vote-back created

service/azure-vote-back created

deployment.apps/azure-vote-front created

service/azure-vote-front created

C:\Users\DC-LAPTOP\azure-voting-app-redis>**kubectl get service azure-vote-front --watch**

NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE

azure-vote-front LoadBalancer 10.0.100.207 40.118.238.48 80:32150/TCP 82s

**Scale applications in Azure Kubernetes Service (AKS)**

**Manually scale pods**

C:\Users\DC-LAPTOP\azure-voting-app-redis>**kubectl get pods**

NAME READY STATUS RESTARTS AGE

azure-vote-back-78d45566f4-2n2jc 1/1 Running 0 4m18s

azure-vote-front-c6598b664-xnkhl 1/1 Running 0 4m18s

C:\Users\DC-LAPTOP\azure-voting-app-redis>**kubectl scale --replicas=2 deployment/azure-vote-front**

deployment.extensions/azure-vote-front scaled

C:\Users\DC-LAPTOP\azure-voting-app-redis>**kubectl get pods**

NAME READY STATUS RESTARTS AGE

azure-vote-back-78d45566f4-2n2jc 1/1 Running 0 6m11s

azure-vote-front-c6598b664-8kx7m 1/1 Running 0 39s

azure-vote-front-c6598b664-xnkhl 1/1 Running 0 6m11s

**Autoscale pods**C:\Users\DC-LAPTOP>**az aks show --resource-group dc1-RGRP --name dc1AKSCluster --query kubernetesVersion**

"1.12.8"

**git clone** [**https://github.com/kubernetes-incubator/metrics-server.git**](https://github.com/kubernetes-incubator/metrics-server.git)

C:\Users\DC-LAPTOP\azure-voting-app-redis>**kubectl autoscale deployment azure-vote-front --cpu-percent=30 --min=2 --max=3**

horizontalpodautoscaler.autoscaling/azure-vote-front autoscaled

C:\Users\DC-LAPTOP\azure-voting-app-redis>**kubectl get hpa**

NAME REFERENCE TARGETS MINPODS MAXPODS REPLICAS AGE

azure-vote-front Deployment/azure-vote-front 0%/30% 2 3 2 51s

**Manually scale AKS nodes**

C:\Users\DC-LAPTOP\azure-voting-app-redis>**az aks scale --resource-group dc1-RGRP --name dc1AKSCluster --node-count 2**

{

"aadProfile": null,

"addonProfiles": null,

"agentPoolProfiles": [

{

"availabilityZones": null,

"count": 2,

"enableAutoScaling": null,

"maxCount": null,

"maxPods": 110,

"minCount": null,

"name": "nodepool1",

"orchestratorVersion": "1.12.8",

"osDiskSizeGb": 100,

"osType": "Linux",

"provisioningState": "Succeeded",

"type": "AvailabilitySet",

"vmSize": "Standard\_DS2\_v2",

"vnetSubnetId": null

}

],

"apiServerAuthorizedIpRanges": null,

"dnsPrefix": "dc1AKSClus-dc1-RGRP-dee29f",

"enablePodSecurityPolicy": null,

"enableRbac": true,

"fqdn": "dc1aksclus-dc1-rgrp-dee29f-87446ccc.hcp.westus.azmk8s.io",

"id": "/subscriptions/dee29fa2-d695-439c-80a5-158ce469882b/resourcegroups/dc1-RGRP/providers/Microsoft.ContainerService/managedClusters/dc1AKSCluster",

"identity": null,

"kubernetesVersion": "1.12.8",

"linuxProfile": {

"adminUsername": "azureuser",

"ssh": {

"publicKeys": [

{

"keyData": "ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDCQg3PKbTnCZ2IJMPGeJSf1J+btg5BPJoM8m+qTVqlPIS/cAkMGEaA2fV8CMY9YKnCJqPizmSmx/GX0zR4pBSmujAJmz3SWJx3BANpLIDbUfsc115PapvXml4mjx2nqUBdDvDqj1VtNvI8KHL3jR+BnbR+DJTJ9yr3i9jGLfRWSOOLex5s3ipM8vFqnaTrK/dUVxjiEFxpJRx/NqCQXgJGp+nWV/zeYinxSRhO3Grc3CsLiMo1IKpqvpt79mlrNn3pYrUC9CD3Snwy/dvMtsJk3Mx/LplTO0cK+D19vGrhNm16GDEZ4s6q9pyZUVZYw4IW10tdYng9KB2AIClnBbRT"

