EJERCICIOS DOCKERS - 3

**Ejercicios para entregar**

# Trabajar con redes docker

1. Vamos a crear dos redes de ese tipo (BRIDGE) con los siguientes datos:

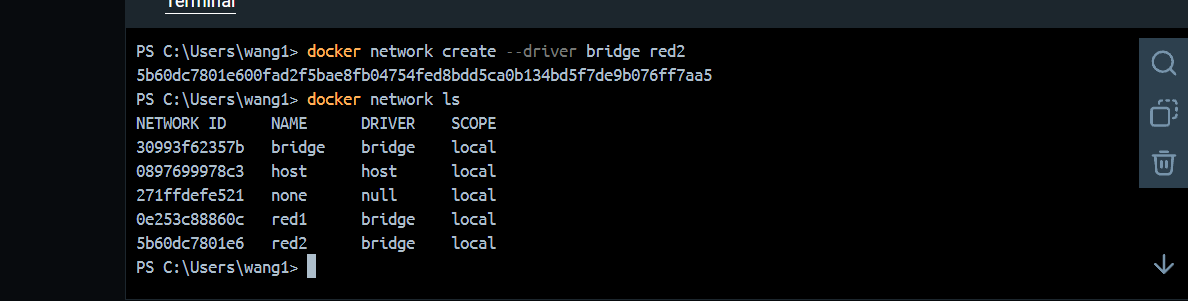
## Red1

* + Nombre: red1
  + Dirección de red: 172.28.0.0
  + Máscara de red: 255.255.0.0
  + Gateway: 172.28.0.1

## Red2

* + Nombre: red2
  + El resto de los datos será proporcionado automáticamente por Docker.



