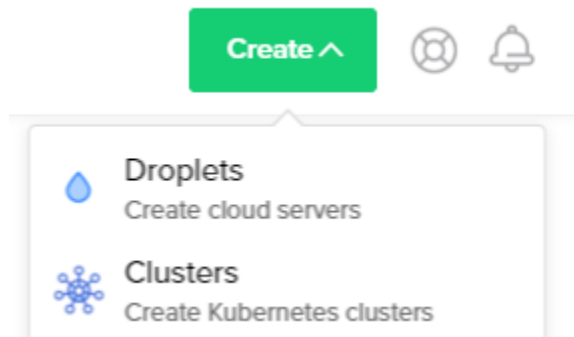


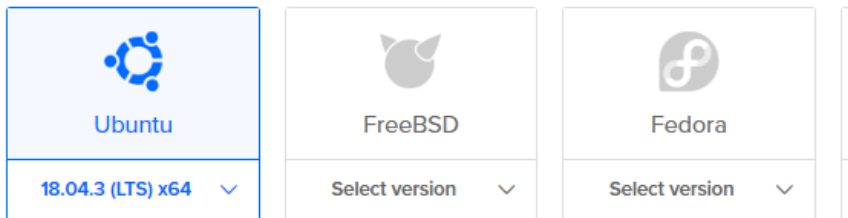
1- Creamos nuestra cuenta en digital ocean. Y seleccionamos Create y luego droplets tran entrar.



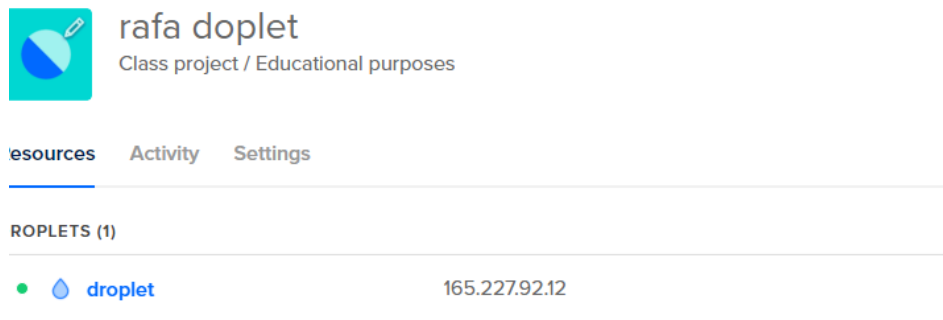
2-seleccionamos en imange Ubuntu, y en todas las opciones demás dejamos las predeterminadas

Choose an image ?

Distributions Container distributions Marketplace Custom images



3-tras haber creado el droplet, esta es la ip con la que nos conectaremos a él.



4- nos conectamos con el siguiente comando. Le damos aceptar y escribimos la contraseña que nos llega al correo. Es de un solo uso así que después la hay que cambiar.

```
C:\Users\Rafa>ssh root@165.227.92.12
The authenticity of host '165.227.92.12 (165.227.92.12)' can't be established.
ECDSA key fingerprint is SHA256:SwA/uN7DExxqV6V4Q/OjME+xZOF0+8QCih033yMYbwY.
Are you sure you want to continue connecting (yes/no)? yes
```

5- actualizamos los paquetes de apt

```
root@droplet:~# sudo apt-get update
Get:1 http://mirrors.digitalocean.com/ubuntu bionic InRelease [242 kB]
Get:2 http://mirrors.digitalocean.com/ubuntu bionic-updates InRelease [88.7 kB]
Get:3 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
Get:4 http://mirrors.digitalocean.com/ubuntu bionic-backports InRelease [74.6 kB]
Get:5 http://mirrors.digitalocean.com/ubuntu bionic/universe amd64 Packages [8570 kB]
Get:6 http://security.ubuntu.com/ubuntu bionic-security/main amd64 Packages [670 kB]
```

6- instalamos la actualización

```
root@droplet:~# sudo apt-get install \
> apt-transport-https \
> ca-certificates \
> curl \
> gnupg-agent \
> software-properties-common
Reading package lists... Done
Building dependency tree
Reading state information... Done
```

7- Agregamos la llave GPG de Docker y creamos un repositorio estable. Al finalizar ejecutamos nuevamente el comando del paso 5.

```
root@droplet:~# curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
OK
root@droplet:~# sudo add-apt-repository \
> "deb [arch=amd64] https://download.docker.com/linux/ubuntu \
> $(lsb_release -cs) \
> stable"
```

8- instalamos la última versión de docker

```
root@droplet:~# apt-get install docker-ce docker-ce-cli containerd.io
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  grub-pc-bin
```

