Javascript Web-Applikation als Docker Container in der Cloud verfügbar machen

**Ausgangslage**

**Sie haben eine fertige Node JS-Webapplikation lokal auf ihrem Gerät und wollen diese nun in der Cloud verfügbar machen.**

**Zielgruppe  
  
Lernende des dritten Lehrjahres Abteilung Informatik**

**Voraussetzungen**

* **Docker installiert**
* **Azure konfiguriert**
* **Kenntnisse mit Azure und Docker**

**Was ist nicht Bestandteil dieser Anleitung**

* **Auf Konfiguration von Docker und Azure Cloud wird nicht näher eingegangen**
* **Intergration der Datenbank ist nicht Bestandteil dieser Anleitung**
* **Port forwarding**

**Bitte beachten**

* ***Kursiv teile in den Befehlen* = Stehen für individuelle Teile, dies hängt davon ab, wie man seine Objekte benannt an. z.B. mein Server heisst vielleicht anders als deiner.**
* **Einstellungen, die nicht im Bild zu sehen sind, wurden immer auf der Standardeinstellung gelassen**
* **Es wurde kein Portforwarding erstellt. D.H der korrekte Port muss angegeben werden beim Aufrufen über den Browser co.**

**Anleitung**

|  |  |  |
| --- | --- | --- |
| ID | Beschreibung | ****Grafik**** |
| 1 | Azure 🡪 Icon Ressourcengruppen anwählen und dann auf Knop „Erstellen“ klicken |  |
| 2 | Spezifikationen eingeben und erstellen |  |
| 3 | Containerregistrierungen -> erstellen -> Spezifikationen eingeben und erstellen klicken |  |
| 4 | Ins Verzeichnis der Webapp navigieren 🡪 File Namens „dockerfile“ erstellen 🡪 Inhalt vom Bild einfügen. Achtung eventuell haben sie einen anderen Port definiert als ich. In diesem Fall ihren Port bei EXPOSE angeben. |  |
| 5 | Im selben Verzeichnis File Namens „.dockerignore“ erstellen 🡪 Inhalt vom Bild einfügen. |  |
| 6 | Sicherstellen, dass im File Pacakge.json unter scripts “start“: “node index.js“ definiert ist. |  |
| 6 | Im Terminal zum Ordner der Webapplikation navigieren |  |
| 7 | Im Terminal den Befehl „docker build -t a*nmeldeserver/registrierungsname“* Die Daten dazu sind auf azure unter Einstellungen 🡪 Zugriffschlüssel zu finden (beim Befehl alles Lowercase). |  |
| 8 | Überprüfen, ob das Image erstellt wurde mit dem Befehl „docker image ls“ |  |
| 9 | Im Anmeldeserver einloggen. |  |
| 10 | Nun muss im Terminal der Username und das Passwort eingegeben werden. Sie finden dies unter Zugriffsschlüssel nachdem sie Administratorbenutzer aktiviert haben. |  |
| 11 | Das Docker Image mit dem Befehl „docker push *repository“* hochladen |  |
| 12 | Um zu kontrollieren ob das Pushen erfolgreich war, auf azure navigieren 🡪 Containerregistrierung 🡪 Dienste 🡪 Repositorys 🡪 auf Repository klicken. |  |
| 13 | Azure 🡪 Containerinstanzen 🡪 Erstellen 🡪 Spezifikationen eingeben 🡪 erstellen   Hier muss als Imagequelle Azure Container Registry gewählt werden, weil wir dort unser Repository gepusht haben. Als Port den vorher definierten Port eingeben . |  |
| 14 | Azure 🡪 Ressourcegruppen 🡪 vorher erstellte Containerinstanz wählen (in meinem Fall: dockertestwebappcontainer) Hier kann der Webapp Container verwaltet werden |  |
| 15 | Nun kann die Webapp im Browser angerufen werden 🡪 FQDN kopieren und Port hinzufügen. |  |