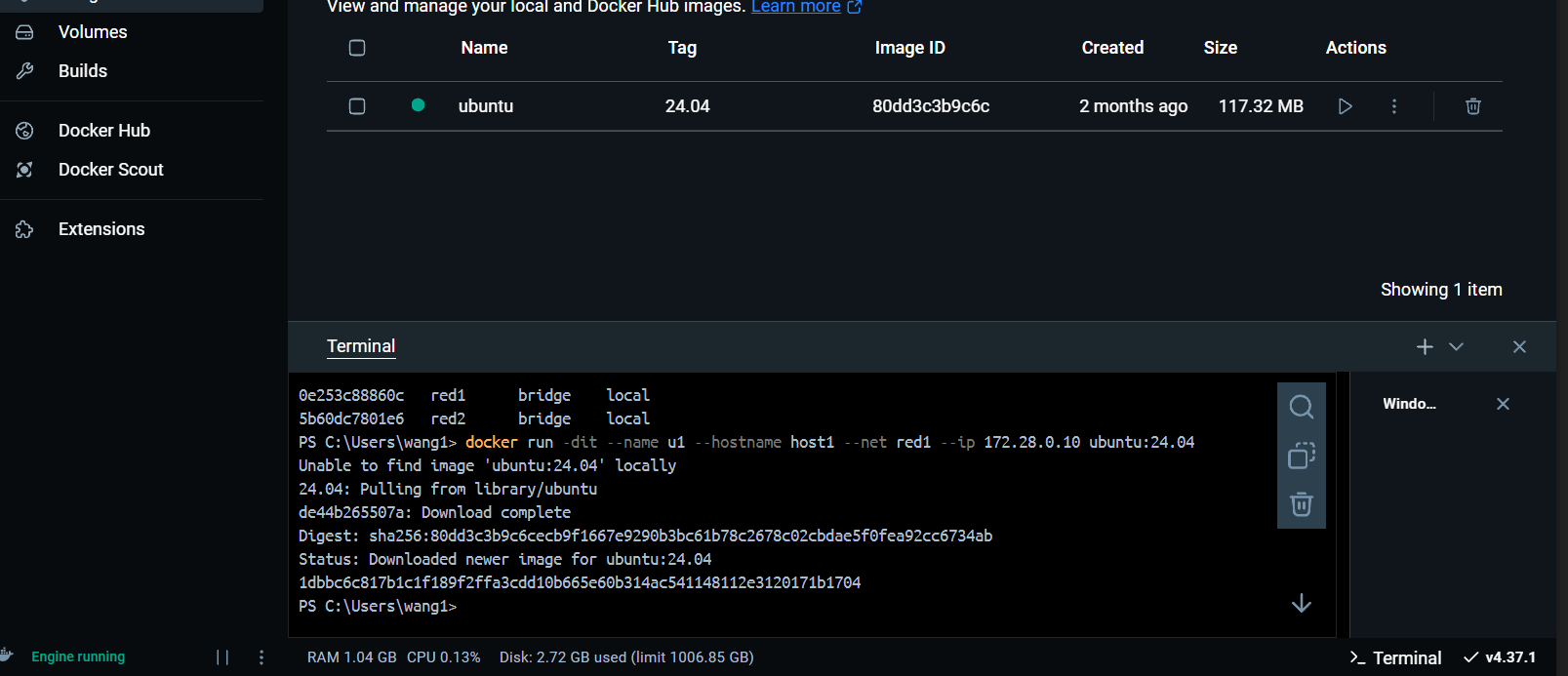


1. Poner en ejecución un contenedor de la imagen que tenga como

ubuntu:24.04

hostname host1 , como IP y que esté conectado a la red1. Lo llamaremos u1 .

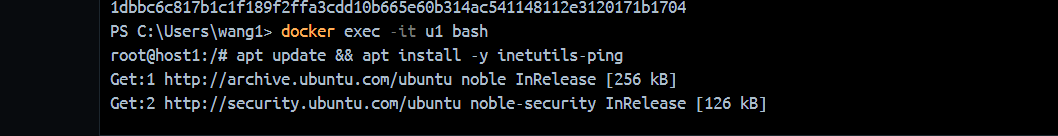
172.28.0.10



docker run -dit --name u1 --hostname host1 --net red1 --ip 172.28.0.10 ubuntu:24.04

1. Entrar en ese contenedor e instalar la aplicación ping ( apt update && apt install inetutils-

ping ).



docker exec -it u1 bash

apt update && apt install -y inetutils-ping

Exit

1. Poner en ejecución un contenedor de la imagen que tenga como

hostname y que esté conectado a la red2. En este caso será docker el que le dé una

host2

ubuntu:24.04

IP correspondiente a esa red. Lo llamaremos u2 .

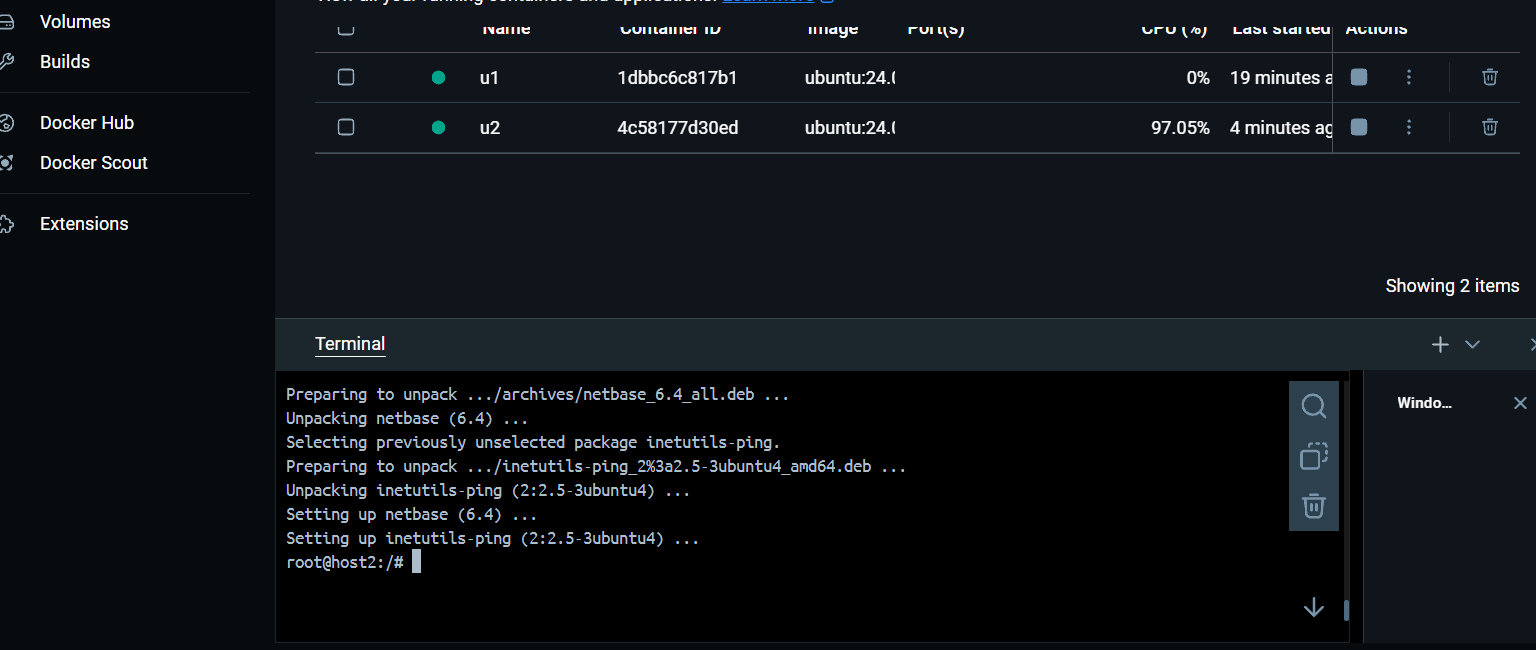
docker run -dit --name u2 --hostname host2 --net red2 ubuntu:24.04



