# Exam Preparation

Problems for the exam preparation for the ["Containers and Clouds" course @ SoftUni](https://softuni.bg/trainings/4117/containers-and-cloud-may-2023).

Your task is to deploy an app to Azure via Docker or Terraform. In order for your to complete successfully the exam, you must choose one of the two following tasks.

## Deploy an ASP.NET Core MVC app to Azure via Docker

You are provided with a .NET application that consists of two projects – one for the **web application** and one for the **SQL Server database**.

### Steps

Your task is to deploy the app to Azure via Docker by executing the following steps:

#### Build a Custom Image (25 pts.)

Create a **Dockerfile** in the **web app root directory** in the solution. The Dockerfile must contain **four** stages:

##### Base

This is the initial stage. Use the "**mcr.microsoft.com/dotnet/aspnet:6.0**" image. Expose ports **80** and **443** and set the working directory to **/app**.

##### Build

Use the "**mcr.microsoft.com/dotnet/sdk:6.0**" image. Set the working direcroty to **/src**. You should copy the **Library.WebApp** and **Library.Data** projects into the container. Restore the packages, required by the **Library** application. Copy all of the files into the current working directory. Build the application in **Release** mode. Don't forget to set the directory for the build artifacts.

##### Publish

Publish the application in **Release** mode. Set the working directory to **/app/publish**. Don't forget to exclude the **AppHost** when publishing.

##### Final

Set the working directory to **/app**. Copy the output from the "**Publish**" stage. Set the entrypoint for the container and the executable file.

**NOTE: Feel free to use the built-in Docker support of Visual Studio.**

#### Orchestrate Containers (25 pts.)

Create a **docker-compose.yaml** file. It should contain the version of the file, definitions of the **two services**: for the **database** and for the **web** **app** and **volumes** definition.

##### Database service

The file should contain:

* **container name**
* **image**
* **exposed ports: 1433:1433**
* **deployment configuration**
  + **resource reservations**
    - **CPUs: 2**
    - **Memory: 2GB**
* **environment variables**
* **volumes**

##### Web app service

The file should contain:

* **container name**
* **build context**
  + the **dockerfile path** for building the container
* **image**
* **exposed ports: 80:80**
* **restart policy**

##### Volumes

* Volume: **sqldata**
  + **Volume driver to use**
  + Additional volume driver options
    - **Share name**: **sql-volume**
    - **Storage account name: librarystorageacc**

#### Create Azure Container Registry (10 pts.)

Create a resource group and a container registry.

##### Resource group

Name: **{username}libraryrg**

##### Container registry

Name: **{username}librarycr**

#### Push Image to Azure Container Registry (10 pts.)

Push the image to your Azure container registry.

#### Create Azure Context (10 pts.)

Create an ACI context to associate Docker with your Azure subscription and resource group.

#### Deploy App to to Azure Container Instances (10 pts.)

Start the application in the Azure Container Instance.

#### Run the App in Azure (10 pts.)

Run the IP in Azure using the IP address of the application.

### Requirements

Provide the **Dockerfile** and the **docker-compose.yaml** files.

Provide **images** of the **Resource Group**, the **Container registry** and the **Container Instances** from Azure Portal, and from the running in a browser app.

Modify your **app domain** to visualize your **SoftUni** username.

### Submission

You can submit your solution in **one** of the following ways **(chose whichever is more convenient for you):**

* Add the **dockerfile**, **docker-compose.yaml** and the **images** in **one .ZIP** file and **submit** it to the **SULS** system;
* **Paste** the **images** in a **Word** document. Add the **dockerfile**, **docker-compose.yaml** and **.docx** file to **one .ZIP** file and **submit** it to the **SULS** system.

## Deploy an ASP.NET Core MVC app to Azure via Terraform

You are provided with a .NET application that consists of two projects – one for the **web application** and one for the **SQL Server database**.

### Steps

Your task is to deploy the app to Azure via Terraform by executing the following steps:

#### Create Azure Resource Group (25 pts.)

Create a **Terraform** configuration to deploy an **Azure** **resource** **group**.

#### Create App Service Plan (25 pts.)

Configure the Terraform configuration file.

#### Write and Apply a Terraform Configuration (25 pts.)

Configure the Terraform configuration file.

#### Separate Configuration to Multiple Files (20 pts.)

Separate the Terraform configuration file to multiple files:

##### main.tf

This should be the main Terraform configuration file.

##### variables.tf

This file should contain the variable declarations.

##### values.tfvars

This file should contain the values for the variables.

##### outputs.tf

This file should contain the output declarations.

#### Apply Configuration (5 pts.)

Deploy the app.

### Requirements

Provide the Terraform configuration files and an image of the deployed app.

### Submission

Add the configuration files and the image to one .ZIP file and submit it to SULS.