

Eine Web App über Azure bereitstellen

Ziel ist es, eine vorher entwickelte Webanwendung über eine Web App nach dem Prinzip *Continuous Integration / Continuous Delivery* bereitzustellen. Zur Bereitstellung wird die eine *Container Registry* in **Azure** verwendet.

Container Registry anlegen

- Loggen Sie sich bei Azure ein.
- Erstellen Sie eine neue Ressource vom Typ *Container Registry*. Verwenden Sie beim Anlegen eine neue Ressourcengruppe **tls-anr-ita-crg**. Nutzen Sie als Bereitstellungsort nach Möglichkeit **Germany West Central**. Der Name der Registry kann frei gewählt werden; im Folgenden wird **tlscontainerhafen** verwendet.
- Öffnen Sie eine **Cloud Shell** (*PowerShell* oder *bash*) und klonen Sie den vorbereiteten Source Code aus Ihrem Repository, das die Webanwendung enthält in Ihren Cloud Shell Space. Sollten Sie keinen eigenen Source Code besitzen, verwenden Sie zum Testen die im Beispiel angegebene URL:

```
PS > git clone https://github.com/JAngres/Docker-Apache.git
```

Container bauen und hochladen

Nach dem Klonen des Source Codes kann der Container gebaut in die angelegte Container Registry hochgeladen werden. Das so erzeugte Image wird mit einem *Tag* versehen, um es später identifizieren zu können. Hier wird der Tag **awesome-webapp** verwendet. Beachten Sie den Punkt als letztes Argument für den Befehl. Im ersten Schritt muss *Docker-Apache* ggf. durch den Namen Ihres geklonten Repositorys ersetzt werden.

```
PS > sl Docker-Apache
PS > az acr build --registry tlscontainerhafen --image awesome-webapp .
```

Web App konfigurieren

- Erstellen Sie eine neue Ressource vom Typ *Web App*.
- Fügen Sie sie zur Ressourcengruppe **tls-anr-ita-crg** hinzu.
- Wählen Sie einen Namen für die URL der App. Im Beispiel wird **webapp-anr4tw** verwendet.
- Wählen Sie beim Veröffentlichen den Punkt *Docker Container*.
- Belassen Sie die übrigen Einstellungen wie vorgeschlagen und gehen Sie zu **Weiter: Docker**.
- Im Tab *Docker* wählen Sie **Azure Container Registry** als Quelle und wählen Sie Ihre Registry (**tlscontainerhafen**) aus.
- Die übrigen Felder sollten automatisch befüllt werden.
- Erstellen Sie die Web App-Ressource und gehen Sie nach der Erstellung zur Seite der Ressource.
- In der oberen Menüleiste klicken Sie auf **Durchsuchen**. (Ist etwas merkwürdig übersetzt. Im Englischen steht dort **Browse** und gemeint ist, sich die Web App im Browser anzuschauen.) Da die App beim

ersten Zugriff hochgefahren werden muss, dauert es ein wenig, bis die Anwendung geladen ist. Sobald der Container läuft, sind alle weiteren Zugriffe (auch bei Änderungen) jedoch quasi in Echtzeit möglich.

- Lassen Sie den Tab mit der Webanwendung geöffnet.

Continuous Delivery über Webhook

- Gehen Sie in der Übersichtsseite für die Web App in Azure zu **Bereitstellungscenter**. Setzen Sie den Radio Button bei *Continuous deployment* auf **On** und klicken Sie in der oberen Menüleiste auf *Speichern*.

Neue Version entwickeln und updaten

- Öffnen Sie wieder die Azure Cloud Shell und navigieren Sie in einen Ordner, in dem eine Quelltextdatei liegt. Im Beispiel ist es die *index.html* Datei der Webanwendung.

```
PS > sl ~/Docker-Apache
PS > code index.html
```

- Verändern Sie die Datei im Editor, indem Sie die Versionsnummer hinaufzählen. Speichern Sie die Datei anschließend, indem Sie rechts oben in die Ecke des Editors klicken und *Speichern* auswählen.
- Bauen Sie anschließend die neue Version der Anwendung.

```
PS > az acr --registry tlscontainerhafen --image awesome-webapp .
```

- Navigieren Sie in Azure zu Ihrer Container Registry und wählen Sie im linken Menü *Webhooks* im Untermenü *Dienste*. Hier sollte nun ein Webhook zu sehen sein, der auf das gerade erzeugte, geupdatete Image der Anwendung hinweist bzw. zeigt.
- Gehen Sie zum geöffneten Tab, in dem die Anwendung läuft. Laden Sie den Tab neu (Taste *F5*). Die neue Versionsnummer sollte nun angezeigt werden, da über den Webhook automatisch das aktualisierte Image angezogen wird.