}

]

}

},

"location": "westus",

"maxAgentPools": 1,

"name": "dc1AKSCluster",

"networkProfile": {

"dnsServiceIp": "10.0.0.10",

"dockerBridgeCidr": "172.17.0.1/16",

"loadBalancerSku": "basic",

"networkPlugin": "kubenet",

"networkPolicy": null,

"podCidr": "10.244.0.0/16",

"serviceCidr": "10.0.0.0/16"

},

"nodeResourceGroup": "MC\_dc1-RGRP\_dc1AKSCluster\_westus",

"provisioningState": "Succeeded",

"resourceGroup": "dc1-RGRP",

"servicePrincipalProfile": {

"clientId": "8346f74f-4904-4c19-abd7-63c7cc42c8a6",

"secret": null

},

"tags": null,

"type": "Microsoft.ContainerService/ManagedClusters",

"windowsProfile": null

}

C:\Users\DC-LAPTOP\azure-voting-app-redis>docker-compose up --build -d

Building azure-vote-front

Step 1/3 : FROM tiangolo/uwsgi-nginx-flask:python3.6

---> 0e5630c7a817

Step 2/3 : RUN pip install redis

---> Using cache

---> 9a6b390a019e

Step 3/3 : ADD /azure-vote /app

---> 02919d4e799a

Successfully built 02919d4e799a

Successfully tagged azure-vote-front:latest

Starting azure-vote-back ... done

Recreating azure-vote-front ... done

C:\Users\DC-LAPTOP\azure-voting-app-redis>az acr list --resource-group dc1-RGRP --query "[].{acrLoginServer:loginServer}" --output table

AcrLoginServer

-------------------------------------

dc1azurecubernetesregistry.azurecr.io

C:\Users\DC-LAPTOP\azure-voting-app-redis>**docker tag azure-vote-front dc1azurecubernetesregistry.azurecr.io/azure-vote-front:v2**

C:\Users\DC-LAPTOP\azure-voting-app-redis>**docker push dc1azurecubernetesregistry.azurecr.io/azure-vote-front:v2**

The push refers to repository [dc1azurecubernetesregistry.azurecr.io/azure-vote-front]

05b4e6a6ffc1: Pushed

ce1834399843: Layer already exists

ec348085b0e6: Layer already exists

c2be8853e0b2: Layer already exists

0f1151f5fc99: Layer already exists

00399b079947: Layer already exists

c82d454eb914: Layer already exists

…

C:\Users\DC-LAPTOP\azure-voting-app-redis>**kubectl get pods**

NAME READY STATUS RESTARTS AGE

azure-vote-back-78d45566f4-2n2jc 1/1 Running 0 30m

azure-vote-front-c6598b664-8kx7m 1/1 Running 0 24m

azure-vote-front-c6598b664-xnkhl 1/1 Running 0 30m

C:\Users\DC-LAPTOP\azure-voting-app-redis>**kubectl scale --replicas=3 deployment/azure-vote-front**

deployment.extensions/azure-vote-front scaled

C:\Users\DC-LAPTOP\azure-voting-app-redis>**kubectl set image deployment azure-vote-front azure-vote-front=dc1azurecubernetesregistry.azurecr.io/azure-vote-front:v2**

deployment.extensions/azure-vote-front image updated

C:\Users\DC-LAPTOP\azure-voting-app-redis>**kubectl get pods**

NAME READY STATUS RESTARTS AGE

azure-vote-back-78d45566f4-2n2jc 1/1 Running 0 33m

azure-vote-front-864d77f5ff-bnztt 1/1 Running 0 41s

azure-vote-front-864d77f5ff-jzhbd 1/1 Running 0 42s

azure-vote-front-864d77f5ff-mdxcs 1/1 Running 0 33s

C:\Users\DC-LAPTOP\azure-voting-app-redis>**kubectl get service azure-vote-front**

NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE

azure-vote-front LoadBalancer 10.0.100.207 40.118.238.48 80:32150/TCP 34m

C:\Users\DC-LAPTOP\azure-voting-app-redis>**az aks get-upgrades --resource-group dc1-RGRP --name dc1AKSCluster --query "id" --output tsv**

/subscriptions/dee29fa2-d695-439c-80a5-158ce469882b/resourcegroups/dc1-RGRP/providers/Microsoft.ContainerService/managedClusters/dc1AKSCluster/upgradeprofiles/default

C:\Users\DC-LAPTOP\azure-voting-app-redis>**az aks upgrade --resource-group dc1-RGRP --name dc1AKSCluster --kubernetes-version 1.13.7**

Kubernetes may be unavailable during cluster upgrades.