1. Entrar en ese contenedor e instalar la aplicación ping ( apt update && apt install inetutils-

ping ).

z



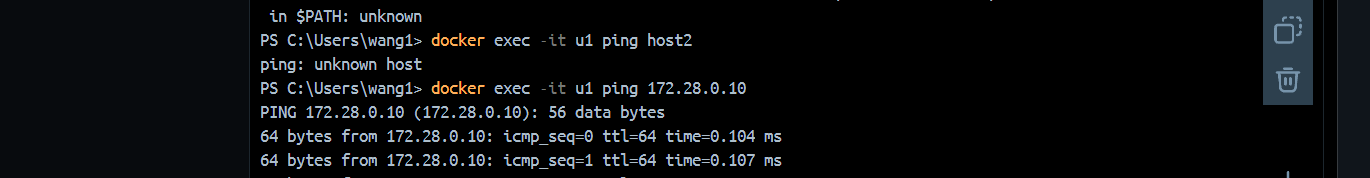
docker exec -it u2 bash

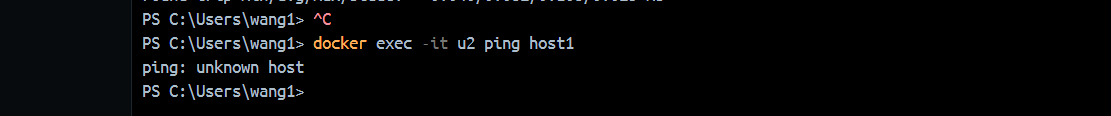
apt update && apt install -y inetutils-ping

exit

Deberás entregar los siguientes pantallazos en un documento pdf:

* Pantallazo donde se vea la configuración de red del contenedor u1.
* Pantallazo donde se vea la configuración de red del contenedor u2.
* Pantallazo donde desde cualquiera de los dos contenedores se pueda ver que no podemos hacer ping al otro ni por ip ni por nombre.





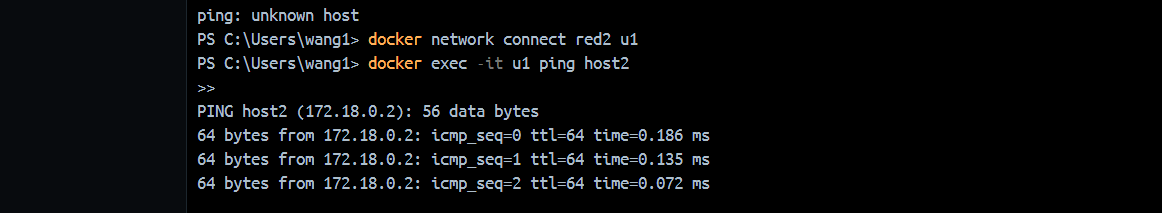
* Pantallazo donde se pueda comprobar que si conectamos el contenedor a la red2

