

Ayudas Práctica 2

```
# Dockerfile.apache
FROM php:apache
COPY ./html /var/www/html
```

```
# Dockerfile.mysql
FROM mysql
ENV MYSQL_ROOT_PASSWORD=root_password
ENV MYSQL_DATABASE=mydatabase
```

```
docker build -t my-php-apache-image -f Dockerfile.apache .
docker build -t my-mysql-image -f Dockerfile.mysql .
```

```
docker network create my-network
```

```
version: '3'
services:
  apache:
    image: my-php-apache-image
    ports:
      - "80:80"
    networks:
      - my-network
    depends_on:
      - mysql
```

```
mysql:
  image: my-mysql-image
  networks:
    - my-network
```

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

```
docker run -d --name mysql-container --network my-network my-mysql-image
```

```
MYSQL_IP=$(docker inspect -f '{{range .NetworkSettings.Networks}}{{.IPAddress}}{{end}}' mysql-container)
```

```
docker run -d --name php-apache-container --network my-network -p 80:80 --link mysql-container:mysql my-php-apache-image
```

```
docker network create my-network
```

```
docker run -d --name mysql-container --network my-network my-mysql-image
```

```
MYSQL_IP=$(docker inspect -f '{{range .NetworkSettings.Networks}}{{.IPAddress}}{{end}}' mysql-container)
```

```
docker run -d --name php-apache-container --network my-network -p 80:80 -e MYSQL_HOST=$MYSQL_IP my-php-apache-image
```

Si migras de Docker Compose a Kubernetes, necesitas ajustar el archivo de configuración para reflejar las especificidades de Kubernetes. En lugar de un archivo `docker-compose.yml`, en Kubernetes utilizas un archivo YAML llamado `deployment.yaml` (o un nombre similar) para describir tus aplicaciones y cómo deben ejecutarse en clústeres de Kubernetes.

A continuación, un ejemplo simple de cómo podría verse un archivo YAML de Kubernetes para un servicio Apache que escucha en el puerto 7777:

```
apiVersion: v1
kind: Pod
metadata:
  name: apache-pod
  labels:
    app: apache
```

```
spec:
  containers:
  - name: apache-container
    image: httpd:2.4
    ports:
    - containerPort: 7777
```

```
apiVersion: v1
kind: Service
metadata:
  name: apache-service
spec:
  selector:
    app: apache
  ports:
  - protocol: TCP
    port: 80
    targetPort: 7777
```

```
kubectl apply -f nombre_del_archivo.yaml
```