Are you sure you want to perform this operation? (y/n): y

{

"aadProfile": null,

"addonProfiles": null,

"agentPoolProfiles": [

{

"availabilityZones": null,

"count": 2,

"enableAutoScaling": null,

"maxCount": null,

"maxPods": 110,

"minCount": null,

"name": "nodepool1",

"orchestratorVersion": "1.13.7",

"osDiskSizeGb": 100,

"osType": "Linux",

"provisioningState": "Succeeded",

"type": "AvailabilitySet",

"vmSize": "Standard\_DS2\_v2",

"vnetSubnetId": null

}

],

"apiServerAuthorizedIpRanges": null,

"dnsPrefix": "dc1AKSClus-dc1-RGRP-dee29f",

"enablePodSecurityPolicy": null,

"enableRbac": true,

"fqdn": "dc1aksclus-dc1-rgrp-dee29f-87446ccc.hcp.westus.azmk8s.io",

"id": "/subscriptions/dee29fa2-d695-439c-80a5-158ce469882b/resourcegroups/dc1-RGRP/providers/Microsoft.ContainerService/managedClusters/dc1AKSCluster",

"identity": null,

"kubernetesVersion": "1.13.7",

"linuxProfile": {

"adminUsername": "azureuser",

"ssh": {

"publicKeys": [

{

"keyData": "ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDCQg3PKbTnCZ2IJMPGeJSf1J+btg5BPJoM8m+qTVqlPIS/cAkMGEaA2fV8CMY9YKnCJqPizmSmx/GX0zR4pBSmujAJmz3SWJx3BANpLIDbUfsc115PapvXml4mjx2nqUBdDvDqj1VtNvI8KHL3jR+BnbR+DJTJ9yr3i9jGLfRWSOOLex5s3ipM8vFqnaTrK/dUVxjiEFxpJRx/NqCQXgJGp+nWV/zeYinxSRhO3Grc3CsLiMo1IKpqvpt79mlrNn3pYrUC9CD3Snwy/dvMtsJk3Mx/LplTO0cK+D19vGrhNm16GDEZ4s6q9pyZUVZYw4IW10tdYng9KB2AIClnBbRT"

}

]

}

},

"location": "westus",

"maxAgentPools": 1,

"name": "dc1AKSCluster",

"networkProfile": {

"dnsServiceIp": "10.0.0.10",

"dockerBridgeCidr": "172.17.0.1/16",

"loadBalancerSku": "basic",

"networkPlugin": "kubenet",

"networkPolicy": null,

"podCidr": "10.244.0.0/16",

"serviceCidr": "10.0.0.0/16"

},

"nodeResourceGroup": "MC\_dc1-RGRP\_dc1AKSCluster\_westus",

"provisioningState": "Succeeded",

"resourceGroup": "dc1-RGRP",

"servicePrincipalProfile": {

"clientId": "8346f74f-4904-4c19-abd7-63c7cc42c8a6",

"secret": null

},

"tags": null,

"type": "Microsoft.ContainerService/ManagedClusters",

"windowsProfile": null

}

C:\Users\DC-LAPTOP\azure-voting-app-redis>**az aks show --resource-group dc1-RGRP --name dc1AKSCluster --output table**

Name Location ResourceGroup KubernetesVersion ProvisioningState Fqdn

------------- ---------- --------------- ------------------- ------------------- --------------------------------------------------------

dc1AKSCluster westus dc1-RGRP 1.13.7 Succeeded dc1aksclus-dc1-rgrp-dee29f-87446ccc.hcp.westus.azmk8s.io

Update an application

Make a change to the sample application, then update the version already deployed to your AKS cluster. Make sure that you're in the cloned *azure-voting-app-redis* directory. The sample application source code can then be found inside the *azure-vote* directory. Open the *config\_file.cfg* file with an editor, such as vi or better yet, **Notepad++**:

Notepad++ azure-vote/azure-vote/config\_file.cfg

Change the values for *VOTE1VALUE* and *VOTE2VALUE* to different values, such as colors. The following example shows the updated values:

Copy

# UI Configurations  
TITLE = 'Azure Voting App'  
VOTE1VALUE = 'Blue'  
VOTE2VALUE = 'Purple'  
SHOWHOST = 'false'

Save and close the file.