9- descargamos la ultima versión de Docker compose, y le damos permisos con el ultimo comando

```
root@droplet:~# sudo curl -L "https://github.com/docker/compose/releases/download/1.25.4/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
% Total    % Received % Xferd Average Speed   Time    Time     Time  Current
           Dload  Upload   Total   Spent    Left    Speed
100 617 100 617 0 0 2843 0 --:--:-- --:--:-- --:--:-- 2843
100 16.3M 100 16.3M 0 0 35.6M 0 --:--:-- --:--:-- --:--:-- 35.6M
root@droplet:~# sudo chmod +x /usr/local/bin/docker-compose
```

10- clonamos el repositorio de github a usar

```
root@droplet:~# git clone https://github.com/zakareto/docker.git
Cloning into 'docker'...
remote: Enumerating objects: 121, done.
remote: Counting objects: 100% (121/121), done.
remote: Compressing objects: 100% (41/41), done.
remote: Total 121 (delta 65), reused 105 (delta 62), pack-reused 0
Receiving objects: 100% (121/121), 7.96 MiB | 9.25 MiB/s, done.
Resolving deltas: 100% (65/65), done.
```

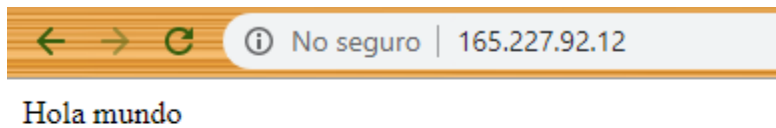
11- ejecutamos Docker-compose up. Para esto, hay que ver ubicarse en la misma carpeta donde esté el archivo .yaml

```
root@droplet:~/docker# ls
archivos dbconfig dbfiles docker-compose.yml
root@droplet:~/docker# docker-compose up
Creating network "docker_default" with the default driver
Pulling db (mariadb/server:10.4)...
10.4: Pulling from mariadb/server
5c939e3a4d10: Pull complete
c63719cdbe7a: Pull complete
19a861ea6baf: Pull complete
651c9d2d6c4f: Pull complete
3442154e0808: Pull complete
Digest: sha256:3442154e08080808080808080808080808080808080808080808080808080808
```

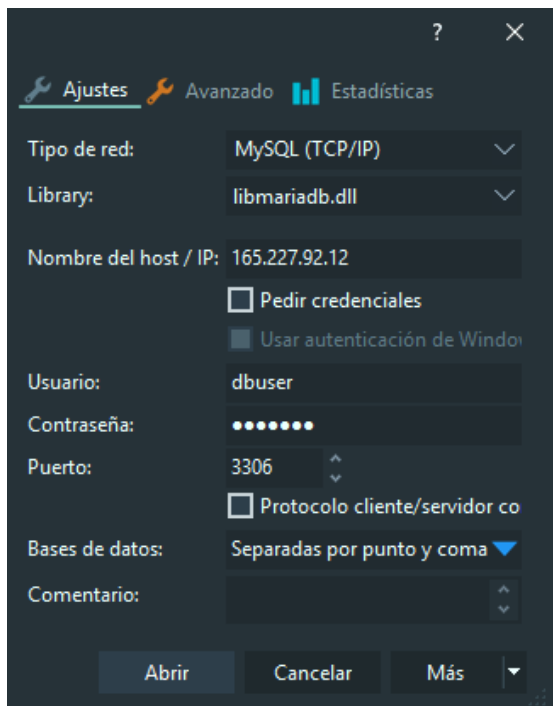
12- tras un rato, si el proceso no termina, oprimimos Ctrl + C para forzar el cierre, y ejecutamos el comando Docker-compose up -d para levantar el contenedor.

```
167 bytesOut:180 ReqTime:0
app_1 | [SYSLOG] CRON[121]: (root) CMD ( [ -x /usr/lib/php
/usr/lib/php/sessionclean; fi)
^CGracefully stopping... (press Ctrl+C again to force)
Stopping docker_app_1 ... done
Stopping docker_db_1 ... done
root@droplet:~/docker# docker-compose up -d
Starting docker_app_1 ... done
Starting docker_db_1 ... done
```

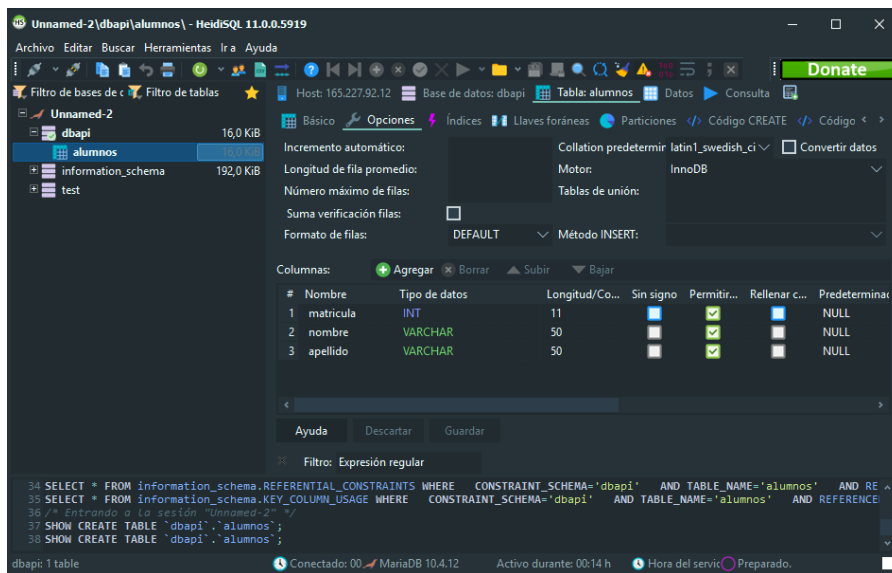
13- la pagina ya debería ser accesible desde la dirección ip



