**ACR 实验步骤**

**创建资源组**

az group create --name learn-deploy-acr-rg --location southeastasia

**设定默认资源组**

az configure --defaults group=learn-deploy-acr-rg location=southeastasia

**定义ACR名称变量**

ACR\_NAME=chengzhacr0919

**创建ACR**

az acr create --resource-group learn-deploy-acr-rg --name $ACR\_NAME --sku Premium

**创建Dockerfile文件**

**nano Dockerfile**

FROM node:9-alpine

ADD <https://raw.githubusercontent.com/Azure-Samples/acr-build-helloworld-node/master/package.json> /

ADD <https://raw.githubusercontent.com/Azure-Samples/acr-build-helloworld-node/master/server.js> /

RUN npm install

EXPOSE 80

CMD ["node", "server.js"]

**build映像**

az acr build --registry $ACR\_NAME --image helloacrtasks:v1 .

**验证映像**

az acr repository list --name $ACR\_NAME --output table

**启用注册表管理员账户**

az acr update -n $ACR\_NAME --admin-enabled true

**检索管理员账户的用户名和密码并记录**

az acr credential show --name $ACR\_NAME

chengzhacr0919 E9JnEqEWB1cWRbliPR0eL/i3fTmSTKf7

**使用Azure CLI部署容器**

az container create \

--resource-group learn-deploy-acr-rg \

--name acr-tasks \

--image $ACR\_NAME.azurecr.io/helloacrtasks:v1 \

--registry-login-server $ACR\_NAME.azurecr.io \

--ip-address Public \

--location southeastasia \

--registry-username chengzhacr0919 \

--registry-password E9JnEqEWB1cWRbliPR0eL/i3fTmSTKf7

**检测容器地址**

az container show --resource-group learn-deploy-acr-rg --name acr-tasks --query ipAddress.ip --output table

**复制到其他区域**

az acr replication create --registry $ACR\_NAME --location japaneast

**查看映像副本**

az acr replication list --registry $ACR\_NAME --output table

**本地映像的创建和上传**

**克隆代码库**

git clone <https://github.com/Azure-Samples/acr-build-helloworld-node>

**进入代码目录**

cd acr-build-helloworld-node

**build映像**

docker build -t helloacrtasks .

**登录到ACR，用户名和密码前前述步骤**

docker login chengzhacr0919.azurecr.io

**打标签**

docker tag helloacrtasks:latest chengzhacr0919.azurecr.io/localhelloacrtasks:latest

**上传**

docker push chengzhacr0919.azurecr.io/localhelloacrtasks:latest

**ACI实验**

**（可选）创建资源组**

az group create --name earn-deploy-acr-rg --location eastus

**定义容器dns名称变量**

DNS\_NAME\_LABEL=aci-demo-$RANDOM

**创建测试容器，该容器位于hub.docker.com**

az container create \

--resource-group learn-deploy-acr-rg \

--name mycontainer \

--image microsoft/aci-helloworld \

--ports 80 \

--dns-name-label $DNS\_NAME\_LABEL \

--location southeastasia

**查看容器FQDN**

az container show \

--resource-group learn-deploy-acr-rg \

--name mycontainer \

--query "{FQDN:ipAddress.fqdn,ProvisioningState:provisioningState}" \

--out table

**创建任务完成即停止的容器**

az container create \

--resource-group learn-deploy-acr-rg \

--name mycontainer-restart-demo \

--image microsoft/aci-wordcount:latest \

--restart-policy OnFailure \

--location southeastasia

**检查容器状态**

az container show \

--resource-group learn-deploy-acr-rg \

--name mycontainer-restart-demo \

--query containers[0].instanceView.currentState.state

**查看容器日志**

az container logs \

--resource-group learn-deploy-acr-rg \

--name mycontainer-restart-demo

**设置数据名称变量**

COSMOS\_DB\_NAME=aci-cosmos-db-$RANDOM

**创建cosmosdb实例**

COSMOS\_DB\_ENDPOINT=$(az cosmosdb create \

--resource-group learn-deploy-acr-rg \

--name $COSMOS\_DB\_NAME \

--query documentEndpoint \

--output tsv)

