 # Despliegues parametrizados: Secrets

Cuando en un variable de entorno indicamos una información sensible, como por ejemplo, una contraseña o una clave ssh, es mejor utilizar un nuevo recurso de Kubernetes llamado Secret.

Los [Secrets](https://kubernetes.io/docs/concepts/configuration/secret/) permiten guardar información sensible que será **codificada** o **cifrada**.

Hay distintos tipos de Secret, en este curso vamos a usar los genéricos y los vamos a crear a partir de un literal. Por ejemplo para guardar la contraseña del usuario root de una base de datos, crearíamos un Secret de la siguiente manera:

```
kubectl create secret generic mariadb --from-literal=password=my-password
```

Podemos obtener información de los Secret que hemos creado con las instrucciones:

```
kubectl get secret
kubectl describe secret mariadb
```

Veamos a continuación cómo quedaría un despliegue que usa el valor de un Secret para inicializar una variable de entorno. Vamos a usar el fichero [`mariadb-deployment-secret.yaml`](files/mariadb-deployment-secret.yaml)

y el fragmento donde definimos las variables de entorno quedaría:

```yaml
...
  spec:
    containers:
     - name: mariadb
       image: mariadb
       ports:
        - containerPort: 3306
          name: db-port
       env:
        - name: MYSQL_ROOT_PASSWORD
          valueFrom:
            secretKeyRef:
              name: mariadb
              key: password
```

Observamos como al indicar las variables de entorno (sección `env`) seguimos indicado el nombre (`name`) pero el valor se indica con un valor de un Secret (`valueFrom: - secretKeyRef:`), indicando el nombre del Secret (`name`) y la clave correspondiente. (`key`).

Nota: Por defecto, k8s no cifra el valor de los Secrets, simplemente los codifica en base64. El cifrado de los Secrets requiere configuraciones adicionales que se detallan en [Encrypting Secret Data at Rest](https://kubernetes.io/docs/tasks/administer-cluster/encrypt-data/)