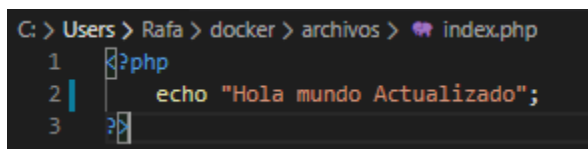
14- agregamos la base de datos a heidiSQL escribiendo la ip, y las credenciales correspondientes, encontradas dentro del archivo .yaml



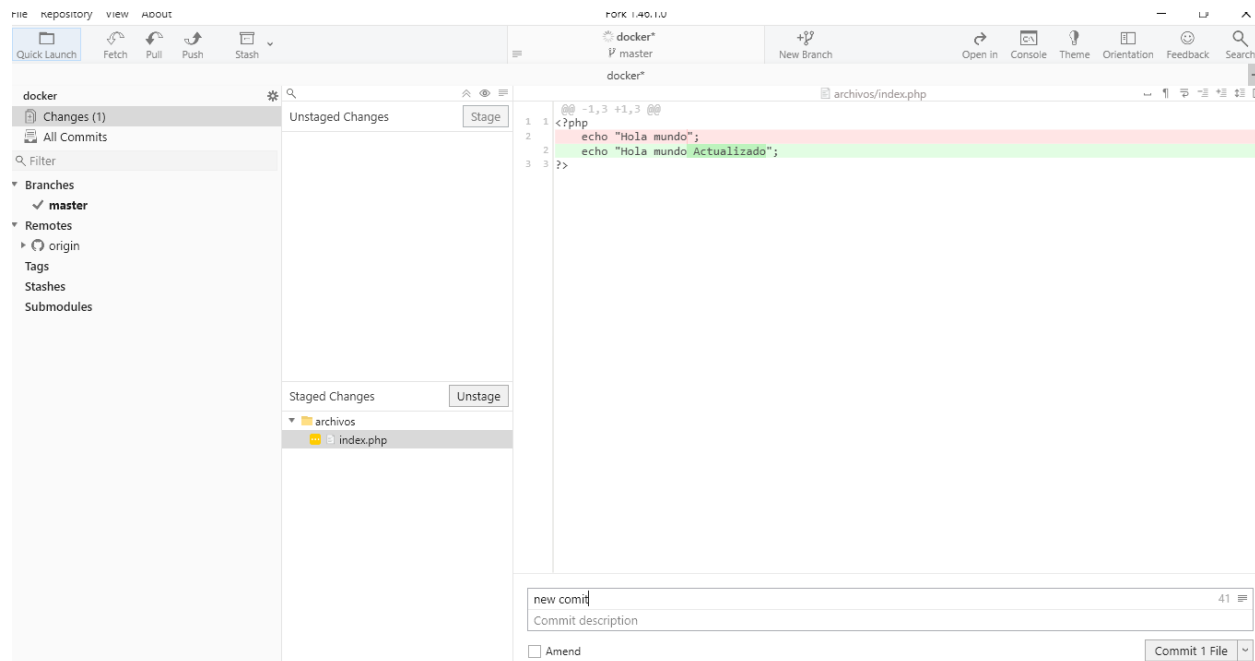
15- tras esto, ya podemos agregar tablas a la base de datos del contenedor, y modificarla



16- modificamos el archivo index.html



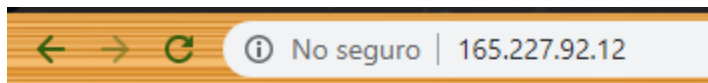
17- hacemos un push desde fork para actualizar el archivo en el repositor



18- hacemos un pull en el contenedor

```
root@droplet:~/docker# git pull origin master
remote: Enumerating objects: 7, done.
remote: Counting objects: 100% (7/7), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 4 (delta 0), reused 4 (delta 0), pack-reused 0
Unpacking objects: 100% (4/4), done.
From https://github.com/zakareto/docker
* branch                master       -> FETCH_HEAD
  7de4e0a..8289de8      master       -> origin/master
Updating 7de4e0a..8289de8
Fast-forward
 archivos/index.php | 2 +
 1 file changed, 1 insertion(+), 1 deletion(-)
```

19 – el cambio debe verse reflejado en el navegador al acceder a la dirección ip



Hola mundo Actualizado