**获取cosmosdb连接字符串作为变量**

COSMOS\_DB\_MASTERKEY=$(az cosmosdb keys list \

--resource-group learn-deploy-acr-rg \

--name $COSMOS\_DB\_NAME \

--query primaryMasterKey \

--output tsv)

**创建前端容器**

az container create \

--resource-group learn-deploy-acr-rg \

--name aci-demo \

--image microsoft/azure-vote-front:cosmosdb \

--ip-address Public \

--location southeastasia \

--environment-variables \

COSMOS\_DB\_ENDPOINT=$COSMOS\_DB\_ENDPOINT \

COSMOS\_DB\_MASTERKEY=$COSMOS\_DB\_MASTERKEY

**获取前端容器IP地址**

az container show \

--resource-group learn-deploy-acr-rg \

--name aci-demo \

--query ipAddress.ip \

--output tsv

**显示上一个容器的环境变量**

az container show \

--resource-group learn-deploy-acr-rg \

--name aci-demo \

--query containers[0].environmentVariables

**创建使用安全环境变量的容器**

az container create \

--resource-group learn-deploy-acr-rg \

--name aci-demo-secure \

--image microsoft/azure-vote-front:cosmosdb \

--ip-address Public \

--location southeastasia \

--secure-environment-variables \

COSMOS\_DB\_ENDPOINT=$COSMOS\_DB\_ENDPOINT \

COSMOS\_DB\_MASTERKEY=$COSMOS\_DB\_MASTERKEY

**显示环境变量，并做对比**

az container show \

--resource-group learn-deploy-acr-rg \

--name aci-demo-secure \

--query containers[0].environmentVariables

**创建存储账号命名变量**

STORAGE\_ACCOUNT\_NAME=mystorageaccount$RANDOM

**创建后端存储**

az storage account create \

--resource-group learn-deploy-acr-rg \

--name $STORAGE\_ACCOUNT\_NAME \

--sku Standard\_LRS \

--location southeastasia

**提取存储账号连接信息作为变量**

export AZURE\_STORAGE\_CONNECTION\_STRING=$(az storage account show-connection-string \

--resource-group learn-deploy-acr-rg \

--name $STORAGE\_ACCOUNT\_NAME \

--output tsv)

**创建文件共享**

az storage share create --name aci-share-demo

**获取存储账号密钥作为变量**

STORAGE\_KEY=$(az storage account keys list \

--resource-group learn-deploy-acr-rg \

--account-name $STORAGE\_ACCOUNT\_NAME \

--query "[0].value" \

--output tsv)

