Ejercicio 2: Docker Compose

- Descripción de la práctica
- Servicios de Azure empleados
- Elementos necesarios para la realización de la práctica
  - Creación del grupo de recursos
  - Creación del Azure Container Registry
  - Imágenes Docker utilizadas
  - Fichero docker-compose.yml
  - Despliegue: creación y configuración del App Service

Descripción de la práctica

Esta práctica consiste en el despliegue, utilizando Docker Compose, de una aplicación multicontenedor. La aplicación está compuesta por dos servicios, una base de datos SQL y un *front-end* expuesto públicamente. Estos recursos serán desplegados en Microsoft Azure, empleando para ello los servicios necesarios que nos ofrece la plataforma.

Servicios de Azure empleados

Los servicios de Azure necesarios para esta práctica son los siguientes:

- Azure App Service, para desplegar la aplicación dockerizada utilizando docker-compose (esta funcionalidad todavía se encuentra en una versión preliminar). Este servicio obliga a la creación de un Azure App Service plan en caso de que no exista un plan creado con anterioridad.
- Azure Container Registry, para almacenar las imágenes Docker necesarias para el despliegue
- Azure Resource Groups, para agrupar todos los recursos necesarios para la realización de la práctica.

Se ha utilizado Azure CLI en local. Si se desea, en su lugar, se puede utilizar el servicio Azure CloudShell.

Elementos necesarios para la realización de la práctica

Creación del grupo de recursos

Para agrupar los recursos que conformarán esta práctica, se crea un grupo de recursos con el siguiente comando:

```
az group create --name ics-ej2 --location "West Europe"
```

Creación de un grupo de recursos

Creación del Azure Container Registry

Para almacenar las imágenes personalizadas, se crea un *container registry*. Para crear uno, se ejecutan los siguientes comandos:

```
az acr create --name icsej2containeregistry --resource-group ics-ej2 --sku Basic
```

Creacion del Azure Container Registry

Además, se habilita el usuario admin con el siguiente comando:

```
az acr update -n icsej2containeregistry --admin-enabled true
```

Habilitado el usuario admin

Imágenes Docker utilizadas

Las imágenes Docker utilizadas como base para está práctica, obtenidas de Docker Hub, son las siguientes:

- mysql:5.7
- wordpress:latest

Se ha generado un tag para cada una de las imágenes, de forma que ahora apunten al ACR generado, y se han subido las imágenes al mismo. Los comandos para generar las imágenes y subirlas al ACR son las siguientes:

```
docker pull mysql:5.7
docker pulk wordpress:latest

# Comando interactivo, hay que añadir las credenciales necesarias
# Home > Resource groups > ics-ej2 > icsej2containeregistry
# Username: icsej2containeregistry
# Password: incluir el contenido de password o password2
docker login icsej2containeregistry.azurecr.io

docker tag mysql:5.7 icsej2containeregistry.azurecr.io/mysql:5.7
docker tag wordpress:latest icsej2containeregistry.azurecr.io/wordpress:latest
docker push icsej2containeregistry.azurecr.io/mysql:5.7
docker push icsej2containeregistry.azurecr.io/wordpress:latest
```

Obtención de Docker Hub, generación del tag y subida a ACR de las imágenes Docker necesarias para la práctica

Imágenes subidas al ACR

Fichero docker-compose.yml

```
---
services:
  mysql:
    image: icsej2containeregistry.azurecr.io/mysql:5.7
```

```
    restart: always
    volumes:
      - ${WEBAPP_STORAGE_HOME}/site/data:/var/lib/mysql
    environment:
      MYSQL_ROOT_PASSWORD: toor
      MYSQL_DATABASE: ics-ej2
      MYSQL_USER: dfr
      MYSQL_PASSWORD: dfr

  wordpress:
    image: icsej2containeregistry.azurecr.io/wordpress:latest
    depends_on:
      - mysql
    ports:
      - "80:80"
    restart: always
    volumes:
      - ${WEBAPP_STORAGE_HOME}/site/wwwroot:/var/www/html
    environment:
      WORDPRESS_DB_HOST: mysql:3306
      WORDPRESS_DB_NAME: ics-ej2
      WORDPRESS_DB_USER: dfr
      WORDPRESS_DB_PASSWORD: dfr
```

El fichero `docker-compose.yml` se compone de dos servicios, `mysql` y `wordpress`, para configurar la BBDD y la página web, respectivamente.