Update the container image

To re-create the front-end image and test the updated application, use [docker-compose](https://docs.docker.com/compose/). The --build argument is used to instruct Docker Compose to re-create the application image:

**docker-compose up --build -d**

Test the application locally

To verify that the updated container image shows your changes, open a local web browser to [**http://localhost:8080**](http://localhost:8080).

Tag and push the image

Tag the *azure-vote-front* image with the login server name of your ACR registry. Get the login server name with the [az acr list](https://docs.microsoft.com/en-us/cli/azure/acr) command:

**az acr list --resource-group myResourceGroup --query "[].{acrLoginServer:loginServer}" --output table**

Replace <acrLoginServer> with your ACR login server name or public registry hostname, and update the image version to *:v2* as follows:

**docker tag azure-vote-front <acrLoginServer>/azure-vote-front:v2**

Use [docker push](https://docs.docker.com/engine/reference/commandline/push/) to upload the image to your registry. Replace <acrLoginServer> with your ACR login server name.

**docker push <acrLoginServer>/azure-vote-front:v2**

Deploy the updated application

To provide maximum uptime, multiple instances of the application pod must be running. Verify the number of running front-end instances with the [kubectl get pods](https://kubernetes.io/docs/reference/generated/kubectl/kubectl-commands#get) command:

**$ kubectl get pods**

If you don't have multiple front-end pods, scale the *azure-vote-front* deployment as follows:

**kubectl scale --replicas=3 deployment/azure-vote-front**

To update the application, use the [kubectl set](https://kubernetes.io/docs/reference/generated/kubectl/kubectl-commands#set) command. Update <acrLoginServer> with the login server or host name of your container registry, and specify the *v2* application version:

**kubectl set image deployment azure-vote-front azure-vote-front=<acrLoginServer>/azure-vote-front:v2**

To monitor the deployment, use the [kubectl get pod](https://kubernetes.io/docs/reference/generated/kubectl/kubectl-commands#get) command. As the updated application is deployed, your pods are terminated and re-created with the new container image.

**kubectl get pods**

Test the updated application

To view the update application, first get the external IP address of the azure-vote-front service:

**kubectl get service azure-vote-front**

Now open a local web browser to the IP address of your service:

Machine generated alternative text:
52.170ß1.78 
O 

**Upgrade Kubernetes in Azure Kubernetes Service (AKS)**

Get available cluster versions

Before you upgrade a cluster, use the [az aks get-upgrades](https://docs.microsoft.com/en-us/cli/azure/aks#az-aks-get-upgrades) command to check which Kubernetes releases are available for upgrade:

**az aks get-upgrades --resource-group myResourceGroup --name myAKSCluster --output table**

Upgrade a cluster

To minimize disruption to running applications, AKS nodes are carefully cordoned and drained. In this process, the following steps are performed:

1. The Kubernetes scheduler prevents additional pods being scheduled on a node that is to be upgraded.
2. Running pods on the node are scheduled on other nodes in the cluster.
3. A node is created that runs the latest Kubernetes components.
4. When the new node is ready and joined to the cluster, the Kubernetes scheduler begins to run pods on it.
5. The old node is deleted, and the next node in the cluster begins the cordon and drain process.

Use the [az aks upgrade](https://docs.microsoft.com/en-us/cli/azure/aks#az-aks-upgrade) command to upgrade the AKS cluster. The following example upgrades the cluster to Kubernetes version *1.12.13*.

**az aks upgrade --resource-group myResourceGroup --name myAKSCluster --kubernetes-version 1.12.13**

Validate an upgrade

Confirm that the upgrade was successful using the [az aks show](https://docs.microsoft.com/en-us/cli/azure/aks#az-aks-show) command as follows:

**az aks show --resource-group myResourceGroup --name myAKSCluster --output table**

That's it! **Now Clean Up!!!**

**az group delete --name myResourceGroup --yes --no-wait**