**显示密钥**

echo $STORAGE\_KEY

**创建前端容器**

az container create \

--resource-group learn-deploy-acr-rg \

--name aci-demo-files \

--image microsoft/aci-hellofiles \

--location southeastasia \

--ports 80 \

--ip-address Public \

--azure-file-volume-account-name $STORAGE\_ACCOUNT\_NAME \

--azure-file-volume-account-key $STORAGE\_KEY \

--azure-file-volume-share-name aci-share-demo \

--azure-file-volume-mount-path /aci/logs/

**显示前端容器公网地址**

az container show \

--resource-group learn-deploy-acr-rg \

--name aci-demo-files \

--query ipAddress.ip \

--output tsv

**在表单中输入一些文本，并单击“提交”。 此操作将创建一个文件，其中包含在 Azure 文件共享中输入的文本**

**运行此 az storage file list 命令，显示文件共享中包含的文件。**

az storage file list -s aci-share-demo -o table

**运行 az storage file download，将文件下载到 Cloud Shell 会话。 将 <filename> 替换为上一步中出现的文件之一**

az storage file download -s aci-share-demo -p <filename>

**运行 cat 命令以打印文件的内容**

cat <filename>

**创建测试用容器**

az container create \

--resource-group learn-deploy-acr-rg \

--name mycontainer \

--image microsoft/sample-aks-helloworld \

--ports 80 \

--ip-address Public \

--location southeastasia

**查看日志**

az container logs \

--resource-group learn-deploy-acr-rg \

--name mycontainer

**获取容器事件**

az container attach \

--resource-group learn-deploy-acr-rg \

--name mycontainer

**在容器中执行命令**

az container exec \

--resource-group learn-deploy-acr-rg \

--name mycontainer \

--exec-command /bin/sh

ls

exit

**运行以下 az container show 命令以获取 Azure 容器实例的 ID，并将 ID 存储在 Bash 变量中。**

CONTAINER\_ID=$(az container show \

--resource-group learn-deploy-acr-rg \

--name mycontainer \

--query id \

--output tsv)

**运行 az monitor metrics list 命令以检索 CPU 使用率信息。**

az monitor metrics list \

--resource $CONTAINER\_ID \

--metric CPUUsage \

--output table

**运行 az monitor metrics list 命令以检索内存使用率信息。**

az monitor metrics list \

--resource $CONTAINER\_ID \

--metric MemoryUsage \

--output table

**使用web app部署容器化应用**

**克隆代码**

git clone <https://github.com/MicrosoftDocs/mslearn-deploy-run-container-app-service.git>

**进入目录**

cd mslearn-deploy-run-container-app-service/dotnet

**可选：声明ACR命名变量**

ACR\_NAME=chengzhacr0919

**build映像**

az acr build --registry $ACR\_NAME --image webimage .

**查看映像**

az acr repository list --name $ACR\_NAME --output table

**可选：使用ACI进行部署**

az container create \

--resource-group learn-deploy-acr-rg \

--name webimage \

--image $ACR\_NAME.azurecr.io/webimage:latest \

--registry-login-server $ACR\_NAME.azurecr.io \

--ip-address Public \

--location southeastasia \

--registry-username chengzhacr0919 \

--registry-password E9JnEqEWB1cWRbliPR0eL/i3fTmSTKf7

**手工操作使用web app进行部署，**

**查看前端页面展示效果**

**在webapp上启用连续部署**

**进入代码路径**

cd ~/mslearn-deploy-run-container-app-service/dotnet/SampleWeb/Pages

**编辑功能**

mv Index.cshtml Index.cshtml.old

mv Index.cshtml.new Index.cshtml

**切换目录**

cd ~/mslearn-deploy-run-container-app-service/dotnet

**更新映像**

az acr build --registry $ACR\_NAME --image webimage .

**查看前端页面变化**

**实现持续集成**

**创建指向github代码库的task**

az acr task create \

--registry $ACR\_NAME \

--name taskhelloworld \

--image helloworld \

--context <https://github.com/cloudzun/acr-build-helloworld-node.git> \

--file Dockerfile \

--git-access-token 3e439389d9d9bd543582a5a8395bf26a4d03f35a

**手动执行初次build**

az acr task run --registry $ACR\_NAME --name taskhelloworld

**手动创建基于容器的webapp，并启用连续部署**

**在github中对server.js进行编辑，并提交**

**观察正在执行的任务**

az acr task logs --registry $ACR\_NAME

**查看任务列表**

az acr task list-runs --registry $ACR\_NAME --output table

**观察web app页面更新情况**

az container create \

--resource-group DockerRg \

--name text-demo \

--dns-name-label chengzhtxtlab \

--cpu 1 \

--memory 4 \

--ports 5000 \

--image mcr.microsoft.com/azure-cognitive-services/sentiment \

--ip-address Public \

--location southeastasia \

--environment-variables \

Eula=accept \

Billing=https://southeastasia.api.cognitive.microsoft.com/text/analytics/v2.1 \

ApiKey=04dd51503cdc41ab9894d9d6d6c7fd89

**清理资源**

az group delete -n learn-deploy-acr-rg --yes