



docker

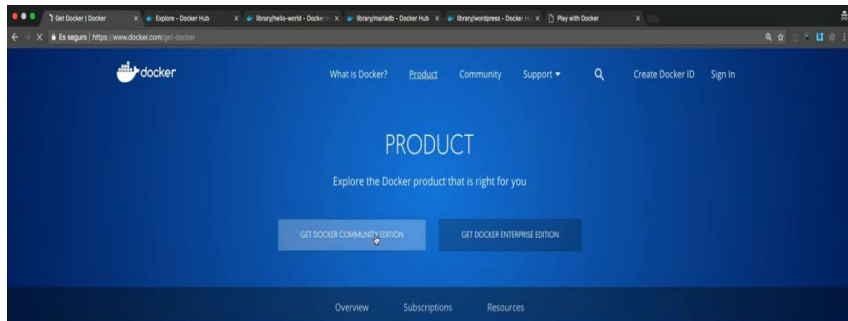
Como crear

Contenedores Docker en la nube

Kerly Valentina Triviño Gonzalez
Juan Esteban Atehortua Cardona
Natalia Higueta Higueta
Valentina Osorno Monsalve



Vamos a ingresar a docker.com, damos click en **product**, y si tenemos instalado el software nos vamos para **get docker community edition**, que es la version para desarrollar y trabajar sobre ella de modo gratuito y por ultimo **download**, si aún no tenemos descargado el software, podemos bajar la pantalla y allí podemos elegir la plataforma que deseamos para la instalación del software.



Get started with Docker, the world's leading software container platform to modernize applications without disruption. Docker is available as Community Edition (CE) and Enterprise Edition (EE), with optimized installers for a variety of infrastructure. The Docker platform and associated toolchain provides the following features as a baseline for both CE and EE.



UNIVERSAL PACKAGING

Easily package apps written in any programming language or service into containers without the risk of incompatibilities or version conflicts. Isolated containers are then portable to any machine running Docker.



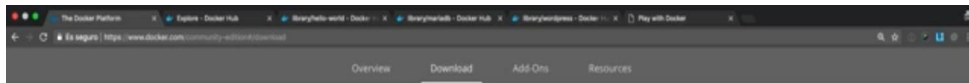
COMPLETE DEVELOPER TOOLKIT

Everything you need to build, test and run multi-container apps from an interactive CLI with support for any programming language. Start from scratch or browse and download from thousands of ready to use containers in Docker Store.



BUILT-IN CONTAINER ORCHESTRATION

Docker comes with clustering to run apps at scale and sophisticated scheduling and monitoring to build highly available and fault tolerant services



Download Docker Community Edition

Developer Desktops



DOCKER CE FOR MAC

An integrated, easy-to-deploy Docker development environment on the Mac for building, assembling, and shipping applications.

[Download from Docker Store](#) [Learn More](#)



DOCKER CE FOR WINDOWS

A native Windows desktop application to easily setup a Docker development environment on a Windows PC.

[Download from Docker Store](#) [Learn More](#)

Linux Servers



DOCKER CE FOR CENTOS DISTRIBUTION

Installer for CentOS distribution environments
Also available with support and certification with Docker EE for CentOS distribution.

[Download from Docker Store](#) [Learn More](#)



DOCKER CE FOR DEBIAN

Installer for Debian environments

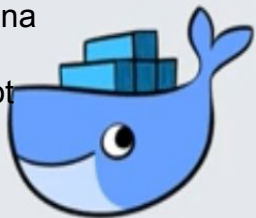
[Download from Docker Store](#) [Learn More](#)



docker

Como se crea Docker en la nube?

Se utiliza una terminal llamada root



Play with Docker

A simple, interactive and fun playground to learn Docker

Start



-para la cuenta de utilización de docker gratis utilizamos este sitio web el servidor gratuito llamado play with docker

-el programa tiene una durabilidad de máximo 4 horas

-buscamos la imagen

una imagen estándar

“hola mundo”

-para hacer la aprobación con: DOCKER PS-A

-se accede a descargar la base de datos mariadb(ella se descarga dentro de la máquina docker)