u1

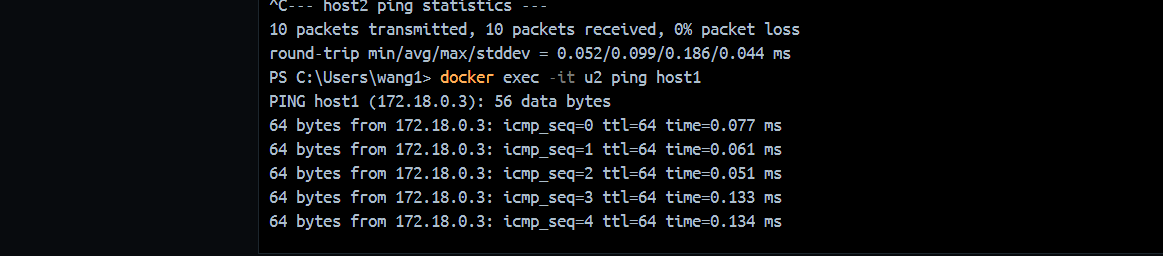
(con docker network connect ), desde el contenedor u1, tenemos acceso al contenedor u2 mediante ping, tanto por nombre como por ip.

docker network connect red2 u1

docker exec -it u1 ping host2



docker exec -it u1 ping host2



# Despliegue de Wordpress + Mariadb

Para la instalación de WordPress necesitamos dos contenedores: la base de datos (imagen mariadb ) y el servidor web con la aplicación (imagen wordpress ). Los dos contenedores tienen que estar en la misma red y deben tener acceso por nombres (resolución DNS) ya que de principio no sabemos que ip va a coger cada contenedor. Por lo tanto, vamos a crear los contenedores en la misma red.

