

Ejercicio 1: Trabajo con Volumes (emptyDir)

Objetivo:

Comprender el funcionamiento de los volúmenes `emptyDir` en Kubernetes, que son útiles para compartir datos entre contenedores en el mismo Pod.

Pasos:

1. Crear un Pod con un Volumen `emptyDir`:

- Crea un archivo YAML llamado `emptydir-pod.yaml` con el siguiente contenido:

```
apiVersion: v1
kind: Pod
metadata:
  name: mypod
spec:
  containers:
    - name: mycontainer1
      image: nginx
      volumeMounts:
        - name: cache-volume
          mountPath: /cache
    - name: mycontainer2
      image: alien777/mi-aplicacion-node:v1
      volumeMounts:
        - name: cache-volume
          mountPath: /cache
  volumes:
    - name: cache-volume
      emptyDir: {}
```

- Aplica el manifiesto con: `kubectl apply -f emptydir-pod.yaml` .

Para experimentar con el ciclo de vida del volumen `emptyDir` en tu Pod `mypod` , sigue estos pasos:

2. Acceder al Primer Contenedor (`mycontainer1`):

```
kubectl exec -it mypod -c mycontainer1 -- /bin/sh
```

3. Escribir en `/cache` :

Una vez dentro del contenedor, crea un archivo en el directorio `/cache` . Por ejemplo:

```
echo "Hola Kubernetes" > /cache/saludo.txt
```

Luego sal del contenedor con `exit` .

4. Acceder al Segundo Contenedor (`mycontainer2`):

```
kubectl exec -it mypod -c mycontainer2 -- /bin/sh
```

5. Leer el Contenido de `/cache` :

Dentro del contenedor, verifica si puedes ver el archivo `saludo.txt` y su contenido:

```
cat /cache/saludo.txt
```

Si todo funciona como se espera, deberías ver "Hola Kubernetes". Luego sal del contenedor con `exit` .

6. Reiniciar un Contenedor:

Puedes simular un reinicio de contenedor deteniendo el proceso principal dentro del contenedor. Sin embargo, para contenedores como `nginx`, es más fácil simplemente cambiar de imagen y desplegar nuevamente, cambia temporalmente a `nginx:alpine` aplica el manifiesto y vuelve a cambiarlo

7. Verificar la Persistencia de Datos:

- Una vez que el Pod se reinicie (Kubernetes lo hará automáticamente), accede nuevamente a uno de los contenedores y verifica si el archivo `/cache/saludo.txt` todavía existe y tiene el mismo contenido.

8. Eliminar y Volver a Crear el Pod:

- Elimina el Pod con `kubectl delete pod mypod` .
- Vuelve a crear el Pod con `kubectl apply -f <archivo del Pod>.yaml` .

9. Verificar si los Datos se Mantienen:

- Una vez que el Pod esté en ejecución nuevamente, accede a uno de los contenedores y verifica si el archivo `/cache/saludo.txt` todavía existe.
- Dado que `emptyDir` es efímero y se elimina con el Pod, los datos no deberían persistir después de que el Pod se elimine y se vuelva a crear.

Ejercicio 2: Exploración de Storage Classes

Objetivo:

Familiarizarse con las clases de almacenamiento disponibles en el clúster y cómo usarlas en PVCs.

Pasos:

1. Revisar las Clases de Almacenamiento Disponibles:

- Ejecuta `kubectl get storageclass` para ver las clases de almacenamiento disponibles.

2. Crear un PVC que Solicite una Clase de Almacenamiento Específica:

- Crea un archivo llamado `storageclass-pvc.yaml` con el siguiente contenido (**ajusta `storageClassName` según lo que hayas encontrado**):

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: mypvc
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  storageClassName: [nombre del storage class]
```

- Aplica el manifiesto con: `kubectl apply -f storageclass-pvc.yaml` .

3. Verificar el Estado del PVC:

```
kubectl get pvc
```

- Busca tu PVC (por ejemplo, `mypvc`) y verifica que el estado sea `Bound` , lo que indica que está en uso.

4. Desplegar una Aplicación que Use el PVC:

Ejemplo básico para MySQL:

```

apiVersion: apps/v1
kind: Deployment
metadata:
  name: mysql-deployment
spec:
  replicas: 1
  selector:
    matchLabels:
      app: mysql
  template:
    metadata:
      labels:
        app: mysql
    spec:
      containers:
        - name: mysql
          image: mysql:5.7
          env:
            - name: MYSQL_ROOT_PASSWORD
              value: "my-secret-pw"
          volumeMounts:
            - name: mysql-storage
              mountPath: /var/lib/mysql
      volumes:
        - name: mysql-storage
          persistentVolumeClaim:
            claimName: mypvc

```

Aplica este manifiesto con **kubectl apply -f mysql.yaml**.

5. Verificar que el Pod Esté Utilizando el PVC:

```
kubectl get pod -o wide
```

- Identifica el Pod que está utilizando el PVC y toma nota de su nombre.

Una vez que hayas confirmado que el PVC está en uso, inspecciona el Pod para asegurarte de que está montando correctamente el PVC:

6. Describir el Pod:

```
kubectl describe pod <nombre_del_pod>
```

- Reemplaza `<nombre_del_pod>` con el nombre real del Pod.
- En la descripción del Pod, busca la sección que detalla los volúmenes y las monturas de volumen para confirmar que el PVC está montado en el contenedor.

7. Conectar a Base de Datos:

- Conéctate a la base de datos en el Pod y realiza algunas operaciones de lectura/escritura.

```
kubectl exec -it <nombre_del_pod> -- mysql -u root -p
```

8. Crear datos:

- Crea una base de datos llamada **movies**

```
CREATE DATABASE movies;
```

- Puedes usar el comando **SHOW DATABASES;** para verificar si la base de datos ha sido creada

```
SHOW DATABASES;
```

- Selecciona la base de datos

```
USE movies;
```

- Crea una tabla

```
CREATE TABLE movies(title VARCHAR(50) NOT NULL,genre VARCHAR(30) NOT NULL,director VARCHA
```



- Agrega información a la tabla

```
INSERT INTO movies VALUE ("Joker", "psychological thriller", "Todd Phillips", 2019);
```

- Verifica la información en la tabla

```
SELECT * FROM movies;
```

9. Eliminar y Recrear el Pod:

- Si tu PVC está configurado con `ReadWriteOnce`, solo un Pod puede montarlo a la vez. Puedes eliminar el Pod actual y dejar que el Deployment lo vuelva a crear.
- Usa `kubectl delete pod <nombre_del_pod>` y espera a que el Deployment cree un nuevo Pod.

10. Verificar Datos Después del Reinicio:

- Vuelve a conectarte a la aplicación o base de datos en el nuevo Pod.

- Verifica si los datos que creaste antes de eliminar el Pod todavía existen.