Inhaltsverzeichnis

Docker und Web-App in Azure 3
Vorbereitungen 3
Ein erster Test 3
Einbinden eines Git Repositories 6

Docker und Web-App in Azure

Ein Selbstversuch...

Vorbereitungen

Ein kostenloses Azure- Konto holen

Installieren der Docker Compose CLI for Linux:

```
curl -L
https://raw.githubusercontent.com/docker/compose-cli/main/scripts/install/in
stall_linux.sh | sh
```

Anlegen eines Bastelverzeichnisses:

```
mkdir -p ~/Playground/docker_Azure
cd ~/Playground/docker_Azure
```

Verbinden von Docker und Azure

```
docker login azure
```

Erstellen eines ACI- Kontests

```
docker context create aci murmel
```

Prüfen, ob's geklappt hat

```
docker context ls
```

Ein erster Test

In den Kontext wechseln

```
docker context use murmel
```

ein öffentliches Image starten

```
docker run -p 80:80 mcr.microsoft.com/azuredocs/aci-helloworld
```

Nach dem Lauf kann man den Container mit `docker rm` wieder entfernen - um das Image braucht man sich keine Sorgen zu machen, denn ich hab's versucht zu löschen, aber die Fehlermeldung erinnerte mich daran, das die Cloud keine Images speichert (muss man die dann für jeden Run erst wieder hochladen??)

Nächster Versuch: Wir basteln uns ein Zip- Image

Dockerfile

```
FROM javieraviles/zip
CMD ["zip", "--help"]
```

```
docker build -t zip .
docker run --rm zip
```

Als nächstes packen wir ein Verzeichnis

Dockerfile

```
FROM javieraviles/zip
CMD ["zip", "-r" , "/zip/out.zip", "/etc"]
```

```
docker build -t zip .
docker run -v $(pwd):/zip --rm zip
```

Jetzt versuchen wir mal, eine Azure Datei Freigabe als Ziel- Verzeichnis in einen Azure docker container einzuhängen...

Das Azure CLI installieren

```
sudo apt install azure-cli
```

Erstellen der Freigabe [1)]

```
az login
```

Die notwendige Resourcengruppe erzeugen

```
export ACI_PERS_RESOURCE_GROUP=murmel
az group create -l northeurope -n $ACI_PERS_RESOURCE_GROUP
```

und nun legen wir innerhalb der Resourcengruppe eine Azure Container Registry an, in der man dann die eigenen Container speichert

```
export ACR_PERS_NAME=murmelimages
az acr create --resource-group $ACI_PERS_RESOURCE_GROUP --name
$ACR_PERS_NAME --sku Basic
```

create_share.sh

```
# Change these four parameters as needed
export ACI_PERS_RESOURCE_GROUP=murmel
```

```
export ACI_PERS_STORAGE_ACCOUNT_NAME=steffen$RANDOM
export ACI_PERS_LOCATION=northeurope
export ACI_PERS_SHARE_NAME=murmelshare

# Create the storage account with the parameters
az storage account create \
    --resource-group $ACI_PERS_RESOURCE_GROUP \
    --name $ACI_PERS_STORAGE_ACCOUNT_NAME \
    --location $ACI_PERS_LOCATION \
    --sku Standard_LRS

# Create the file share
az storage share create \
  --name $ACI_PERS_SHARE_NAME \
  --account-name $ACI_PERS_STORAGE_ACCOUNT_NAME
```

```
az login
./create_share.sh
```

Erhalten der Anmeldeinformationen für das Speicherkonto[2)] Um eine Azure-Dateifreigabe als Volume in Azure Container Instances einzubinden, benötigen Sie drei Werte: den Namen des Speicherkontos, den Freigabenamen und den Speicherzugriffsschlüssel.

Speicherkontoname: Wenn Sie das obige Skript verwendet haben, wurde der Name des Speicherkontos in der Variablen $ACI_PERS_STORAGE_ACCOUNT_NAME gespeichert. Um den Kontonamen einzusehen, geben Sie Folgendes ein:

```
echo $ACI_PERS_STORAGE_ACCOUNT_NAME
```

Freigabename: Dieser Wert ist bereits definiert als $ACI_PERS_SHARE_NAME.

