Tutorial azure para crear contenedores en kubernetes con docker

Juan Casado Ballesteros

Descargar los recursos del tutorial

El contendido del tutorial se encuentra en

```
git clone https://github.com/Azure-Samples/azure-voting-app-redis.git
```

Inciar localmente mediante docker-compose

Utilizaremos la herramienta docker-compose para crear una instancia local del contido descargado. La información sobre el contenido que vamos a crear se encuentra en el archivo **docker-compose.yaml**. Previamente habremos debido de haber iniciado el daemon de docker.

```
docker-compose up -d
```

Se descargarán los contenedores de: 'azure_example_tutorial_default' y 'tiangolo/uwsgi-nginx-flask:python3.6'. Desdede pip se intalará redis.

Comprobamos que se haya creado todo correctamente.

```
docker images
docker ps
```

Podremos acceder a la instancia local desde *http://localhost:8080*.

Finalizar la instancia local

Una vez hayamos comprobado que todo funciona correctamente podremos finalizar la instancia local mediante.

```
docker-compose down
```

Subir la imagen a azure

Para poderlo hacer deberemos tener instalado azure *az*, podremoc omprobarlo mediante

```
az --version
```

Crear un container registry

- Crear recursos de grupo
- Crear una instancia de Azure Container Registry
- Entrar en en el Azure Container Registry

```
az group create --name <resourceGroup> --location eastus
az acr create --resource-group <resourceGroup> --name <resource> --sku Basic
az acr login --name <resource>
```

Preparar la imagen y subirla

Debemoc proporcionar un tag apropiado a las imágenes que queramos subir pues el tag es usado para redireccionarlas correctamente a la ubicación que deseemos. Debemos obtener el loginServer lo cual lo podemos hacer mediante:

```
az acr list --resource-group <resourceGroup> \
--query "[].{acrLoginServer:loginServer}" --output table
```

Utilizamos ahora el login server para poporcionar el tag corrspopndiente a la imagen:

```
docker tag azure-vote-front <resourceLogin>
```

Una vez la imagen tenga el tag podremos subir la a azure:

```
docker push <resourceLogin>
```

Podremos comprobar que la imagen ha sido correctamente subida a Auzere. Para ello debemos preguntar al container registry (no al grupo). La hemos subido como v1 lo cual iondica la versión de la imagen.

```
az acr repository list --name <resource> --output table
az acr repository show-tags --name <resource> \
--repository azure-vote-front --output table
```

Crear un cluster de kubernetes

Utilizaremos el Azure Active Directory para indicar al cluster de kubernetes con qué recursos de azure puede interactuar. Crearemos un service principal tras lo

que nos proporcionará una cuenta cuya información deberemos guardar.

```
az ad sp create-for-rbac --skip-assignment
```

Proporcionar permiso al cluster

Primero obtendremos la ID de nuestro recurso y posteriormente la utilizamos para dar permisos sobre él a la aplicación.

```
az acr show --resource-group <resourceGroup> \
--name <resource> --query "id" --output tsv
az role assignment create --assignee <appId> \
--scope <resourceId> --role acrpull
```

Crear el cluster

Crearemos un cluster de kubernetes y nos conectaremos a él

```
az aks create \
--resource-group <resourceGroup> \
--name <clusterName> \
--node-count 1 \
--service-principal <appId> \
--client-secret <password> \
--generate-ssh-keys
az aks install-cli
az aks get-credentials --resource-group <resourceGroup> \
--name <clusterName>
kubectl get nodes
```

Con *kubectl get nodes* podremos comprobar que el proceso se haya realizado correctamente

Iniciar la aplicación

Lanzaremos públicamente la aplicación para que se pueda acceder a ella

Actualizar el manifest de kubernetes

Debemos incluir el recurso en el manifiesto de kubernetes para lo que utilizaremos el resourceLogin el cual deberemos de incluir en **azure-vote-all-in-one-redis.yaml** en el apartado de image de modo que quede como:

```
image: <resourceLogin>.azurecr.io/azure-vote-front:v1
```

Iniciar la aplicación

Iniciaremos la aplicación desde kubernetes. Esto hará pública nuestra aplicación.

```
kubectl apply -f azure-vote-all-in-one-redis.yaml
```

Probar la aplicación

Podemos monitorizar la aplicación, al lanzar el comando obtendremos también la IP desde la que podremos acceder e ella.

```
kubectl get service azure-vote-front --watch
```

También podremos ver el estado de los contenedores que hayamos lanzado con kubernetes.

```
kubectl get pods
```

Escalar la aplicación