## Crear una red Docker para ambos contenedores

docker network create wordpress-net

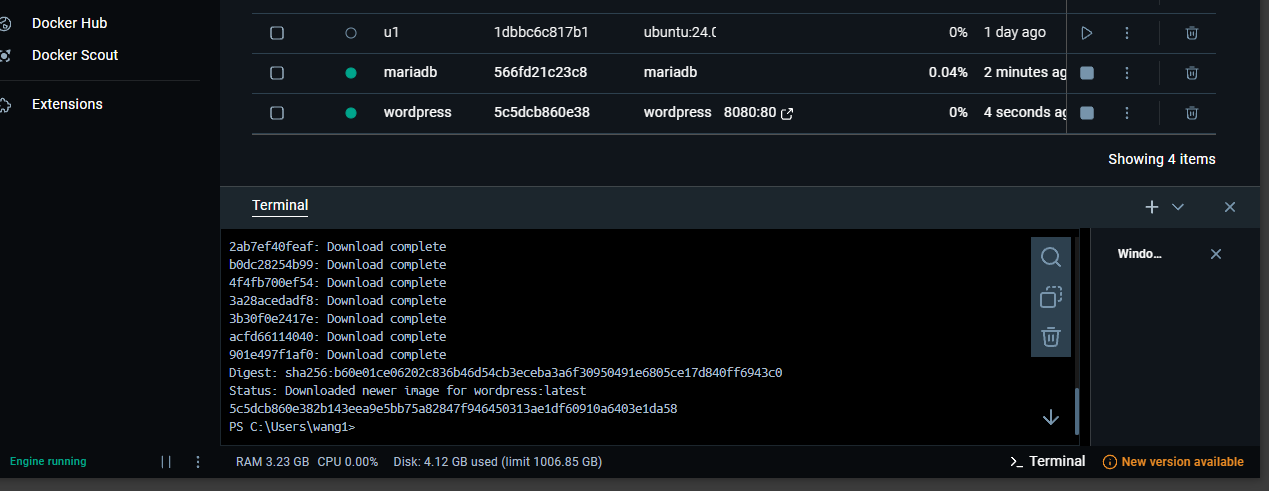
## Crear el contenedor de MariaDB

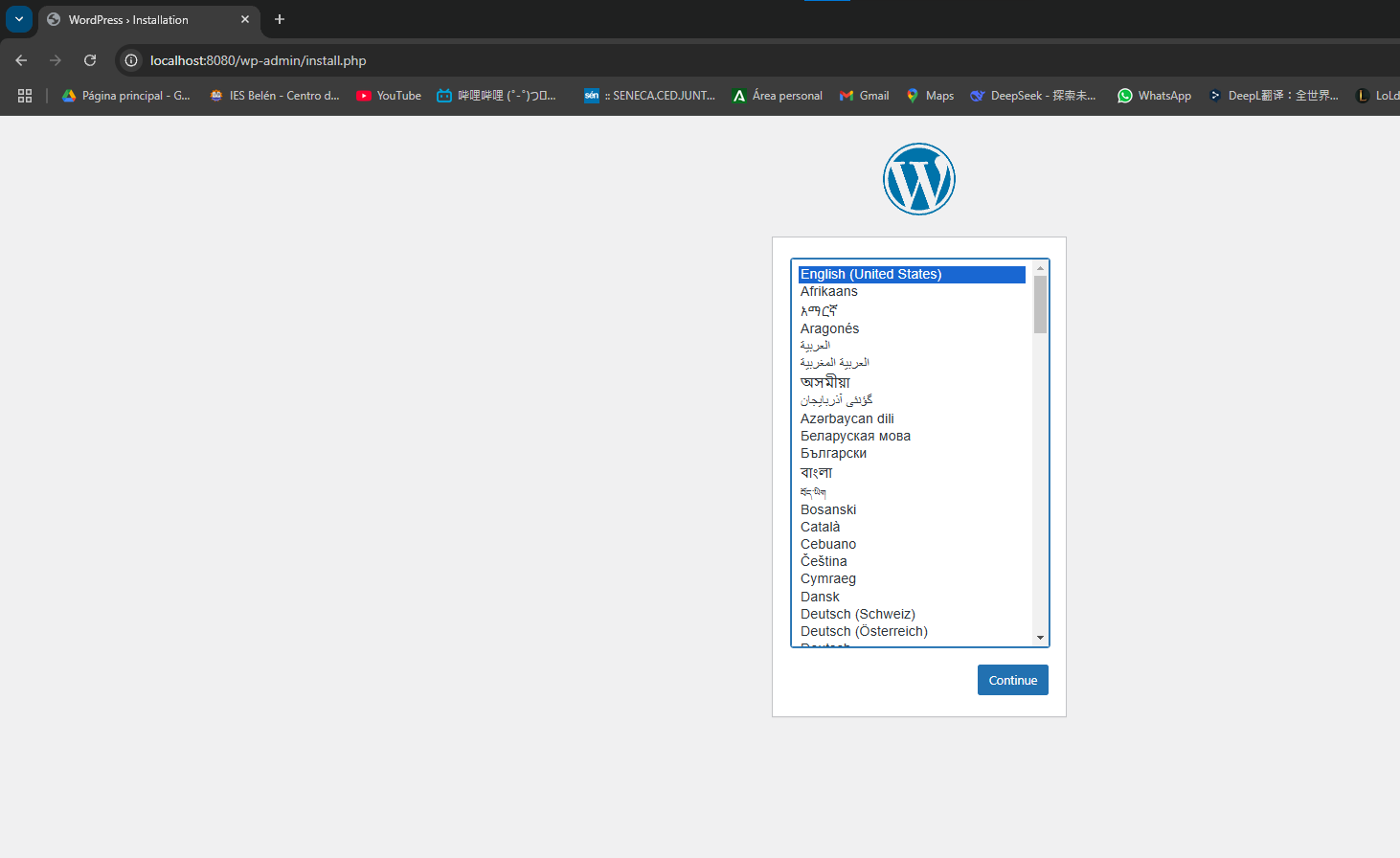
docker run -d --name mariadb --network wordpress-net -e MYSQL\_ROOT\_PASSWORD=root -e MYSQL\_DATABASE=wordpress -e MYSQL\_USER=wp\_user -e MYSQL\_PASSWORD=wp\_pass mariadb



### Crear el contenedor de WordPress

docker run -d --name wordpress --network wordpress-net -e WORDPRESS\_DB\_HOST=mariadb -e WORDPRESS\_DB\_USER=wp\_user -e WORDPRESS\_DB\_PASSWORD=12345678 -e WORDPRESS\_DB\_NAME=wordpress -p 8080:80 wordpress





Deberás entregar los siguientes pantallazos en un documento pdf:

* Pantallazo donde se vea la creación del servidor de Mariadb
* Pantallazo donde se vea la creación del CMS WordPress con los parámetros de la BBD.
* Pantallazo del acceso al WordPress desde el navegador.