Speicherkontoschlüssel: Diesen Wert finden Sie mithilfe des folgenden Befehls:

```
export STORAGE_KEY=$(az storage account keys list --resource-group
$ACI_PERS_RESOURCE_GROUP --account-name \
$ACI_PERS_STORAGE_ACCOUNT_NAME --query "[0].value" --output tsv)
```

echo $STORAGE_KEY

Zuerst bauen wir das Image lokal mit dem passenden Tag für die ACR. Der Tag setzt sich zusammen aus `murmelimages+ .azurecr.io/`

Dockerfile

```
FROM javieraviles/zip
CMD ["zip", "-r" , "/zip/out.zip", "/etc"]
```

```
docker build -t $ACR_PERS_NAME.azurecr.io/zip:v1 .
```

anmelden an den ACR

```
az acr login --name $ACR_PERS_NAME
```

image pushen

```
docker push $ACR_PERS_NAME.azurecr.io/zip:v1
```

den Container starten

```
az container create \
 --resource-group $ACI_PERS_RESOURCE_GROUP \
 --name $ACR_PERS_NAME \
 --image $ACR_PERS_NAME.azurecr.io/zip:v1 \
 --azure-file-volume-account-name $ACI_PERS_STORAGE_ACCOUNT_NAME \
 --azure-file-volume-account-key $STORAGE_KEY \
 --azure-file-volume-share-name $ACI_PERS_SHARE_NAME \
 --azure-file-volume-mount-path /zip \
 --registry-username $ACR_PERS_NAME  \
 --registry-password <registry-password>
```

Die erzeugte Datei findet sich dann in der $ACI_PERS_SHARE_NAME Dateifreigabe

Einbinden eines Git Repositories

Zuerst wird der Aufruf so anpasst, das er auf das Eingaben- Verzeichnis zeigt, was dann als git eingebunden werden soll

Dockerfile

```
FROM javieraviles/zip
CMD ["zip", "-r" , "/zip/git.zip", "/git"]
```

```
docker build -t $ACR_PERS_NAME.azurecr.io/git:v1 .
```

anmelden an den ACR

```
az acr login --name $ACR_PERS_NAME
```

image pushen

```
docker push $ACR_PERS_NAME.azurecr.io/git:v1
```

Das Ganze ist ähnlich wie der „normale“ Aufruf, aber mit zusätzlichen Parametern:

- –gitrepo-url: Der gewünschte Repository- Link
- –gitrepo-mount-path : Wohin das Rep im container gemountet werden soll

- –gitrepo-dir: Optional das Verzeichnis innerhalb des zu klonenden Volumes
- –gitrepo-revision: Optional der Commithash der zu klonenden Revision

den Container starten

```
az container create \
 --resource-group $ACI_PERS_RESOURCE_GROUP \
 --name $ACR_PERS_NAME \
 --image $ACR_PERS_NAME.azurecr.io/git:v1 \
 --azure-file-volume-account-name $ACI_PERS_STORAGE_ACCOUNT_NAME \
 --azure-file-volume-account-key $STORAGE_KEY \
 --azure-file-volume-share-name $ACI_PERS_SHARE_NAME \
 --azure-file-volume-mount-path /zip \
 --registry-username $ACR_PERS_NAME  \
 --registry-password <registry-password>\
 --gitrepo-url https://github.com/stko/valuepipe.git \
 --gitrepo-mount-path /git
```

1)
https://docs.microsoft.com/de-de/azure/container-instances/container-instances-volume-azure-files#create-an-azure-file-share
2)
https://docs.microsoft.com/de-de/azure/container-instances/container-instances-volume-azure-files#get-storage-credentials

From:
http://koehlers.de/wiki/ - **Steffen Köhlers Online- Bastelbuch**

Permanent link:
http://koehlers.de/wiki/doku.php?id=pc:azuredocker

Last update: **2022/05/22 10:49**