En el servicio de `wordpress` se ha abierto el puerto 80 para poder acceder a la página web. Además, se ha incluido la directiva `depends_on` para controlar el orden de inicio (`mysql -> wordpress`) y apagado (`wordpress -> mysql`) de los servicios

Se han configurado dos volúmenes para persistir los datos de la BBDD y los contenidos de la página web. Se puede ver, a través de un cliente FTP, los datos que se están persistiendo. Además de esto, se debe añadir la configuración `WEBSITES_ENABLE_APP_SERVICE_STORAGE` (ver la sección Creación y configuración del App Service) a `true` en el App Service

Acceso al sistema de ficheros del App

Ambos servicios cuentan con la siguiente configuración de entorno: - `mysql`: + `MYSQL_ROOT_PASSWORD`: contraseña del usuario `root` de la BBDD + `MYSQL_DATABASE`: nombre de la BBDD inicial, que se utiliza para la página web + `MYSQL_USER`: usuario de mysql + `MYSQL_PASSWORD`: contraseña del usuario `MYSQL_USER`

- `wordpress`:
  - `WORDPRESS_DB_HOST`: dirección de la BBDD utilizada. En el contexto del docker-compose, se puede utilizar el nombre del servicio de BBDD

(`mysql`) como referencia. El puerto por defecto de MySQL es el 3306.
- `WORDPRESS_DB_NAME`: nombre de la BBDD. Se utiliza la definida en `MYSQL_DATABASE`
- `WORDPRESS_DB_USER`: usuario de la BBDD. Debe ser el mismo que el definido en `MYSQL_USER`
- `WORDPRESS_DB_PASSWORD`: contraseña del usuario `WORDPRESS_DB_USER`

Despliegue: creación y configuración del App Service

Para la creación del App Service se requiere un App Service Plan, que define el tipo de recursos utilizados para la ejecución de las aplicaciones. Los comandos para crear el App Service y su plan asociado son:

```
az appservice plan create --name ASP-icsej2-a972 --resource-group ics-ej2 --sku S1 --is-linu
az webapp create --resource-group ics-ej2 --plan ASP-icsej2-a972 --name ics-dfr --multiconta
```

En las siguientes imágenes se crea tanto el App Service Plan como el App Service asociado a dicho plan.

Creación del App Service Plan

Creación del App Service

Para configurar el acceso a ACR, se deben añadir las credenciales de acceso como configuración al App Service. Para ello, se ejecutarán los siguientes comandos:

```
az webapp config appsettings set --resource-group ics-ej2 --name ics-dfr --settings DOCKER_R
az webapp config appsettings set --resource-group ics-ej2 --name ics-dfr --settings DOCKER_R
az webapp config appsettings set --resource-group ics-ej2 --name ics-dfr --settings DOCKER_R
```

Debido a una issue con este servicio, para configurar la persistencia, se debe de añadir la variable `WEBSITES_ENABLE_APP_SERVICE_STORAGE` en la configuración y fijar su valor a `true`, ya que esta no está definida por defecto. Para añadir dicha configuración se ejecuta el siguiente comando:

```
az webapp config appsettings set --resource-group ics-ej2 --name ics-dfr --settings WEBSITES
```

La configuración final del App Service es la siguiente:

Comando para fijar la variable WEBSITES_ENABLE_APP_SERVICE_STORAGE

Esta es la vista desde la interfaz web de la configuración de

Configuración del App Service

NOTA: la variable `WEBSITE_HEALTHCHECK_MAXPINGFAILURES` no es necesaria. Se incluye automáticamente al configurar el healthcheck de la aplicación. Adicionalmente, la siguiente imagen muestra dicha configuración del *healthcheck* para este App Service. Se puede ver que el valor de la variable coincide con el valor de *Load balancing threshold*

Healthcheck del App Service

Para monitorizar el despliegue de la aplicación, es posible consultar los logs que Docker Engine genera. Es posible consultarlos desde la interfaz web (App Service -> Deployment -> Deployment Center -> Logs)

(Añadir foto dos logs da última ejecución cando haxa que entregar)

Finalmente y una vez ha terminado el despliegue, la primera vez que accedamos a la dirección donde se encuentra alojada la aplicación, se iniciará el proceso de instalación de Wordpress. Como se han configurado los datos de conexión con la BBDD en el docker-compose, esa parte de la instalación no será necesario realizarla.