Los comandos de mariaDB

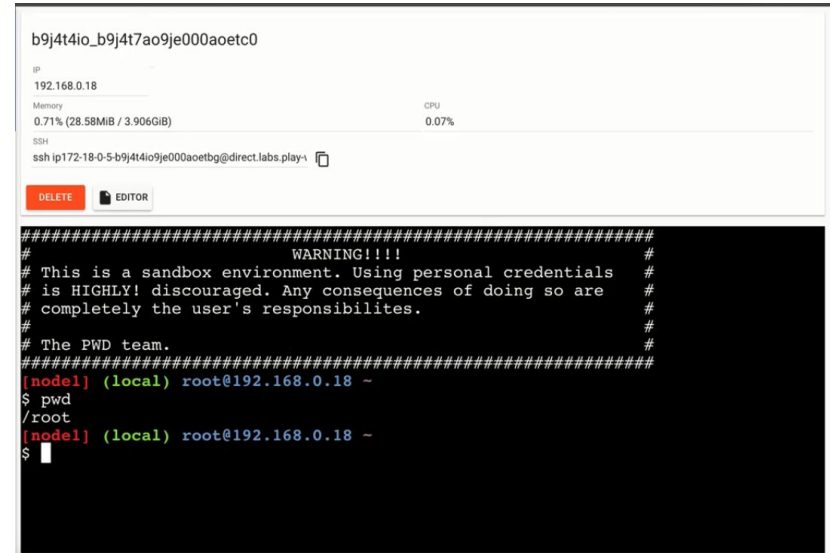
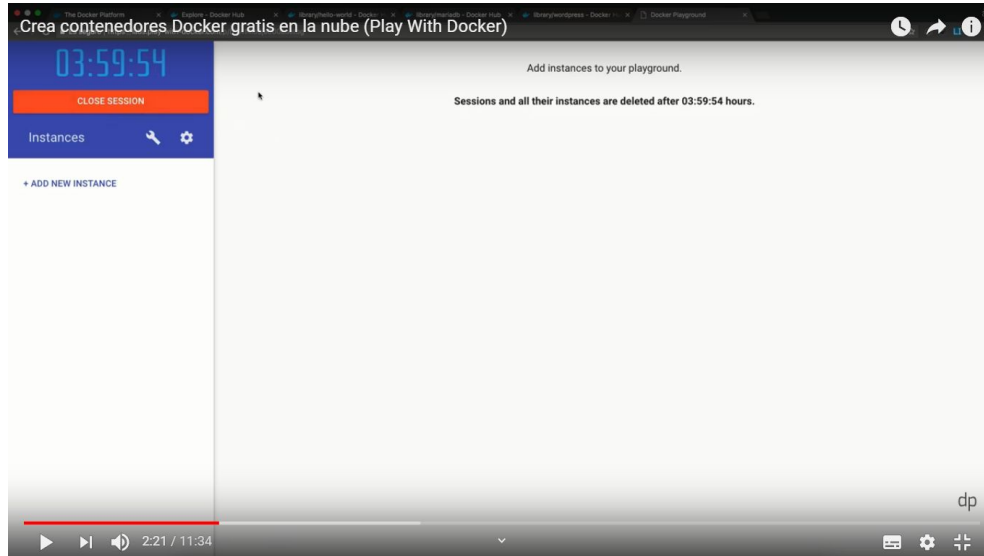
```
b9j4t4io_b9j4t7ao9je000aoetc0
IP
192.168.0.18
Memory
1.00% (39.83MiB / 3.90GiB)
CPU
0.23%
SSH
ssh ip172-18-0-5-b9j4t4io9je000aoetbg@direct.labs.play-
[DELETE] [EDITOR]
[nodet] (local) root@192.168.0.18 ~
$ docker ps -a
CONTAINER ID        IMAGE               PORTS              COMMAND              NAMES              CREATED
STATUS
fa3ef6213ea6       hello-world         "/hello"           modest_lalande      39 seconds ago
Exited (0) 38 seconds ago
[nodet] (local) root@192.168.0.18 ~
$
```

Arrancamos con una base de datos de mariadb es una rama de MYSQL, estas hace unas series de cosas que hacen descargar las imágenes de mariadb en la página de docker lo que hace es una instalación la cual extrae el contenido descargado no está dentro del ordenador si no que esta adentro de la maquina de docker, mariadb ejecuta un toque de RAM para ejecutar la imagen y levantar el ordenador de varios parámetros

MARIADB: Realiza cambios en los sitios web con solo cambiar un archivo, la cual no hay necesidad de modificar todo el código web.



al ver accedido a docker no dirigimos donde dice new instante y esta lo que va hacer es crear un nuevo servidor en la cual en la vamos a trabajar y en docker como hay muchas ayudas visuales para poder entender lo mejor empezar con el terminal, y en la parte de arriba aparece IP, el uso de memoria etc.





docker

Crea contenedores Docker gratis en la nube (Play With Docker)

```
1 docker run
2   -e MYSQL_ROOT_PASSWORD=mysqlpass
3   -e MYSQL_DATABASE=wordpress
4   --name wordpressdb
5   -v "$PWD/database":/var/lib/mysql
6   -d mariadb:latest
```

