Cada vez que creamos un contenedor, esté se conecta a una red virtual y Docker hace una configuración del sistema para que la máquina tenga una ip interna, tenga

acceso al exterior, podamos mapear puertos,…

Para ver las redes definidas, podemos hacerlo mediante: docker network ls. Compruébalo

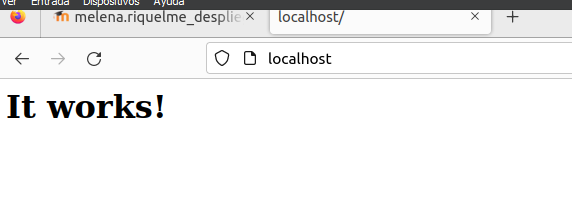
Por defecto los contenedores que creamos se conectan a la red de tipo bridge llamada bridge (por defecto el direccionamiento de esta red es 172.17.0.0/16).

Para conectar un contenedor a la red del equipo, se debe conectar a la red host. Prueba a conectar el contenedor de Apache a la red del equipo mediante:

docker run -d --name mi\_servidor --network host nombre\_de\_imagen



Prueba a accerder al puerto 80 con la dirección del equipo desde el navegador.



Podemos crear nuevas redes en modo puente. Crea una nueva red mediante:

docker network create mired



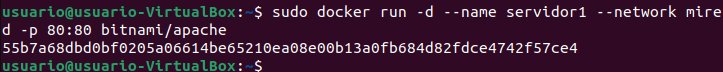
Observa sus características:

docker network inspect mired



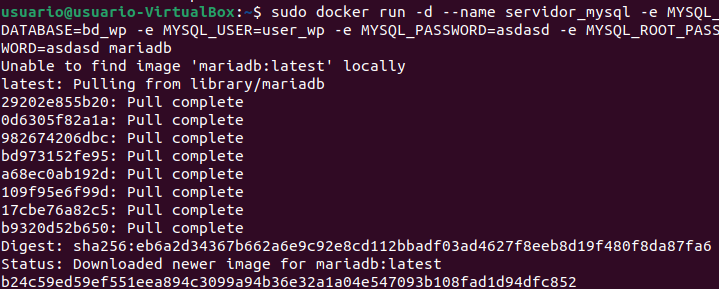
Y crea un contenedor (el de Apache) en esa red:

docker run -d --name mi\_servidor --network mired -p 80:80 nombre\_de\_imagen



Para ver cómo se enlazan contenedores, crea un contenedor de mariadb:

docker run -d --name servidor\_mysql -e MYSQL\_DATABASE=bd\_wp -e MYSQL\_USER=user\_wp -e MYSQL\_PASSWORD=asdasd -e MYSQL\_ROOT\_PASSWORD=asdasd mariadb



Y enlázalo con el contenedor Apache:

docker run -d --name servidor --link servidor\_mysql:mariadb nombre\_de\_imagen

Así, las variables de entorno del primer contenedor pueden ser accedidas por el segundo. Para verlas, ejecuta:

docker exec servidor env

Otro mecanismo que se realiza para permitir la comunicación entre contenedores asociados es modificar el fichero /etc/hosts. Podemos comprobarlo:

docker exec servidor cat /etc/hosts

Por último, conecta el contenedor de mariadb con la red que creaste anteriormente:

docker run -d --name servidor\_mysql --network mired -e MYSQL\_DATABASE=bd\_wp -e MYSQL\_USER=user\_wp -e MYSQL\_PASSWORD=asdasd -e MYSQL\_ROOT\_PASSWORD=asdasd mariadb

docker run -d --name servidor --network mired nombre\_de\_imagen

En este caso no se comparten las variables de entorno, y la resolución de nombres de los contenedores se hace mediante un servidor dns que se ha creado en el gateway de la red que hemos creado

docker exec -it servidor bash

apt update && apt install dnsutils -y

dig servidor\_mysql