

Despliegues parametrizados: Variables de entorno

Para añadir alguna configuración específica a la hora de lanzar un contenedor, se usan variables de entorno del contenedor cuyos valores se especifican al crear el contenedor para realizar una configuración concreta del mismo.

Por ejemplo, si estudiamos la documentación de la imagen `mariadb` en [Docker Hub](https://hub.docker.com/_/mariadb), podemos comprobar que podemos definir un conjunto de variables de entorno como `MYSQL_ROOT_PASSWORD`, `MYSQL_DATABASE`, `MYSQL_USER`, `MYSQL_PASSWORD`, etc., que nos permitirán configurar de alguna forma determinada nuestro servidor de base de datos (indicando la contraseña del usuario root, creando una determinada base de datos o creando un usuario con una contraseña por ejemplo).

De la misma manera, al especificar los contenedores que contendrán los Pods que se van a crear desde un Deployment, también se pondrán inicializar las variables de entorno necesarias.

Configuración de aplicaciones usando variables de entorno

Vamos a hacer un despliegue de un servidor de base de datos mariadb. Si volvemos a estudiar la documentación de esta imagen en [Docker Hub](https://hub.docker.com/_/mariadb) comprobamos que obligatoriamente tenemos que indicar la contraseña del usuario root inicializando la variable de entorno `MYSQL_ROOT_PASSWORD`. El fichero de despliegue que vamos a usar es

[` mariadb-deployment-env.yaml `](files/mariadb-deployment-env.yaml), y vemos el fragmento del fichero donde se define el contenedor:

```
` ``yaml
...
spec:
  containers:
    - name: contenedor-mariadb
      image: mariadb
      ports:
        - containerPort: 3306
          name: db-port
      env:
        - name: MYSQL_ROOT_PASSWORD
          value: my-password
` ``
```

En el apartado `containers` hemos incluido la sección `env` donde vamos indicando, como una lista, el nombre de la variable (`name`) y su valor (`value`). En este caso hemos indicado la contraseña `my-password`.

Vamos a comprobar si realmente se ha creado el servidor de base de datos con esa contraseña del root:

```
kubectl apply -f mariadb-deploymen-env.yaml
```

```
kubectl get all
```

...

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
deployment.apps/mariadb-deployment	1/1	1	1	5s

kubectl exec -it deployment.apps/mariadb-deployment -- mysql -u root -p

Enter password:

...

MariaDB [(none)]>