8:41 / 11:34 Lín. 5, Gol. 9 Espacios: 4 UTF-8 LF

1. Elegimos que contraseña le vamos a colocar
2. El nombre que le vamos a colocar a la base de datos.
3. El nombre que le vamos a colocar al contenedor, que en este caso es Wordpressdb.
4. Lo que está dentro del contenedor, queremos que esté disponible en la otra ruta del servidor "\$PWD/database" el cual corresponde al mismo directorio que estamos lanzando el mismo comando de (DOCKER RUN)
5. Y por último [mariadb: latest] sería la imagen que estamos descargando.



docker

Crea contenedores Docker gratis en la nube (Play With Docker)



How to use this image

```
$ docker run --name some-wordpress --link some-mysql:mysql -d wordpress
```

The following environment variables are also honored for configuring your WordPress instance:

- `-e WORDPRESS_DB_HOST=...` (defaults to the IP and port of the linked `mysql` container)
- `-e WORDPRESS_DB_USER=...` (defaults to "root")
- `-e WORDPRESS_DB_PASSWORD=...` (defaults to the value of the `MYSQL_ROOT_PASSWORD` environment variable from the linked `mysql` container)
- `-e WORDPRESS_DB_NAME=...` (defaults to "wordpress")
- `-e WORDPRESS_TABLE_PREFIX=...` (defaults to "", only set this when you need to override the default table prefix in `wp-config.php`)
- `-e WORDPRESS_AUTH_KEY=...`, `-e WORDPRESS_SECURE_AUTH_KEY=...`, `-e WORDPRESS_LOGGED_IN_KEY=...`, `-e WORDPRESS_NONCE_KEY=...`, `-e WORDPRESS_AUTH_SALT=...`, `-e WORDPRESS_SECURE_AUTH_SALT=...`, `-e WORDPRESS_LOGGED_IN_SALT=...`, `-e WORDPRESS_NONCE_SALT=...` (default to unique random SHA1s)

If the `WORDPRESS_DB_NAME` specified does not already exist on the given MySQL server, it will be created automatically upon startup of the `wordpress` container, provided that the

`WORDPRESS_DB_USER` specified has the necessary permissions to create it.

Acá nos muestran los parámetros que podemos utilizar.
Pero eso lo veremos luego ya que trae mucha información.



docker

```
b9j4t4io_b9j4t7ao9je000aoetC0

IP
192.168.0.18

Memory
21.09% (843.4MiB / 3.906GiB)

CPU
0.17%

SSH
ssh ip172-18-0-5-b9j4t4io9je000aoetbg@direct.labs.play

[DELETE] [EDITOR]

(node1) (local) root@192.168.0.18 ~
$ ls html/
index.php          wp-config-sample.php  wp-login.php
license.txt        wp-config.php         wp-mail.php
readme.html        wp-content            wp-settings.php
wp-activate.php    wp-cron.php           wp-signup.php
wp-admin           wp-includes           wp-trackback.php
wp-blog-header.php wp-links-opml.php     xmlrpc.php
wp-comments-post.php wp-load.php

(node1) (local) root@192.168.0.18 ~
$ ls database/
aria_log.00000001  ib_logfile1          mysql
aria_log_control  ibdata1              performance_schema
ib_buffer_pool    ibtmp1               tc.log
ib_logfile0       multi-master.info    wordpress

(node1) (local) root@192.168.0.18 ~
$
```

[ls]= Volúmenes

Encontramos el fichero de la base de datos y junto con ella encontramos la base de datos WORDPRESS.

hacemos esto por lo que queremos trabajar sobre esta base de datos aunque esto implique tiremos a wordpress de ultimo, eso no importa por lo que tenemos toda la información guardada, la permanencia de los datos depende al volumen, depende de este emparejamiento de un fichero de que ya existe en el servidor con uno que existe en nuestra máquina de docker con



docker

```
Crea contenedores Docker gratis en la nube (Play With Docker)
1  docker run
2      -e WORDPRESS_DB_PASSWORD=mysqlpass
3      --name wordpress
4      --link wordpressdb:mysql
5      -p :80:80
6      -v "$PWD/html":/var/www/html
7      -d wordpress
```

En este caso que Wordpress es el segundo comando que ejecutamos. Tenemos: la contraseña que evidentemente es la misma que usamos anteriormente, el nombre del contenedor [wordpress], establecemos un enlace de la wordpressdb:mysql, el puerto 80 para que salga por el puerto 80 de la web, también tenemos el volumen que ya anteriormente lo hemos hablado, y por último el nombre del contenedor.



docker

<https://www.youtube.com/watch?v=YNXJDyAj4dM>